Université Gustave Eiffel

Université Badji Mokhtar-Annaba

École doctorale : N° 532 MSTIC

Faculté : Technologie
Département : Informatique

THESE

Présentée en vue de l'obtention du diplôme de
Doctorat 3ème cycle

L'amélioration de la sécurité dans l'internet des objets

Filière : Informatique

Spécialité : Réseaux et Sécurité Informatique

Par

Ilyes Ahmim

Composition du Jury

Président du jury :	Makhlouf Derdour	Professeur	Université Larbi Ben M'hidi - Oum El Bouaghi
Rapporteure :	Ghoulmi Nacira	Professeure	Université Badji Mokhtar - Annaba
Examinateur :	Rachid Abdelrezak	Professeur	Université Gustave Eiffel - France
Examinateur :	Hafid Mohamed	Professeur	Université Badji Mokhtar - Annaba
Examinateur :	Sahraoui Abdelatif	MCA	Université Larbi Tebessi - Tébessa
Examinateur :	Khoukhi Lyes	Professeur	ENSICAEN - France
Invité :	Bouakkaz Feriel	MCB	EFREI Paris - France

Année 2025

الملخص

النمو السريع لإنترنت الأشياء (IoT) وتكنولوجيا الاتصالات المحمولة يوفر تقدمًا كبيرًا عبر مختلف المجالات، وخاصة في المناطق الحضرية، حيث يقدم حلولًا مبتكرة لمشكلة الازدحام المروري، وهو التحدي الذي فشلت التوسعات في البنية التحتية الأساسية للنقل في حله بفعالية. تتيح أنظمة الإنترنت الخاصة بالمركبات الحديثة (IoV) تفاعلًا سلسًا بين المستخدمين والمركبات وأجهزة إنترنت الأشياء ومنصات الخدمات، بهدف تحسين كفاءة النقل، والسلامة على الطرق، وحماية البيئة. ومع ذلك، لا تزال هناك عدة تحديات، خاصة فيما يتعلق بنقل البيانات بشكل آمن وحماية خصوصية المستخدمين. بالإضافة إلى ذلك، فإن الحجم الكبير للبيانات في الوقت الفعلي التي تنتجها المركبات والمسافرون وأجهزة إنترنت الأشياء يخلق عبئًا كبيرًا على الخوادم، مما يؤدي إلى تحديات في التأخير التي تتطلب حلولًا فعالة.

يظهر تحدٍ آخر في الحالات الطارئة، مثل الازدحام المروري في المناطق ذات التقاطعات المعقدة، وفي المناطق الريفية أو المتضررة من الكوارث حيث يصبح تركيب وحدات الطرق (RSU) مكلفًا للغاية. يبقى تقديم المعلومات في الوقت الفعلي للمستخدمين في هذه الحالات تحديًا رئيسيًا يجب التغلب عليه. ولحسن الحظ، يُتوقع استخدام الطائرات بدون طيار لاستعادة الاتصالات الطارئة في المواقف الحرجة لأنها قادرة على تغطية مناطق شاسعة، وتتحرك أسرع من المركبات الأرضية، وتتركز على المشكلات العاجلة، وليست مقيدة بشبكات الطرق. ومع ذلك، فإن التحدي الرئيسي يكمن في إنشاء اتصال آمن في الوقت الفعلي بين الطائرات بدون طيار والمستخدمين، وضمان التحقق الصحيح من هوية المستخدمين.

بالإضافة إلى ذلك، فإن نشر عدة طائرات بدون طيار تعمل معًا لإنجاز مهام معقدة يعقد تطبيق اللوائح الجديدة "RemoteID" التي أجبرت إدارة الطيران الفيدرالية (FAA) على استخدامها ابتداءً من عام 2022، والتي تتطلب من كل طائرة بدون طيار بث هويتها. لمنع الكشف عن الهويات الحقيقية للطائرات في السرب، من الضروري اقتراح بروتوكول يسمح للطائرات بالتوثيق بطريقة مجهولة. في مواجهة هذه التحديات، تنشأ مشكلة رئيسية أخرى وهي أن أساليب الأمان التقليدية لا يمكن تطبيقها بفعالية على إنترنت المركبات وإنترنت الطائرات بدون طيار بسبب استهلاكها الكبير للموارد.

في هذه الأطروحة، نقترح حلولاً للتوثيق مصممة خصيصًا لمعالجة هذه التحديات، مع التركيز بشكل خاص على الأداء لضمان ملاءمتها للتطبيقات العملية. تركز المساهمة الأولى على تحسين بروتوكول توثيق تم نشره مؤخرًا ويتضمن ثلاثة كيانات (المستخدم، مركز البيانات، والمركبة). أظهر تحليلنا للأمان وجود ثغرة رئيسية تتعلق بسرية الرسائل، مما يسمح للمهاجم باستنتاج مفتاح الجلسة المشترك بين المركبة والكيانات الأخرى. تعالج تحسيناتنا هذه الثغرة، مما يجعل البروتوكول قابلاً للتطبيق في إنترنت الأشياء المتنقل. تتضمن تحسينات أخرى تعزيز بروتوكول منشور آخر، تم تكييفه لجمع البيانات الكبيرة والأنظمة التي تتطلب استجابات سريعة. يلغي البروتوكول التواصل المباشر بين المستخدم ومركز البيانات، بهدف تقليل التكاليف وتقليل المخاطر المرتبطة بهذه الاتصالات المباشرة. سمح لنا تحليلنا للأمان بتصحيح خلل يؤثر على توثيق الهوية بين المستخدمين وأجهزة إنترنت الأشياء المنتشرة، مما يضمن وصولًا آمنًا إلى البيانات.

تتعلق المساهمة الثانية بالتكامل الآمن للطائرات بدون طيار، حيث اقترحنا بروتوكولًا جديدًا للتوثيق وتبادل المفاتيح يسمى "LASeR"، مصممًا للتفاعلات بين المستخدمين والطائرات بدون طيار المنتشرة في مناطق محددة. يتميز "LASeR" بالوصول القائم على المناطق، مما يضمن المراقبة المستمرة مع تقديم حل مثالي بين التكلفة

والأمان. أظهر تحليل أمان رسمي، تم تنفيذها باستخدام أدوات مثل "Scyther" و"AVISPA" و"Tamarin"، بالإضافة إلى تحليل غير رسمي، قوته ضد اثني عشر هجومًا معروفًا.

تتناول المساهمة الثالثة تحدي أسراب الطائرات بدون طيار، حيث اقترحنا بروتوكول توثيق جديد يسمى "2AS-DS"، متوافق مع لوائح إدارة الطيران الفيدرالية مع ضمان إخفاء الهوية وعدم إمكانية تتبع الاتصالات. تُظهر التحليلات الأولية للأمان أن "2AS-DS" تتحمل هجمات مثل الاعتراض، وإعادة التشغيل، والاستنساخ، بينما تقلل من الحمل على محطة الثقة، مما يحسن الأداء من حيث الكفاءة والسرعة.

في الختام، تسهم الحلول المقترحة في هذه الأطروحة بشكل فعال في تعزيز أمان الاتصالات في إنترنت الأشياء، مع التركيز بشكل خاص على الكائنات المتنقلة، مثل المركبات والطائرات بدون طيار، التي تلعب دورًا محوريًا في الثورة التكنولوجية العالمية.

الكلمات المفتاحية: إنترنت الأشياء، بروتوكول الأمان، الهجوم الإلكتروني، المصادقة، التشفير الخفيف، تحليل الشفرات، أنظمة النقل الذكية

Abstract

The fast growth of the IoT (Internet of Things) and mobile communication technology offers significant advancements across various fields. In particular, in urban areas, these technologies provide innovative solutions to the problem of traffic congestion. Modern IoV (Internet of vehicles) systems enable seamless interaction between users, vehicles, IoT devices, and service platforms, aiming to improve transport efficiency, road safety, and environmental protection. However, several challenges remain, particularly regarding the secure data transfer. Additionally, the large volume of real-time data produced by vehicles, and IoT devices, which creates a significant burden on servers, that require efficient solutions.

Another crucial challenge to consider is that of emergency situations, such as rural or disaster-stricken regions. Providing real-time information to users in these situations remains a major challenge to overcome. Fortunately, drones are expected to be used for restoring emergency communication in critical situations, as they are not confined to road networks. However, the challenge of establishing secure real-time communication between drones and users, remains crucial.

At the same time, the introduction of multiple drones forming a swarm complicates the application of the FAA's new RemoteID regulations introduced in 2022, which require each drone to broadcast its identity. To prevent the real identities of the drones in the swarm from being disclosed, it is crucial to propose a protocol that allows the drones to authenticate themselves anonymously. In light of these challenges, another major issue is that traditional security methods cannot be effectively applied to the IoV and the IoD (Internet of Drones) due to their high resource consumption.

In this thesis, we propose authentication solutions specifically designed to address these challenges. The first contribution focuses on improving a recently published authentication protocol. Our cryptanalysis revealed a major vulnerability regarding message confidentiality, allowing an attacker to deduce the session key shared between the vehicle and the other entities (user and data center). Our improvement addresses this vulnerability, making the protocol applicable to IoV. Another improvement involves the enhancement of a another published protocol, adapted to big data collection and systems requiring fast responses. The protocol eliminates direct communication between the user and the data center, aiming to reduce costs. Our cryptanalysis allowed us to correct a flaw affecting user authentication.

The second contribution relates to the secure integration of drones, where we proposed a new authentication protocol named LASeR, designed for interactions between users and drones. LASeR stands out for its zone-based access, ensuring continuous monitoring while offering an optimal compromise between cost and security.

The third contribution addresses the challenge of drone swarms, for which we proposed a new authentication protocol named 2AS-DS, compliant with FAA regulations. A security analysis shows that 2AS-DS withstands attacks such as interception, and cloning, while reducing the load on the trust station.

In conclusion, the solutions proposed in this thesis actively contribute to enhancing the security of communications in the IoT, with a specific focus on mobile objects, such as vehicles and drones, which play a central role in the global technological revolution.

Mots clés : *Internet of things, Security protocol, Computer attack, Authentication, Lightweight cryptography, cryptanalysis, Intelligent Transportation Systems*

Résumé

La croissance rapide de IoT (Internet des objets) et des technologies de communication mobile offre des avancées significatives dans divers domaines. En particulier dans les zones urbaines, ces technologies apportent des solutions innovantes au problème de la congestion routière. Les systèmes IoV modernes (Internet des véhicules) permettent une interaction fluide entre les utilisateurs, les véhicules, les dispositifs IoT et les plateformes de services, dans le but d'améliorer l'efficacité des transports et la sécurité routière. Cependant, plusieurs défis persistent, notamment en ce qui concerne le transfert sécurisé des données. De plus, le volume important de données en temps réel produites par les véhicules et les dispositifs IoT crée une charge significative sur les serveurs, qui nécessitent des solutions efficaces.

Un autre défi crucial à considérer est celui des situations d'urgence, telles que les régions rurales ou sinistrées. Fournir des informations en temps réel aux utilisateurs dans ces situations reste un défi majeur à surmonter. Heureusement, l'utilisation des drones pour rétablir la communication d'urgence dans les situations critiques s'avère prometteuse, car ils ne dépendent pas des infrastructures routières. Cependant, le défi de mettre en place une communication sécurisée en temps réel entre les drones et les utilisateurs, reste crucial.

Parallèlement, l'introduction de plusieurs drones formant un essaim, complique l'application des nouvelles régulations RemoteID de la FAA, introduites en 2022. En effet, ces régulations exigent que chaque drone diffuse son identité. Afin d'éviter la divulgation des identités réelles des drones de l'essaim, il est important de proposer un protocole permettant aux drones de s'authentifier de manière anonyme. Face à ces défis, un autre problème majeur est que les méthodes de sécurité traditionnelles ne peuvent pas être appliquées efficacement à IoV et IoD (Internet des Drones) en raison de leur consommation élevée de ressources.

Dans cette thèse, nous proposons des solutions d'authentification spécifiquement conçues pour relever ces défis. La première contribution porte sur une amélioration d'un protocole d'authentification récemment publié. Notre cryptanalyse a révélé une vulnérabilité majeure concernant la confidentialité des messages, permettant à un attaquant de déduire la clé de session partagée entre le véhicule et les entités (utilisateur/centre de données). Notre amélioration corrige cette vulnérabilité, rendant ainsi le protocole applicable à l'IoV. Une autre amélioration concerne l'amélioration d'un autre protocole publié, adapté à la collecte de big data et aux systèmes nécessitant des réponses rapides. Le protocole élimine la communication directe entre l'utilisateur et le centre de données, dans le but de réduire les coûts communications. Notre cryptanalyse a permis de corriger une faille affectant l'authentification des utilisateurs.

La deuxième contribution concerne l'intégration sécurisée des drones, nous avons proposé un nouveau protocole d'authentification nommé LASeR, destiné aux interactions entre les utilisateurs et les drones. LASeR se distingue par son accès basé sur des zones géographiques, assurant une surveillance continue

tout en offrant un compromis optimal entre coût et sécurité.

La troisième contribution concerne le défi des essaims de drones, pour lequel nous avons proposé un nouveau protocole d'authentification, nommé 2AS-DS, conforme aux réglementations de la FAA. Une analyse de sécurité montre que 2AS-DS résiste aux attaques telles que l'interception et le clonage, tout en réduisant la charge sur la station de confiance.

En conclusion, les solutions proposées dans le cadre de cette thèse participent activement à l'amélioration de la sécurité des communications dans IoT, en se concentrant spécifiquement sur les objets mobiles, tels que les véhicules et les drones, qui jouent un rôle central dans la révolution technologique à l'échelle mondiale.

Mots clés : *Internet des objets, Protocole de sécurité, attaque informatique, Authentification, Cryptographie légère, Cryptanalyse, Systèmes de transport intelligents*

Table des matières

Table des matières **vii**

Table des figures **xiii**

Liste des tableaux **xv**

Liste des abbréviations et acronymes **xvii**

1 Introduction **1**

1.1 De l'Internet des objets vers l'Internet des drones 1

1.2 Éléments de la problématique et défis scientifiques 3

1.2.1 Éléments de la problématique 3

1.2.2 Défis scientifiques 4

1.3 Objectifs de recherche 4

1.4 Principales contributions de la thèse et leur originalité 5

1.5 Plan de la thèse 7

2 Revue de la littérature **9**

2.1 Prérequis de la cryptographie 10

2.1.1 Fonctions de hachage cryptographiques 10

2.1.1.1 Propriétés clés des fonctions de hachage : 10

2.1.1.2 Application de la fonction de hachage en cryptographie 10

2.1.2 Système cryptographie à courbe elliptique (ECC) 11

2.1.3 Échange de clés Diffie-Hellman (D-H) 12

2.1.3.1 D-H et les propriétés du système de distribution de clés publiques . 12
2.1.3.2 La sécurité du mécanisme de distribution de clés publiques de Diffie-Hellman 12
2.1.4 Cryptographie à base d'identité 12
2.1.5 Fonction physique non clonable (PUF) 13
2.1.5.1 Principe et métriques d'évaluation des PUF 13
2.1.5.2 Classification des PUF 14
2.1.5.3 PUF pour l'authentification [10, 11] 15
2.2 Problèmes de sécurité et d'authentification dans un environnement IoV/IoD. 17
2.2.1 Les protocoles d'authentification existants dans l'IoV 17
2.2.1.1 Analyse de la Sécurité des protocoles existants dans l'IoV : 25
2.2.1.2 Analyse des Performances des protocoles existants dans l'IoV : . . 25
2.2.1.3 Synthèse sur la sécurité et la performance des protocoles existants dans l'IoV : 26
2.2.2 Les protocoles d'authentification existants dans l'IoD 29
2.2.2.1 Analyse de la Sécurité des protocoles existants dans l'IoD : 32
2.2.2.2 Analyse des Performances des protocoles existants dans l'IoD : . . 34
2.2.2.3 Synthèse sur la sécurité et la performance des protocoles existants dans l'IoD : 36

3 Démarche et méthodologie de l'ensemble du travail de recherche 38
3.1 Méthodologie pour la conception du protocole d'authentification dans IoV/IoD . . . 38
3.2 Volet 1 : Conception du protocole d'authentification pour IoV 39
3.2.1 Modèle de réseau de l'IoV 39
3.2.2 Les Modèles de menace : 39
3.2.3 Protocole d'authentification et ses défis 40
3.2.4 Technique cryptographique 40
3.2.5 Améliorations proposées dans IoV 41
3.2.6 Évaluation de la sécurité 41
3.2.7 Évaluation des performances 42
3.3 Volet 2 : Conception du protocole d'authentification pour IoD 42

3.3.1 Étude de cas dans l'IoD . . . 43
3.3.2 Évaluation de la sécurité . . . 44
3.3.3 Outils de Vérification de Sécurité : AVISPA, Scyther et Tamarin . . . 44
3.3.4 Évaluation des performances . . . 46
3.4 Conclusion . . . 47

4 Analyse de la sécurité et amélioration du protocole d'authentification à trois facteurs utilisant fonction physique non clonable pour l'Internet des véhicules. 48

4.1 Introduction . . . 48
4.2 Le modèle de système . . . 49
4.3 Revue du Schéma de Jiang et al. . . . 51
4.3.1 Configuration du Système . . . 51
4.3.2 Phase d'Enregistrement . . . 52
4.3.2.1 Enregistrement de l'Utilisateur . . . 52
4.3.2.2 Enregistrement des Capteurs de Véhicules . . . 52
4.3.3 Phase de Connexion et d'Authentification . . . 53
4.4 Énoncé du Problème du Schéma de Jiang et al. . . . 54
4.4.1 Corrections au protocole de Q. Jiang et al. . . . 55
4.5 Cryptanalyse du protocole de Jiang et al. . . . 56
4.5.1 Modèle d'attaque . . . 56
4.5.2 L'attaque de déduction d'une clé de session entre le capteur du véhicule et le centre de données. . . . 56
4.5.3 L'attaque de déduction d'une clé de session entre le capteur du véhicule et l'utilisateur . . . 57
4.6 Amélioration du Schéma . . . 59
4.7 Analyse des Performances . . . 62
4.8 Conclusion . . . 62

5 Amélioration du schéma d'authentification des utilisateurs pour la collecte de Big Data dans un système de transport intelligent basé sur l'IoT. 64

5.1 Introduction . . . 64
5.2 Revues du schéma UAP-BCIoT . . . 66

5.2.1 Phase d'initialisation du système 66
5.2.2 Phase d'enregistrement des dispositifs IoT 67
5.2.3 Phase d'enregistrement de l'utilisateur 67
5.2.4 Phase de connexion et d'authentification 68
5.3 Énoncé du problème 68
5.4 Solution proposée 70
5.4.1 Solution côté Cloud-Gateway 70
5.4.2 Solution côté utilisateur 71
5.5 Analyse des performances 71
5.5.1 Comparaison des mesures de sécurité 71
5.5.2 Comparaison des coûts de calcul 72
5.5.3 Comparaison des coûts de communication 73
5.6 Conclusion 75

6 LASeR : Protocole d'authentification à distance léger et sécurisé pour l'Internet des Drones 76

6.1 Introduction 76
6.1.1 Contribution 77
6.2 LASeR : Modèle de Système 79
6.2.1 Modèle de Réseau 79
6.2.2 Modèle de Menace 79
6.3 Le Protocole LASeR Proposé 80
6.3.1 Phase de Pré-déploiement 81
6.3.2 Phase d'Enregistrement de l'Utilisateur (UR) 81
6.3.3 Phase de Connexion et d'Authentification (LA) 82
6.3.4 Validité des Clés de Session 83
6.3.5 Phase de mise à jour du mot de passe et/ou des données biométriques (PU) . 85
6.3.6 Phase de révocation et de ré-enregistrement de l'utilisateur (RR) 85
6.4 Analyse de sécurité 87
6.4.1 Vérification formelle : AVISPA 87
6.4.2 Vérification formelle : Scyther 88

6.4.3 Vérification formelle : Tamarin 89
6.4.4 Analyse de sécurité informelle 90
6.4.4.1 Authentification mutuelle 90
6.4.4.2 Attaque par rejeu 91
6.4.4.3 Attaques d'usurpation d'identité (IA) 91
6.4.4.4 Attaque de l'homme du milieu 91
6.4.4.5 Intraçabilité et anonymat de l'utilisateur 92
6.4.4.6 Attaque de divulgation de secrets éphémères (ESL) 92
6.4.4.7 Attaque par vol de dispositif mobile et attaque de l'initié privilégié 92
6.4.4.8 Attaque par capture de drone 93
6.4.4.9 Attaque par déni de service (DoS) 93
6.4.4.10 Attaque par mise à jour de mot de passe et/ou de données biométriques 93
6.5 Analyse des performances 94
6.5.1 Fonctionnalités et sécurité 94
6.5.2 Analyse des coûts computationnels 95
6.5.3 Analyse des coûts de communication 96
6.5.4 Coût énergétique computationnel 97
6.5.5 Coût énergétique de la communication 98
6.6 Conclusion 99

7 2AS-DS : Schéma d'authentification anonyme basé sur fonction physique non clonable pour les essaims de drones 100

7.1 Introduction 100
7.2 Modèle de Système 102
7.3 Le protocole 2AS-DS proposé 104
7.3.1 Initialisation du système 104
7.3.2 Phase d'enregistrement de l'essaim de drones 104
7.3.3 Authentification d'un essaim de drones 105
7.4 Analyse de sécurité 109
7.4.1 Anonymat des drones 109
7.4.2 Non-traçabilité 109

7.4.3 Résilience aux attaques de masquerade et de type homme du milieu 109
7.4.4 Résilience aux attaques de rejeu et de clonage 109
7.5 Analyse comparative des mesures de sécurité 110
7.6 Conclusion 110

8 Conclusion **112**
8.1 Synthèse des travaux 112
8.2 Limitations des travaux réalisés 113
8.3 Orientations de recherches futures 115

Bibliographie **117**

Annexe A Liste des publications **126**

Table des figures

1.1 Utilisation des drones dans le domaine des transports 2

1.2 Vue d'ensemble des contributions de la thèse 6

2.1 Protocole d'authentification basé sur une PUF utilisant un mécanisme de défi-réponse 16

3.1 Processus de conception d'un protocole d'authentification pour l'IoV/IoD 39

3.2 Techniques de vérification de la sécurité 42

4.1 Modèle de système et phase d'authentification 50

4.2 Illustration de la dérivation d'une clé de session entre le capteur du véhicule et le centre de données dans le schéma de Jiang et al. 58

4.3 Illustration de la dérivation d'une clé de session entre le capteur du véhicule et l'utilisateur dans le schéma de Jiang et al. 59

4.4 Phase d'authentification de notre protocole améliorée 61

5.1 Modèle de réseau du schéma UAP-BCIoT 66

5.2 Phase d'authentification du schéma UAP-BCIOT 68

6.1 Modèle de réseau (architecture) du protocole LASeR proposé 78

6.2 Phase d'enregistrement des utilisateurs 81

6.3 Phase de connexion et d'authentification du LASeR proposé. 84

6.4 Phase de mise à jour du mot de passe et/ou des données biométriques. 86

6.5 Phase de révocation et de réenregistrement de l'utilisateur 87

6.6 Résultats de la simulation avec les backends CL-AtSe et OFMC. 88

6.7 Résultats de l'analyse de sécurité de LASeR avec Scyther. 89
6.8 Résultats de l'analyse de sécurité de LASeR avec Tamarin. 90
6.9 Comparaison du coût computationnel pour compléter la phase d'authentification. . . 96
6.10 Comparaison du coût de communication entre Ui, GSS et Dj 97
6.11 Comparaison de la consommation d'énergie liée à la communication. 99

7.1 Modèle de réseau (architecture) du protocole 2AS-DS proposé 103
7.2 Phase d'enregistrement de l'essaim de drones. 105
7.3 Diagramme de séquence pour l'authentification d'un essaim de drones. 106
7.4 Authentification, accord de clé et demande de liste des drones. 107

8.1 Modèle système de recharge sans fil pour véhicules électriques dans un segment routier 115

Liste des tableaux

2.1 Synthèse des questions de recherche sur l'état de l'art. 18
2.2 Mécanismes d'authentification existants pour l'IoV : étude comparative 24
2.3 Opérations cryptographiques et leurs coûts en temps pour l'IoV [28] 26
2.4 Comparaison des mécanismes d'authentification pour l'IoV : fonctionnalité et sécurité 27
2.5 Comparaison des mécanismes d'authentification pour l'IoV : analyse des coûts de calcul 28
2.6 Mécanismes d'authentification existants pour l'IoD : étude comparative 31
2.7 Comparaison des mécanismes d'authentification pour l'IoD : fonctionnalité et sécurité 33
2.8 Temps d'opérations cryptographiques sur le drone et le serveur 34
2.9 Comparaison des mécanismes d'authentification pour l'IoD : analyse des coûts de calcul 35
2.10 Comparaison des mécanismes d'authentification pour l'IoD : analyse des coûts de communication 36

4.1 Notations utilisées dans notre protocole amélioré pour l'IoV 51
4.2 Comparaison des coûts de calcul entre le protocole de Jiang et al. et notre amélioration 62

5.1 Notations utilisées dans le protocole UAP-BCIOT 67
5.2 Comparaison de la sécurité et des fonctionnalités 72
5.3 Temps d'exécution pour diverses opérations utilisées par UAP-BCIoT 73
5.4 Performance Comparison 74

6.1 Liste des notations du protocole LASeR proposé. 80
6.2 Comparaison de la sécurité et des fonctionnalités du protocole LASeR par rapport aux protocoles existants 94

6.3 Le temps d'exécution des opérations utilisées pour évaluer LASeR et les protocoles existants. 95
6.4 Comparaison des coûts computationnels et communicationnels du protocole LASeR par rapport aux protocoles existants. 95
6.5 Comparaison de la consommation d'énergie computationnelle (μJ) 97
6.6 Comparaison du coût de communication au niveau du drone (bits) 98

7.1 Notations pour le mécanisme d'authentification des essaims de drones. 104
7.2 Évaluation des performances en fonction des exigences de sécurité 110

Liste des abbréviations et acronymes

2AS-DS Anonymous Authentication Scheme for Drone Swarms.
6LoWPAN IPv6 over Low Power Wireless Personal Area Networks.

AES Advanced Encryption Standard.
AVISPA Automated Validation of Internet Security Protocols and Applications.

DDoS Distributed Denial of Service.
DoS Denial of Service.

ECC Elliptic Curve Cryptography.
ESL Ephemeral Secret Leakage.

IoD Internet of Drones.
IoE Internet of Everything.
IoT Internet of Things.
IoV Internet of Vehicles.
ITS Intelligent Transportation System.

LASeR Lightweight and Secure Remote User Authentication Protocol for Internet of Drones.

MITM Man-in-the-Middle.

PKC Public Key Cryptography.
PKG Private Key Generator.
PKI Public Key Infrastructure.
PUF Physical Unclonable Function.

RSA Rivest-Shamir-Adleman (encryption algorithm).
RSU Roadside Unit.

TLS Transport Layer Security.

UAP-BCIoT User Authentication Protocol for Big Data Collection in IoT-based Intelligent Transportation System.
UAV Unmanned Aerial Vehicle (Drone).

V2G Vehicle-to-Grid Communication.
V2I Vehicle-to-Infrastructure Communication.
V2P Vehicle-to-Pedestrian Communication.
V2V Vehicle-to-Vehicle Communication.
V2X Vehicle-to-Everything Communication.

WSN Wireless Sensor Networks.

Chapitre 1

Introduction

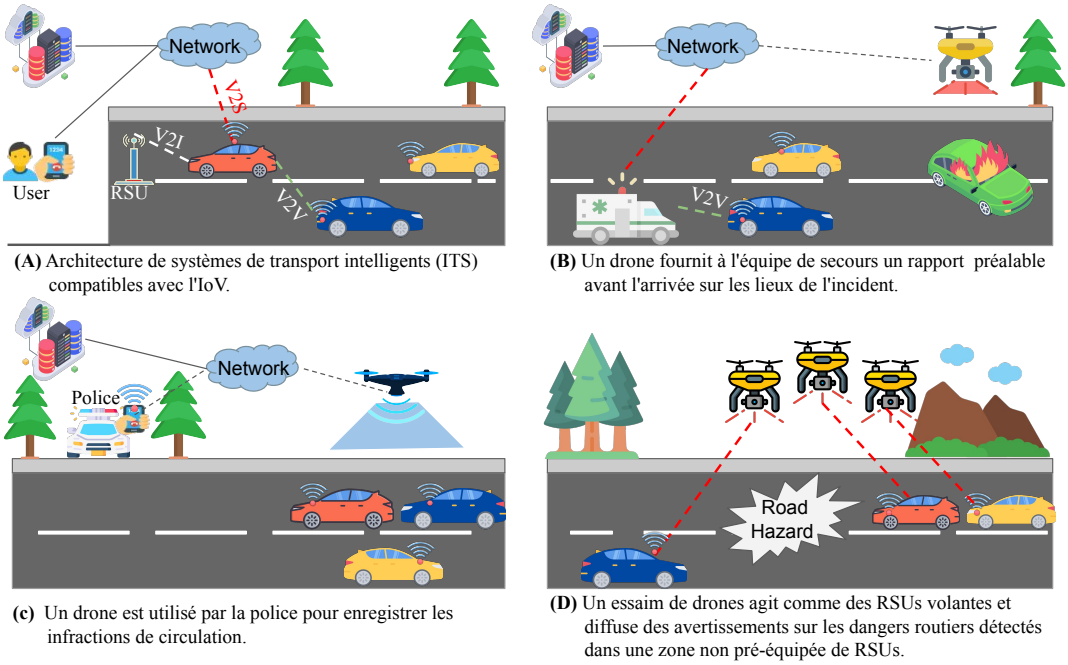
1.1 De l'Internet des objets vers l'Internet des drones

Ces dernières années, les progrès dans l'Internet des objets (IoT) et d'autres technologies ont amélioré la communication entre les appareils intelligents, les capteurs et diverses entités, ouvrant la voie à des applications innovantes dans divers domaines, tels que les villes intelligentes et les transports. L'industrie moderne de la fabrication automobile et les avancées dans les technologies de capteurs et de communication sans fil ont conduit à l'émergence de l'Internet des véhicules (IoV). Le concept de l'IoV vise à créer une infrastructure interconnectée pour l'échange d'informations et de ressources entre véhicules intelligents, ce qui facilitera le développement des systèmes de transport intelligents (ITS). Grâce à l'IoV, l'ITS impliquera un nombre croissant d'automobiles connectées, intelligentes et permettra une connectivité continue entre les véhicules, les infrastructures routières et les piétons. Cette infrastructure interconnectée apportera des avantages tels qu'une sécurité routière améliorée, une conduite plus sûre, une circulation plus fluide et une meilleure gestion du stationnement. Cependant, les véhicules manquent généralement de ressources embarquées suffisantes pour le traitement des données, ce qui nécessite de s'appuyer sur des serveurs cloud via des unités en bord de route (RSU) pour des tâches telles que la vérification d'identité et le traitement des données en temps réel. Par ailleurs, des défis subsistent dans la mise en œuvre des IoV, notamment en ce qui concerne l'allocation des ressources de routage, les interruptions/déconnexions des services réseau, l'authentification des messages, la diffusion du canal sécurisé et l'impact des coûts élevés. Ces problématiques exigent de trouver des solutions novatrices pour garantir un bon fonctionnement et une sécurité optimale des réseaux IoV.

L'IoV a évolué pour englober non seulement les véhicules, mais également les véhicules aériens sans pilote (UAV) ou les drones. Cette évolution a donné naissance à l'IoD (Internet des Drones), un sous-ensemble de l'IoV qui se concentre sur les interactions, la connectivité et les échanges entre les drones et d'autres appareils IoT. Notamment, dans les zones à intersections complexes et dans

les zones rurales ou sinistrées sans RSU, les drones jouent un rôle vital en tant que plateformes de communication intermédiaires. Grâce à leur capacité à survoler des restrictions terrestres et à étendre la couverture réseau, ils garantissent la continuité des services de communication, ce qui renforce la robustesse et la résilience des réseaux.

L'Émergence de l'Internet des Drones (IoD) dans l'Évolution de l'Internet des Véhicules (IoV) De nouveaux systèmes de communication et de navigation sont intégrés aux transports traditionnels, formant l'Internet des véhicules (IoV), où toutes les entités, véhiculaires ou statiques, peuvent échanger des informations comme la vitesse et la localisation. Par rapport à l'IoT, l'IoV est plus adapté pour des services fiables dans les villes intelligentes, en combinant les capacités de l'IoT avec les réseaux véhiculaires, créant ainsi un grand réseau de dispositifs connectés. L'IoV permet des communications véhicule-à-everything (V2X), classées en inter-véhicules et intra-véhicules. Les inter-véhicules incluent véhicule-à-véhicule (V2V), véhicule-à-piéton (V2P), véhicule-à-infrastructure (V2I), véhicule-à-réseau (V2G) et véhicule-à-unités en bord de route (V2R). Les intra-véhicules incluent véhicule-à-capteurs (V2S). Les défis de mise en œuvre des IoV sont particulièrement marqués dans les zones à intersections complexes, les zones rurales. De plus, pour une automatisation complète du système de transport, il est essentiel d'automatiser aussi les équipes de soutien, la police de la circulation, les enquêtes routières et les équipes de secours, ce qui peut être réalisé grâce à des drones intelligents, comme le montre la figure 1.1.



(A) Architecture de systèmes de transport intelligents (ITS) compatibles avec l'IoV.

(B) Un drone fournit à l'équipe de secours un rapport préalable avant l'arrivée sur les lieux de l'incident.

(c) Un drone est utilisé par la police pour enregistrer les infractions de circulation.

(D) Un essaim de drones agit comme des RSUs volantes et diffuse des avertissements sur les dangers routiers détectés dans une zone non pré-équipée de RSUs.

FIGURE 1.1 : Utilisation des drones dans le domaine des transports

Les principales applications des drones dans le domaine des transports sont les suivantes :

Rapport d'accident : Les équipes de secours peuvent utiliser des drones pour accéder rapidement aux

lieux d'accidents.

Surveillance aérienne des forces de l'ordre : Les drones peuvent surveiller différents tronçons de route pour détecter et intercepter les véhicules en infraction. Ils sont capables de modifier les feux de signalisation pour contraindre un véhicule à s'arrêter ou d'envoyer un message direct au conducteur pour lui ordonner de se stopper.

Surveillance des conditions d'autoroute : Un essaim de drones peut se rendre à une zone précise pour réaliser une tâche spécifique. Par exemple, en cas d'accident sur une portion d'autoroute sans RSU, le centre de gestion du trafic peut envoyer un essaim de drones sur le lieu de l'accident pour diffuser des informations et avertir les véhicules approchant de l'incident.

Méthode de reconnaissance de comportement : Un essaim de drones peut détecter des comportements suspects ou anormaux des véhicules circulant sur la route.

Les drones sont également déployés pour des applications liées aux consommateurs, telles que la transmission de données, la surveillance/la collecte de données et la livraison de produits. De grandes entreprises (comme Google et Meta) ont exprimé leur volonté d'utiliser les drones pour étendre la couverture Internet. De même, les géants de la logistique (Amazon, DHL, etc.) ont également réalisé des avancées significatives dans la livraison de produits par drones [1]. Pour concrétiser ces applications, L'IoD utilise une architecture de coordination automatisée, ce qui permet d'optimiser l'utilisation de l'espace aérien, qui est à la fois vaste et limité. L'espace aérien est subdivisé en zones de vol, chacune étant contrôlée de manière centralisée par les autorités de l'aviation, assurant ainsi une gestion efficace et sécurisée des drones.

1.2 Éléments de la problématique et défis scientifiques

1.2.1 Éléments de la problématique

L'émergence de l'Internet des Objets (IoT) a favorisé le développement de nouvelles applications, telles que l'Internet des Véhicules (IoV) et l'Internet des Drones (IoD). Ces technologies offrent des possibilités d'intégration synergique dans divers secteurs, notamment les transports et les villes intelligentes. Cependant, ces applications présentent des exigences variées en termes de qualité de service et de sécurité. Les réseaux sans fil (réseaux ouverts) jouent un rôle essentiel dans la transmission de grandes quantités de données dans les systèmes à haute mobilité (IoV, IoD), ce qui en fait une cible privilégiée pour les cyberattaques. Par conséquent, les protocoles utilisés doivent être suffisamment flexibles pour répondre à ces besoins, tout en tenant compte des contraintes de puissance et de capacité de calcul des nœuds impliqués.

De plus, certaines applications nécessitent une communication ultra-fiable à faible latence, car des informations incorrectes ou retardées peuvent avoir des conséquences graves. Par exemple, dans le

domaine IoV, les véhicules doivent communiquer avec l'infrastructure pour prendre des décisions critiques. De même, les drones dans les applications IoD doivent souvent collecter et transmettre des informations cruciales pour des emplacements spécifiques. La précision et la fiabilité des données reçues sont donc essentielles pour les prises de décision.

L'adoption croissante des drones a également provoqué des inquiétudes importantes en ce qui concerne la sécurité et la confidentialité. Les situations où des drones amateurs pénètrent sans autorisation dans des zones sensibles, telles que les zones militaires et aéroportuaires, ont causé de graves problèmes de sécurité. En réaction, des mesures telles que l'Identification à Distance (RemoteID) ont été imposées par la Federal Aviation Administration (FAA) [62] et les autorités européennes afin de renforcer la sécurité. Cependant, ces réglementations ont aussi créé des débats, en particulier concernant la préservation de la vie privée des opérateurs de drones.

1.2.2 Défis scientifiques

Le progrès de l'IoV et de l'IoD pose de nombreux défis scientifiques majeurs. Ces défis sont particulièrement importants dans des environnements à haute mobilité, où les connexions sont continuellement établies et interrompues. En effet, les applications telles que la gestion du trafic des véhicules ou la surveillance continue d'une zone précise à l'aide de drones reposent sur une connexion ininterrompue. Comme l'échange d'informations dans des espaces publics, elles sont particulièrement vulnérables aux attaques, ce qui rend l'authentification et la génération de clés cruciales à chaque nouvelle connexion. Cette authentification constitue la première ligne de défense pour garantir l'authenticité des nœuds, et les clés générées sont utilisées pour assurer la confidentialité et l'intégrité des données.

En outre, les capteurs des véhicules et les drones sont des nœuds de ressources limitées, ce qui restreint l'utilisation de mécanismes de sécurité traditionnels. Cela souligne la nécessité de protocoles d'authentification légers, à la fois flexibles et capables de répondre à des exigences spécifiques en matière de sécurité, tout en tenant compte des contraintes de puissance de calcul, de communication et d'énergie des nœuds.

Enfin, la mise en œuvre de la réglementation RemoteID [62], bien qu'essentielle pour la sécurité, a provoqué des discussions sur la préservation de la vie privée. Il est donc essentiel de trouver un équilibre entre les exigences de sécurité et la préservation de la confidentialité, tout en respectant les réglementations en vigueur.

1.3 Objectifs de recherche

L'objectif principal de cette thèse est de créer, déployer et analyser des systèmes de sécurité pour l'Internet des objets mobiles, plus précisément IoV et IoD, tout en se focalisant sur la gestion des clés et l'authentification. Le but est d'examiner ces solutions, Sur la base des observations fournies, cette

thèse aborde trois principales questions de recherche, comme suit :

- Q1 :Comment concevoir des protocoles d'authentification et de gestion des clés sécurisés pour les applications IoV, tout en tenant compte des contraintes liées au coût de calcul, à la mémoire et à la communication ?

  De nombreux protocoles d'authentification et d'établissement de clés pour les applications IoV utilisent des mécanismes de sécurité traditionnels, ce qui est trop coûteux pour les nœuds IoV contraints en ressources. Il est donc nécessaire de formuler des solutions légères qui offrent une robustesse adéquate pour le processus d'établissement des clés dans IoV.

- Q2 : Comment les drones, lorsqu'ils sont intégrés dans les systèmes de transport intelligents, peuvent-ils établir une communication sécurisée avec l'utilisateur tout en tenant compte des contraintes énergétiques ?

  L'Internet des drones doit adopter les normes Internet pour la communication. Cependant, Les drones peuvent être incapables pour mettre en œuvre les mécanismes de sécurité Internet usuels pour ce type d'interactions. Cette question de recherche s'intéresse à l'intégration de schémas d'authentification et de gestion de clés légers pour sécuriser efficacement les réseaux de l'IoD. En particulier, la solution doit prendre en compte les particularités des drones et des réseaux, comme leur mobilité et le besoin de les remplacer lorsqu'ils atteignent la fin de leur cycle de vie. L'objectif est de garantir une surveillance continue d'une zone spécifique, en offrant un accès optimisé qui assure une couverture sans interruption.

- Q3 : Comment rendre les drones identifiables en temps réel par les autorités publiques pendant le vol tout en garantissant la confidentialité des utilisateurs de drones ?

  La Federal Aviation Administration (FAA) et les autorités européennes ont mis en place une réglementation connue sous le nom RemoteID (Remote Identification), qui impose à tous les drones de diffuser régulièrement leur identité. Toutefois, face aux inquiétudes des amateurs de drones concernant la confidentialité, il est crucial de trouver des solutions qui permettent d'authentifier les drones sans divulguer leur identité. L'objectif de Cette approche est de se conformer aux normes réglementaires tout en préservant la vie privée des utilisateurs.

1.4 Principales contributions de la thèse et leur originalité

Pour répondre aux trois questions de recherche posées, l'étude a été structurée en trois étapes principales. Tout d'abord, une étude de la littérature a été menée afin de comprendre les concepts de sécurité appliqués à l'IoV et l'IoD, ainsi que les protocoles d'authentification et de gestion des clés existants. Ensuite, des solutions de sécurité légères ont été conçues pour ces environnements, en mettant l'accent sur l'authentification et la gestion des clés. Enfin, ces solutions ont été évaluées en termes d'efficacité

et de sécurité. La figure 1.2 illustre la correspondance entre les contributions et les problématiques de recherche.

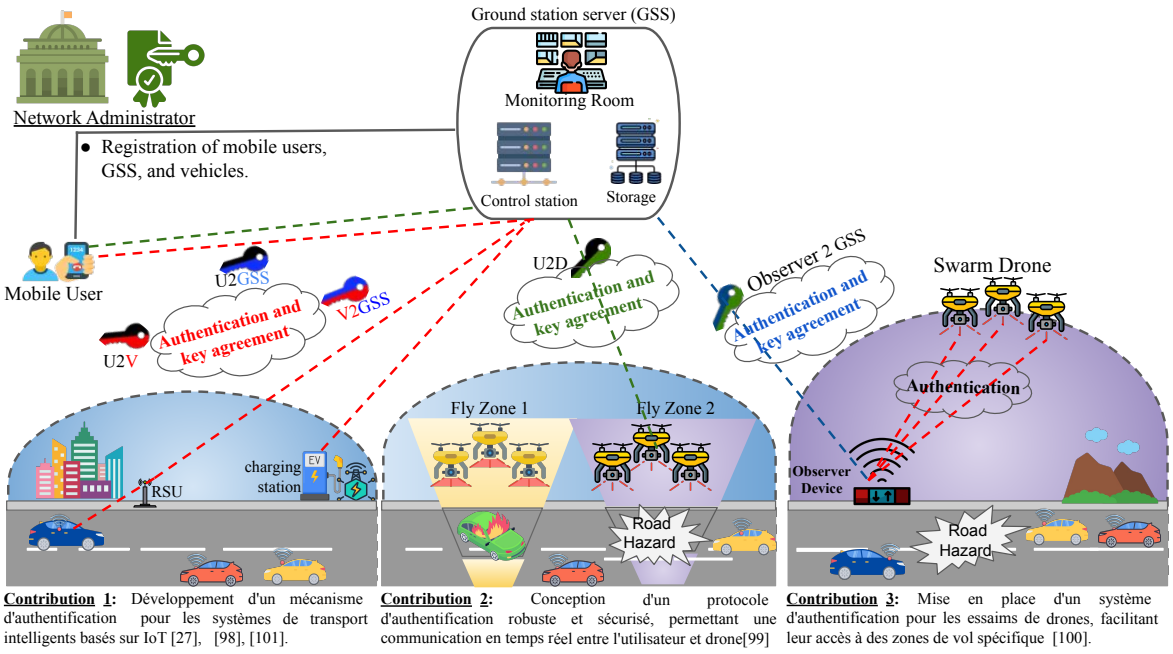


FIGURE 1.2 : Vue d'ensemble des contributions de la thèse

La première contribution de cette thèse réside dans la conception d'un protocole d'authentification et d'établissement de clés pour l'IoV [27]. Ce protocole vise à authentifier mutuellement les différentes entités, c'est-à-dire à vérifier l'identité de chaque acteur impliqué dans la communication (le véhicule, l'utilisateur et le centre de données). Cela garantit que chaque partie peut confirmer l'authenticité des autres avant d'établir une connexion sécurisée, réduisant ainsi les risques d'usurpation d'identité. L'authentification mutuelle permet d'assurer des échanges fiables et vérifiés entre le véhicule, l'utilisateur et le centre de données. En outre, le protocole génère des clés de cryptage distinctes pour chaque pair d'entités : une clé de cryptage pour sécuriser la communication entre le véhicule et l'utilisateur, une autre pour la communication entre le véhicule et le centre de données, et enfin une clé pour assurer la sécurité des échanges entre l'utilisateur et le centre de données. Ces clés légères sont optimisées pour garantir une communication sécurisée tout en préservant la performance.

La deuxième contribution porte sur l'authentification et la gestion des clés dans les communications entre drones et utilisateurs. Face aux contraintes énergétiques et aux besoins de surveillance en temps réel, nous avons conçu un protocole léger d'authentification nommé LASeR, spécifiquement adapté aux drones [99]. Ce protocole repose sur un accès basé sur des zones géographiques, garantissant une surveillance continue d'une zone spécifique tout en optimisant la consommation énergétique. Contrairement aux approches existantes, LASeR permet d'établir une communication sécurisée avec un coût réduit en termes de calcul et de transmission des données, ce qui le rend compatible avec les

drones ayant des ressources limitées. Une analyse de sécurité approfondie, incluant des vérifications formelles avec des outils comme AVISPA, Scyther et Tamarin, a permis de démontrer la robustesse du protocole face aux attaques courantes.

Enfin, la troisième contribution s'attaque au défi d'authentifier les drones en temps réel tout en préservant leur anonymat. Pour répondre aux exigences de la réglementation RemoteID imposée par la FAA, nous avons développé un protocole d'authentification anonyme pour les essaims de drones, nommé 2AS-DS [100]. Ce protocole permet aux drones de prouver leur identité sans révéler d'informations sensibles aux attaquants potentiels, tout en permettant aux autorités de vérifier l'authenticité des drones en cas de besoin. Grâce à une analyse de sécurité détaillée, nous avons démontré que 2AS-DS résiste aux attaques d'interception, de clonage et de type "man-in-the-middle", tout en réduisant la charge sur la station de confiance.

En résumé, les solutions proposées dans cette thèse visent à renforcer la sécurité des communications dans l'IoT, en mettant un accent particulier sur les objets mobiles tels que les véhicules et les drones, qui jouent un rôle clé dans la révolution technologique mondiale.

1.5 Plan de la thèse

Le travail de cette thèse s'articule autour de plusieurs parties, pour objectif principal d'aborder de manière approfondie les problématiques de sécurité dans les environnements IoV et IoD, tout en proposant des solutions originales en matière d'authentification. Le plan détaillé est le suivant :

Chapitre 2 propose une discussion des concepts cryptographiques fondamentaux. Il examine et analyse les travaux existants, en les divisant en deux volets : les mécanismes d'authentification appliqués à l'IoV et à l'IoD. Pour chaque volet, les différentes techniques cryptographiques sont comparées, avec une attention particulière portée sur leurs avantages/limitations en termes de sécurité, leur résistance aux attaques, ainsi que leur impact en termes de performances, notamment les coûts en communication et en calcul.

Chapitre 3 détaille la méthodologie adoptée tout au long de cette thèse. Il établit la corrélation entre les objectifs de recherche définis dans la section 1.3 et les contributions scientifiques issues de cette thèse, permettant ainsi de mieux comprendre le parcours scientifique et méthodologique de l'ensemble de l'étude. Cette méthodologie repose sur deux volets principaux : l'authentification dans l'Internet des véhicules (IoV) et l'Internet des drones (IoD).

Chapitre 4 présente la première contribution : 'Security Analysis on Three-Factor Authentication Protocol Using Physical Unclonable Function for IoV'. Cette étude porte sur la cryptanalyse et l'amélioration d'un protocole d'authentification récemment publié, impliquant trois entités (utilisateur, centre de données et véhicule). Après notre cryptanalyse, nous avons révélé une vulnérabilité majeure

concernant la confidentialité des messages, permettant à un attaquant de déduire la clé de session partagée entre le véhicule et les autres entités.

Chapitre 5 propose une amélioration d'un autre protocole publié, adapté aux besoins des systèmes de transport intelligents basés sur l'IoT, particulièrement ceux nécessitant une collecte de données massives (big data) et des réponses en temps réel. L'article présenté est intitulé "Enhancement of a User Authentication Scheme for Big Data Collection in IoT-Based Intelligent Transportation System".

Chapitre 6 présente la deuxième contribution intitulée 'LASeR : Lightweight and Secure Remote User Authentication Protocol for Internet of Drones', où nous proposons un nouveau protocole d'authentification et d'échange de clés léger et sécurisé. Ce protocole est conçu pour sécuriser les interactions entre les utilisateurs et les drones déployés dans des zones spécifiques, tout en garantissant une efficacité en termes de performances.

Chapitre 7 introduit la troisième contribution avec l'article '2AS-DS : Anonymous Authentication Scheme Based on Physical Unclonable Function for Drone Swarms'. Dans ce travail, nous proposons un protocole d'authentification anonyme conforme aux réglementations de la FAA, garantissant à la fois la confidentialité, l'intraçabilité des communications, et la conformité avec les exigences de sécurité des essaims de drones.

Chapitre 8 Le chapitre 8 présente une synthèse détaillée des travaux réalisés tout au long de la thèse. Il commence par récapituler les contributions principales et les résultats obtenus, en mettant en lumière l'impact de ces travaux dans le domaine de l'IoV et de l'IoD. Ensuite, les limitations des recherches menées sont exposées, afin de souligner les défis non résolus et les aspects qui pourraient être améliorés dans des travaux futurs. Enfin, ce chapitre propose des pistes pour de futures recherches, visant à approfondir les résultats obtenus et à explorer de nouvelles orientations susceptibles d'apporter des solutions innovantes.

Chapitre 2

Revue de la littérature

Dans ce chapitre, nous explorerons une revue de la littérature essentielle pour la compréhension des concepts cryptographiques et de leurs applications dans la conception de protocoles d'authentification dans des environnements spécifiques tels que l'Internet des Véhicules (IoV) et l'Internet des Drones (IoD). En effet, la cryptographie joue un rôle clé dans la protection des données, en assurant la confidentialité et l'intégrité des informations échangées. Elle est essentielle non seulement pour sécuriser les données sensibles, mais aussi pour garantir l'authentification des utilisateurs et des appareils dans des environnements complexes, tout en répondant aux exigences strictes de performance et de gestion des ressources.

Ce chapitre se structure autour de plusieurs sections. Tout d'abord, nous définissons les différentes techniques cryptographiques essentielles, telles que les fonctions de hachage, les systèmes cryptographiques à courbe elliptique (ECC), et l'échange de clés Diffie-Hellman. Ces mécanismes sont cruciaux pour la sécurité des échanges. Ensuite, nous aborderons des concepts avancés comme l'utilisation des fonctions physiques non clonables (PUF), qui offrent des solutions innovantes pour l'authentification dans les réseaux IoV/IoD. Ces techniques permettent de renforcer la sécurité tout en minimisant les risques de falsification.

La seconde partie de ce chapitre est divisée en deux volets principaux. Le premier volet est dédié aux protocoles d'authentification pour l'IoV, et le deuxième volet est consacré aux protocoles d'authentification pour l'IoD, en détaillant les problématiques de sécurité spécifiques à ces environnements. Nous mettrons en lumière les protocoles existants, leur évaluation en termes de sécurité et de performances, ainsi que leur capacité à répondre aux besoins des systèmes IoV et IoD.

Ce chapitre s'achèvera par une synthèse des problèmes de sécurité et des défis d'authentification, permettant ainsi d'identifier, à travers cette revue, les meilleures pratiques et méthodologies pour la conception de protocoles robustes et performants, dans le but de renforcer la sécurité dans les réseaux IoV et IoD.

2.1 Prérequis de la cryptographie

2.1.1 Fonctions de hachage cryptographiques

Le hachage est un processus qui transforme des données en une chaîne fixe de caractères, appelée "haché" ou "empreinte", à l'aide d'une fonction mathématique spécifique, connue sous le nom de fonction de hachage.

$$H : \{0,1\}^* \to \{0,1\}^n$$

$$M \mapsto H(M)$$

Ce mécanisme est crucial pour l'authentification, car il est souvent utilisé lors de la transmission d'informations chiffrées pour assurer l'intégrité des données. Cet algorithme empêche les modifications non autorisées des informations transmises. Un algorithme de hachage fonctionne de manière similaire à un contrôle de somme (checksum). Cependant, contrairement à ce dernier, qui est conçu pour détecter des erreurs accidentelles, un hachage est destiné à identifier les altérations intentionnelles. Lorsqu'on applique un algorithme de hachage aux informations, une séquence courte de bits, appelée "haché", est générée. Le moindre changement dans le message original entraîne généralement une modification significative du haché, ce qui permet de détecter les modifications même minimes.

2.1.1.1 Propriétés clés des fonctions de hachage :

Déterminisme : Une fonction de hachage doit toujours produire la même sortie pour une entrée donnée.

Taille de sortie fixe : La sortie d'une fonction de hachage a une longueur fixe, quelle que soit la taille de l'entrée.

Rapidité : Le calcul du haché doit être rapide, même pour des entrées volumineuses.

Résistance aux collisions : Il doit être pratiquement impossible de trouver deux entrées différentes produisant le même haché.

Résistance à la préimage : Il doit être difficile, voire impossible, de retrouver l'entrée initiale à partir du haché.

Absence de clé : Contrairement aux algorithmes de chiffrement, une fonction de hachage ne repose pas sur une clé secrète.

2.1.1.2 Application de la fonction de hachage en cryptographie

Les fonctions de hachage sont des primitives cryptographiques essentielles qui permettent d'assurer l'intégrité des données. Contrairement aux algorithmes de chiffrement symétrique ou asymétrique, elles ne visent pas à chiffrer des informations mais à produire une empreinte unique d'un message

donné. Elles sont souvent utilisées en combinaison avec d'autres mécanismes cryptographiques pour garantir la sécurité des systèmes d'information. Voici quelques exemples d'utilisation des fonctions de hachage :

Stockage de mots de passe : Les mots de passe ne sont généralement pas stockés en clair dans les bases de données, mais sous forme de hachés pour éviter qu'ils soient directement exploitables en cas de fuite de données.

Signature numérique : Une fonction de hachage est appliquée à un message avant d'être signée avec un algorithme de chiffrement asymétrique. Cela permet de garantir l'intégrité du message sans avoir à chiffrer son contenu en entier.

Vérification de l'intégrité des fichiers : Des empreintes de fichiers (hashes) sont souvent utilisées pour vérifier qu'un fichier téléchargé ou stocké n'a pas été altéré.

Dérivation de clé (KDF - Key Derivation Function) : Certaines fonctions de hachage sont utilisées pour dériver des clés cryptographiques à partir d'un secret, notamment dans les protocoles de chiffrement.

2.1.2 Système cryptographie à courbe elliptique (ECC)

Cette partie présente le concept fondamental du système de cryptographie utilisant les courbes elliptiques (ECC). Koblitz et Miller [2, 3] ont proposé l'intégration des courbes elliptiques dans les systèmes cryptographiques en 1985 et 1987. Ce système repose sur la structure algébrique des courbes elliptiques dans des espaces finis. Le principal avantage de l'ECC est qu'il nécessite une clé de taille inférieure à celle d'autres méthodes, telles que RSA, tout en offrant une sécurité équivalente ou meilleure.

Une équation de courbe elliptique est définie sous la forme

$$E_p(a,b) : y^2 = x^3 + ax + b \pmod{p}$$

sur un corps fini premier F_p^*, où a et b appartiennent à F_p, avec $p > 3$ et $4a^3 + 27b^2 \neq 0 \pmod{p}$.

La sécurité des systèmes cryptographiques basés sur les courbes elliptiques est complexe à garantir, et cette recherche applique cette approche dans le cadre de ce schéma.

1. **Le problème du logarithme discret sur les courbes elliptiques (ECDLP)** : Étant donnés deux points P et Q sur $E_p(a,b)$, et avec $Q = s \cdot P$, il est difficile de déterminer l'entier $s \in F_p^*$.
2. **Le problème de Diffie-Hellman (CDLP)** : Étant donnés trois points P, $s \cdot P$, et $t \cdot P$ sur $E_p(a,b)$, où $s,t \in F_p^*$, il est difficile de calculer $(s \cdot t) \cdot P$ sur $E_p(a,b)$.

2.1.3 Échange de clés Diffie-Hellman (D-H)

L'échange de clés Diffie-Hellman (D-H) est un protocole spécifique pour l'établissement sécurisé de clés cryptographiques, proposé en 1976. Il permet aux parties d'échanger des messages pour créer une clé de session sur un canal non sécurisé, sans conditions préalables. Cette clé de session peut ensuite servir de clé symétrique pour chiffrer les échanges de messages ultérieurs. cette thèse présentera le problème D-H dans les prochaines sections.

2.1.3.1 D-H et les propriétés du système de distribution de clés publiques

D-H exploite les propriétés du système de distribution de clés publiques tel que décrit ci-dessous :

- Ce schéma permet d'établir une clé commune pour les deux parties de la communication.
- Les parties déterminent ensemble le contenu de la clé pour communiquer.
- Ce schéma repose sur le calcul d'indices dans un groupe fini.
- Seules les deux parties peuvent avoir connaissance de cette clé de session commune.

2.1.3.2 La sécurité du mécanisme de distribution de clés publiques de Diffie-Hellman

La sécurité du mécanisme de distribution de clés de Diffie-Hellman repose sur le Problème de Diffie-Hellman (DHP).

Problème de Diffie-Hellman (DHP) : Considérons x, y et z comme des valeurs inconnues dans $\mathbb{Z}_q$. À partir de $g^x \mod n$ et $g^y \mod n$, il faut déterminer $g^z \mod n$ et vérifier que $z = xy \mod n$.

2.1.4 Cryptographie à base d'identité

Shamir a introduit la cryptographie basée sur l'identité (IBC) en 1985. Dans ce système, la clé publique d'un utilisateur peut contenir des informations spécifiques. Le Générateur de Clés Privées (PKG) fournit la clé secrète à chaque utilisateur. Quand le nœud A veut envoyer un message au nœud B, il chiffre le message avec la clé publique de ce dernier. Le nœud B reçoit le texte chiffré et le déchiffre avec sa clé privée. Il peut aussi associer la clé publique du nœud B avec la clé privée du nœud A pour établir une clé de communication privée. Le problème de l'IBC réside dans le fait que le PKG distribue toutes les clés secrètes des nœuds, ce qui lui permet de connaître toutes ces clés. Ainsi, le PKG peut déchiffrer tous les messages et falsifier les communications, tout en ne garantissant ni confidentialité ni non-répudiation. Contrairement à une infrastructure traditionnelle de gestion de clés (PKI), l'IBC allège cette gestion. Lorsque le nœud A veut contacter le nœud B, il envoie son identifiant au PKG. Celui-ci le convertit en une valeur de hachage, puis en un point sur la courbe elliptique, formant ainsi la clé publique du nœud B. Le nœud B renvoie cette clé publique au nœud A. Celui-ci choisit une

valeur aléatoire x, et obtient une valeur de couplage en utilisant x, la clé publique du nœud B, et le paramètre public du PKG. Le nœud B calcule une valeur de couplage avec la clé secrète reçue du PKG. Si les valeurs sont identiques, les deux nœuds peuvent se faire confiance et utiliser cette valeur pour établir une clé de session, permettant ainsi une communication privée.

2.1.5 Fonction physique non clonable (PUF)

Les fonctions physiques non clonable (PUFs) sont des mécanismes légers permettant de générer une identité distincte pour chaque appareil, en se basant sur des caractéristiques physiques spécifiques apparues durant la fabrication, qui sont imprévisibles et impossibles à dupliquer. Ces PUFs représentent des outils clés pour renforcer la sécurité des systèmes matériels et ont attiré un intérêt croissant récemment.

2.1.5.1 Principe et métriques d'évaluation des PUF

Le terme *"Fonction Physique à Sens Unique"* a été introduit par Pappu en 2001 [4]. Dans [5], les chercheurs ont présenté le concept de *"Fonction Physique Aléatoire"*. Pour distinguer cette dernière de la Fonction Pseudo-Aléatoire, on utilise le terme *"Fonction Physiquement Non Clonable"* (PUF). Cette méthode de sécurité matérielle, fondée sur les variations inhérentes des objets physiques, se divise en deux catégories : les Objets Uniques et les Fonctions Physiquement Non Clonables (PUFs). Un PUF est un système complexe qui est simple à fabriquer mais impossible à reproduire, même si un attaquant dispose d'un accès physique ou du procédé de fabrication exact. Ce système, lorsqu'il reçoit une stimulation (entrée C), produit une réponse unique (sortie R_C).

Une fonction physique non clonable, ou PUF, se décrit formellement par $f(\mathrm{PUF})\,C \Rightarrow R$, où C représente les défis et R les réponses associées. Cette fonction $f(\mathrm{PUF})$ doit remplir certaines propriétés [12].

1. **Fiabilité :** Pour chaque défi, la réponse doit rester identique à différents moments, soit, pour tout c_i dans C, $\mathrm{PUF}_t(c_i)$ correspond à $\mathrm{PUF}_{t_0}(c_i)$, même si t_0 est postérieur à t.
2. **Unicité :** Aucune fonction ne doit être équivalente à $PUF(\cdot)$, c'est-à-dire que si pour chaque c_i dans C, $\mathrm{PUF}_t(c_i)$ est égal à $\mathrm{PUF}_{t_0}(c_i)$, alors $\mathrm{PUF}_t(\cdot)$ et $\mathrm{PUF}_{t_0}(\cdot)$ sont identiques.
3. **Unidirectionnalité :** $\mathrm{PUF}(\cdot)$ doit être non inversible, soit $\mathrm{PUF}_{t_0} \circ \mathrm{PUF}^{-1}$ n'est pas l'identité sur C.
4. **Évaluation facile :** $\mathrm{PUF}(\cdot)$ doit être simple à calculer, de telle sorte que pour tout c_i, $O(\mathrm{PUF}(c_i))$ soit constant.
5. **Réplique difficile :** Il doit être impossible de dupliquer $\mathrm{PUF}(\cdot)$, donc si $\mathrm{PUF}(\cdot)$ est égale à $\mathrm{PUF}_{t_0}(\cdot)$ pour tout c_i dans C, la complexité de créer $\mathrm{PUF}_{t_0}(\cdot)$ doit tendre vers l'infini.

6. **Prédiction complexe :** Les réponses de PUF$(\cdot)$ doivent être presque impossibles à prévoir, donc la complexité de prédire PUF(c_n) en utilisant des paires $(c, \text{PUF}(c))$ tend vers l'infini.

2.1.5.2 Classification des PUF

Le PUF peut être classifié selon les critères suivants :

1. **Classification selon les caractéristiques physiques de la construction :**

   (a) **PUF intrinsèques [6] :** Ces PUF contiennent une source cachée d'aléa, ce qui signifie que l'incertitude est naturellement intégrée au dispositif à cause des variations durant la fabrication, et ils sont évalués en interne. Les PUF intrinsèques sont distingués selon leur principe de fonctionnement, qu'ils reposent sur le retard ou sur la mémoire.

   - **PUF basés sur le retard :** L'évaluation aléatoire se fonde sur le temps de réponse d'un circuit numérique. Parmi ces PUFs se trouvent les PUFs à arbitre, les oscillateurs à anneau et les PUFs à glitch.
   - **PUF basés sur la mémoire :** L'évaluation aléatoire est déterminée par les propriétés des cellules de mémoire. Les PUFs de ce type incluent les SRAM, les PUFs en papillon, les Flip-Flop, etc.

   (b) **PUF non intrinsèques [7] :** Ces PUFs possèdent des caractéristiques aléatoires explicites et sont évalués de l'extérieur. L'évaluation interne est souvent préférée, car elle permet des actions complémentaires telles que le hachage, le chiffrement, etc., et aide à contrer les attaques par canaux auxiliaires et les attaques de l'homme du milieu.

2. **Classification selon la technologie employée pour l'implémentation [8] :**

   (a) **PUF électriques :** La variation aléatoire dépend de caractéristiques telles que la résistance, la capacité, la tension, la fréquence, etc. La majorité des PUFs électriques sont réalisés en silicium, car ils sont intégrés dans une puce de ce matériau.

   (b) **PUF non électriques :** Ces PUFs reposent sur des caractéristiques non électriques, telles que les propriétés optiques, magnétiques, acoustiques, etc.

3. **Classification selon le nombre de paires Défi-Réponse (CRP) [9]**

   (a) **PUF forts :** Ces PUFs supportent un nombre très élevé (exponentiel par rapport aux composants physiques d'un circuit) de CRPs, avec une spécification rendant impossible la lecture de tous les CRPs en un temps raisonnable. Cela empêche un adversaire de déduire des CRPs inconnus à partir d'une connaissance limitée de CRPs. En particulier, les PUF forts doivent résister à l'ingénierie inverse par modélisation via apprentissage automatique (ML), qui pourrait utiliser un sous-ensemble de CRPs pour entraîner un algorithme et modéliser le circuit. Une fois

qu'un modèle est établi pour un PUF, ses principales caractéristiques de sécurité, comme l'imprévisibilité et l'inclonabilité, sont compromises. Ce type de PUF convient bien pour l'identification de circuits et la génération de clés secrètes.

(b) **PUF faibles :** Ces PUFs prennent en charge un nombre restreint de CRPs, voire un seul. Ils sont principalement utilisés pour générer des clés secrètes cryptographiques nécessaires à des cas d'utilisation sécurisés. Cependant, ils ne sont pas idéaux pour l'identification de circuits, en raison de leur nombre limité de CRPs, ce qui les rend vulnérables aux attaques par rejeu et par canal auxiliaire. Selon la littérature, ce type de PUF est moins exposé aux attaques par modélisation .

(c) **PUF contrôlés :** Ces PUFs combinent des PUF forts avec une logique de contrôle (algorithme) pour encapsuler les entrées et sorties. Cette architecture peut bloquer les attaques par modélisation, sauf si les sorties des PUF forts sont découvertes, ce qui pourrait alors compromettre le PUF contrôlé. Certaines applications cryptographiques, comme l'authentification par carte à puce ou le stockage privé dans les ordinateurs, nécessitent que certains CRPs inconnus pour éviter le clonage du circuit PUF.

2.1.5.3 PUF pour l'authentification [10, 11]

Les PUFs réalisent deux objectifs en sécurité : la génération de clés et l'authentification

(a) **Génération de clés :** Les services de sécurité, comme la confidentialité ou l'intégrité des données, utilisent le chiffrement ou l'authentification des messages, nécessitant une clé secrète. Les PUFs faibles sont utilisés pour éviter de stocker des clés sur la puce, en employant la sortie des PUFs comme clé secrète. Le défi principal est l'absence de garantie que les mêmes sorties soient produites à partir des mêmes entrées, en raison du bruit et des variations environnementales. Pour pallier cela, des méthodes de correction d'erreurs sont appliquées, telles que les codes BCH, le canal binaire symétrique (BSC), ou encore la construction de syndrome. Les PUFs faibles peuvent donc générer des clés pour des applications cryptographiques, en deux étapes principales : la phase d'initialisation et celle de reconstruction.

- **Phase d'initialisation :** Le circuit PUF génère une réponse dont le code de correction est sauvegardé.
- **Phase de reconstruction :** La même sortie doit être reproduite. Le PUF est alimenté avec une entrée pour produire une sortie corrigée à l'aide du code de la phase d'initialisation. Pour les services nécessitant une clé, la sortie est hachée pour obtenir la longueur voulue pour l'utilisation cryptographique (AES, etc.).

(b) **Authentification :** La communication M2M, cruciale dans l'IoT, nécessite une vérification de l'identité des dispositifs pour établir la confiance. Les PUFs permettent l'authentification sans recourir à

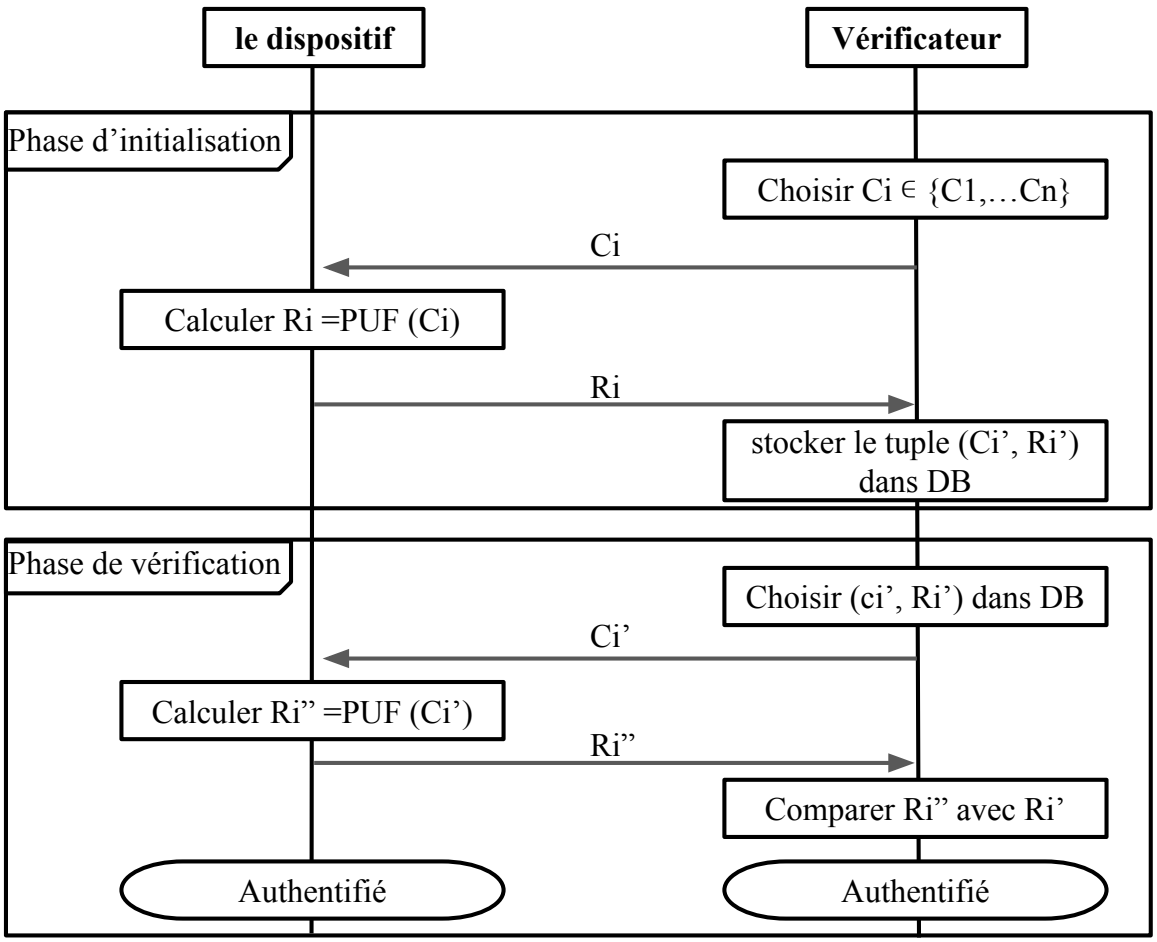


FIGURE 2.1 : Protocole d'authentification basé sur une PUF utilisant un mécanisme de défi-réponse

des protocoles cryptographiques coûteux. Cette méthode d'authentification, même dans des dispositifs à ressources limitées (comme RFID ou NFC), repose sur le protocole Défi/Réponse. Les PUFs offrent des réponses uniques aux défis, identifiant chaque dispositif de manière distincte. L'utilisation de PUFs forts complique la tâche d'un adversaire voulant cloner un dispositif.

Le processus d'authentification comporte deux étapes : l'initialisation et la vérification comme le montre la figure 2.1

- **Phase d'initialisation :** Avant l'utilisation des dispositifs, un ensemble de CRPs de chaque PUF est enregistré dans une base de données sécurisée.
- **Phase de vérification :** Lorsqu'une demande d'authenticité est effectuée, un ensemble de CRPs enregistrés est appliqué au dispositif. La réponse obtenue est comparée à celle de la base de données ; si elle correspond, le dispositif est authentifié, sinon, il est considéré comme un faux. Il est crucial de stocker un grand nombre de CRPs (PUF fort) pour éviter la réutilisation et la vulnérabilité aux attaques de type "man-in-the-middle"

2.2 Problèmes de sécurité et d'authentification dans un environnement IoV/IoD.

L'Internet des Véhicules et l'Internet des Drones sont deux technologies qui se complètent mutuellement pour former un écosystème intelligent (transport intelligent, ville intelligente, etc.). Le IoV, par exemple, repose sur une connexion continue pour des applications comme la conduite autonome, la gestion du trafic et la sécurité routière. Toutefois, dans les zones peu peuplées, l'implantation d'infrastructures de communication n'est pas toujours possible, ce qui restreint la couverture. Dans cette situation, les drones possèdent une importance cruciale en élargissant la portée des communications au-delà des infrastructures terrestres classiques. Par ailleurs, les systèmes IoV et IoD posent des défis importants en termes de sécurité et d'authentification. Il est possible que les systèmes IoV soient vulnérables à des attaques comme l'écoute passive et le compromis des véhicules, ce qui met en danger la sécurité des passagers. En revanche, les drones sont vulnérables à des risques tels que la prise de contrôle à distance ou les attaques par déni de service (DoS). Il est donc essentiel de garantir une gestion sécurisée des clés cryptographiques et d'assurer une authentification robuste afin de prévenir l'intrusion de dispositifs malveillants. La non-existence d'un tel mécanisme de sécurité peut provoquer des vulnérabilités exploitables. Il est essentiel de mettre en œuvre des mécanismes robustes afin d'éviter toute fuite d'informations. Afin d'explorer ces enjeux, une analyse de l'état de l'art sera réalisée en deux parties la première concerne les protocoles d'authentification liés à l'IoV, la seconde concerne les protocoles d'authentification liés à l'IoD. Chaque partie va répondre aux questions de recherche spécifiques. Le tableau 2.1 détaille les questions formulées et fournit la justification de chaque question. Chaque partie inclura une classification et une analyse approfondie des schémas.

2.2.1 Les protocoles d'authentification existants dans l'IoV

Dans cette section, nous résumons et discutons les travaux connexes sur les protocoles d'authentification pour l'Internet des véhicules. Jiang et al. [14] ont présenté une méthode d'authentification appelée ABAH, qui utilise une liste de révocation de certificats (CRL) pour vérifier les révocations et les signatures, ce qui a entraîné une augmentation de la surcharge, de la latence et du délai de transmission du réseau.

Jiang et al. [14] ont proposé une approche efficace pour améliorer la gestion et la sécurité des véhicules. Ils ont utilisé des pseudonymes et des signatures pour garantir la protection de la vie privée ainsi que l'authentification par lots. Cependant, ce schéma reposait sur un certificat délivré par une entité de confiance pour les unités en bord de route (RSU), ce qui entraînait un processus de gestion des certificats relativement complexe et augmentait les coûts de maintenance du système. Sutrala et al. [104] ont analysé l'approche de Jiang et al. [14] et ont conçu un protocole d'authentification de groupe sans certificat pour l'Internet des Véhicules (IoV). Ce protocole utilise l'algorithme de signature

N°Q	Question	Motivation
Q1	Quels types de problèmes sont présents dans l'authentification IoV/IoD ?	Reconnaître les types de problèmes dans l'authentification IoV/IoD.
Q2	Quelle est la contribution de l'authentification dans les systèmes IoV/IoD ?	Synthétiser les efforts de recherche, en mettant l'accent sur les thèmes communs dans les contributions de recherche.
Q3	Quels indicateurs de performance sont les plus couramment utilisés ?	Identifier les indicateurs de performance fréquemment utilisés spécifiques à l'authentification dans l'IoV/IoD.
Q4	Quels sont les avantages de chaque protocole d'authentification dans l'IoV/IoD ?	Mettre en avant les avantages offerts par les protocoles existants pour l'authentification et la sécurité de l'IoV/IoD.
Q5	Quels sont les défis d'authentification dans les environnements IoV/IoD ?	Mettre en lumière les limitations des travaux de recherche centrés sur l'authentification dans l'IoV/IoD.
Q6	Comment les avancées en matière d'authentification IoV/IoD répondent-elles aux défis collectifs dans l'environnement d'authentification IoV/IoD ?	Identifier les tendances et directions dans l'authentification IoV/IoD.

TABLE 2.1 : Synthèse des questions de recherche sur l'état de l'art.

numérique à courbe elliptique afin d'authentifier les véhicules à proximité en lots.

Wazid et al. [15] ont proposé un protocole décentralisé d'authentification légère et d'échange de clés. Ce protocole utilise uniquement une fonction de hachage cryptographique et des opérations XOR, et il s'appuie sur un modèle VANET basé sur des clusters. Ce protocole prend en charge trois types distincts d'authentification et d'échange de clés (véhicule-à-véhicule, véhicule-à-cluster et véhicule-à-RSU). Les clés secrètes sont également utilisées dans ce protocole pour protéger la communication. Cependant, ce protocole ne prend pas en compte la protection conditionnelle de la vie privée.

Ying et al. [16] ont proposé un protocole d'authentification léger et anonyme basé sur une carte à puce pour les réseaux véhiculaires, où l'authenticité des véhicules et la validation des messages de données sont toutes deux vérifiées à l'aide d'opérations cryptographiques à faible coût.

Mohit et al. [17] ont conçu un schéma sécurisé pour les systèmes véhiculaires intelligents basé sur les réseaux de capteurs sans fil (WSN) avec un faible coût de calcul et de communication. Cependant, le schéma ne résiste pas aux attaques de l'homme du milieu (MITM), à l'échange de clés de session, à l'authentification mutuelle sécurisée et à l'anonymat de l'utilisateur.

Zhou et al. [18] ont expliqué deux types différents de composants d'infrastructure et le schéma

d'authentification de l'infrastructure. Les véhicules sont équipés d'unités embarquées (OBU), tandis que les unités de communication routière (RSU) sont placées le long de la route en tant qu'infrastructure. Ce schéma peut être élargi, avec la conception d'un système d'authentification à seuil isolé fiable, ce qui représente une approche prometteuse pour garantir une communication V2I sécurisée en pratique. Néanmoins, il repose sur la discussion des auteurs concernant la fiabilité de tous les dispositifs et de la clé privée produite par son assistant. En général, les réseaux ouverts à haute mobilité, rendant l'adaptation plus complexe.

Chen et al. [19] ont proposé un autre protocole d'authentification pour assurer la sécurité de l'Internet des véhicules (IoV) lors de la transmission des données. De plus, il a été découvert que le schéma était vulnérable aux attaques replay, aux attaques par usurpation de position et aux attaques par usurpation d'identité. Le temps d'authentification était également long.

Wang et al. [20] proposent un schéma d'authentification véhicule-à-véhicule basé sur la gestion de la fiabilité de voisinage, où seuls les véhicules proches de le lieu de l'accident sont autorisés à signaler l'événement. Après réception des rapports par le RSU, ce dernier effectue une demande au serveur cloud pour vérifier la fiabilité du véhicule rapportant.

Ma et al. [21] ont proposé un schéma d'authentification sécurisée et d'échange de clés pour un système véhiculaire intelligent basé sur le fog, sans appariements bilinéaires, ce qui le rend extrêmement efficace sur le plan computationnel. Le schéma ne réalise pas la négociation sécurisée des clés de session ni l'authentification mutuelle, mais il supporte néanmoins la préservation de la vie privée.

Vasudev et al. [22] ont proposé un schéma d'authentification léger entre les véhicules, l'autorité de confiance et les serveurs de véhicules dans un environnement IoV. Leur schéma est léger, car il utilise simplement une fonction de hachage. Malheureusement, nous avons découvert que leur schéma est vulnérable aux attaques de "fuite de secret éphémère (ESL)", aux attaques par rejeu, aux "attaques par vol de carte à puce" et aux "attaques par usurpation d'identité".

Jiang et al. [25] ont proposé un mécanisme d'authentification à trois facteurs basé sur des PUF a été proposé pour l'Internet des véhicules. Ce protocole, solide et extrêmement léger, offre la possibilité de créer trois clés de session entre les acteurs clés : le véhicule, l'utilisateur et le centre de données. Cependant, après une analyse approfondie, nous avons identifié une vulnérabilité majeure dans ce protocole, le rendant vulnerable à une attaque de divulgation de clé secrète, en particulier pour les clés de session entre véhicule et le centre de données, ainsi qu'entre véhicule et l'utilisateur. Pour remédier à cette faille, nous avons amélioré le protocole afin d'assurer une communication confidentielle et anonyme au sein du système. Othman et al. [24] ont proposé un protocole d'authentification de messages basé sur des PUF (Physical Unclonable Functions) qui préserve la confidentialité et le partage de secrets afin de garantir la sécurité.

De même, Wang et al. [26] ont proposé un nouveau schéma d'authentification multi-serveur et d'échange de clés pour les environnements IoV, qui utilise une carte à puce et un mot de passe pour

maintenir les clés privées confidentielles.

Enfin, Srinivas et al. [48] ont proposé un "Protocole d'authentification sécurisée pour la collecte de Big Data dans un système de transport intelligent basé sur l'IoT appelé UAP-BCIoT", qui aborde le problème de la protection de la sécurité des dispositifs intelligents IoT dans le contexte des systèmes de transport intelligents (ITS). UAP-BCIoT peut assurer une protection contre un certain nombre d'attaques bien connues. De plus, leur schéma est parfait pour la collecte et l'analyse de Big Data dans les systèmes de transport intelligents basés sur l'IoT. Cependant, après une analyse approfondie de ce schéma, nous avons trouvé qu'il contient une faille. Cette faille affecte l'authentification entre les utilisateurs honnêtes et les dispositifs IoT déployés à l'intérieur des véhicules.

Le Tableau tableau 2.2 résume ces différents protocoles en comparant les techniques cryptographiques utilisées, les entités du modèle de réseau, ainsi que les avantages et limitations de chaque schéma d'authentification dans le système IoV.

Schéma	Année	Techniques cryptographiques	Entités du modèle de réseau	Phases	Avantages/Limites
Jiang et al. [14]	2016	-Hachage -Appairage bilinéaire -ECDLP	-Véhicules -Autorité -RSU	-Initialisation du système -Émission du certificat de la RSU -Génération des clés privées et pseudonymes des véhicules -Distribution des clés de groupe et authentification mutuelle - Mise à jour périodique des clés de groupe	-Les informations du véhicule de l'utilisateur ne peuvent pas être prises par un véhicule malveillant, ni être divulguées, tout en maintenant une confidentialité restreinte dans les VANETs. Pas d'analyse de sécurité formelle
Wazid et al. [15]	2017	-Hachage - Opérations XOR	-Véhicules -Cluster heads -RSU - Serveur cloud	-Enregistrement (RSU/Vs) -Authentification et accord de clé (V2V/V2CH) -Établissement de clé RSU2RSU -Mise à jour du mot de passe	-Applicable aux applications basées sur les VANETs de prochaine génération. -La phase d'analyse des Big Data n'est pas implémentée.
Mohit et al. [17]	2017	- Fonction de hachage	- Véhicules -Serveur - Autorité de confiance	-Configuration du système -Enregistrement de l'utilisateur -Connexion de l'utilisateur -Authentification -Changement de mot de passe	Protocole d'accord de clé léger. Pas d'analyse de sécurité formelle
Ying et al. [16]	2017	-Fonction de hachage -Diffie-Hellman -Chargement des cartes à puce avec les informations de connexion par une autorité de confiance centralisée	-Autorité de confiance -Véhicules - RSU	-Enregistrement de l'utilisateur -Connexion de l'utilisateur -Authentification de l'utilisateur -Authentification des données -Changement de mot de passe	Utilise des opérations cryptographiques à faible coût. L'autorité de confiance utilise un coût computationnel élevé, ce qui la rend vulnérable aux attaques DoS.

Zhou et al. [18]	2018	-ECC - Fonction de hachage	-Véhicules -RSU -Autorité de confiance	-Initialisation du système -Génération des clés -Étape de signature -Étape de vérification	Protocole d'accord de clé léger. La mobilité élevée et les topologies dynamiques ne sont pas prises en compte.
Chen et al. [19]	2019	- Fonction de hachage -Exponentiation modulaire -ECC	-Véhicules - RSU -Autorité	-Connexion de l'utilisateur -Authentification de l'utilisateur	Résiste à diverses attaques. La mobilité élevée et les topologies dynamiques ne sont pas prises en compte.
Wang et al. [20]	2019	- Fonction de hachage -Appairage bilinéaire -Exponentiation modulaire -ECC	-Véhicules	-Authentification	Utilise des opérations cryptographiques à faible coût, mais leur utilisation répétée alourdit le protocole.
Ma et al. [21]	2019	-Fonction de hachage -ECC	-Véhicules -Nœud de Fog - Serveur cloud	-Configuration -Enregistrement de l'utilisateur du véhicule -Enregistrement du nœud de Fog -Authentification mutuelle et d'accord de clé	Supporte la préservation de la vie privée. Pas d'analyse de sécurité formelle Leur schéma ne maintient pas la propriété d'anonymat de l'utilisateur.
Vasudev et al. [22]	2020	- Fonction de hachage	- Véhicules -Serveur -Autorité	-Enregistrement -Connexion, Authentification et de communication	Protocole d'accord de clé léger Pas d'analyse de sécurité formelle

Srinivas et al. [48]	2020	-Fonction de hachage -Applique extracteur flou ECC	-Utilisateur -IoT déployés dans les véhicules -Cloud-Gateway -Autorité de confiance	-Initialisation du système -Inscription des dispositifs IoT -Connexion et d'authentification -Mise à jour du mot de passe et/ou des données biométriques -Ajout dynamique des dispositifs IoT -Révocation des appareils mobiles des utilisateurs -Analyse des Big Data -validation des identifiants des dispositifs IoT	Schéma est parfait pour la collecte et l'analyse de Big Data dans ITS basés sur l'IoT. Nous avons identifié une faille affectant l'authentification entre les utilisateurs et les dispositifs IoT déployés dans les véhicules.
Othman et al. [24]	2021	-Fonction de hachage -Appairage bilinéaire -Exponentiation modulaire -PUF	-Véhicules -RSU -Autorité de confiance	-Configuration du système -Inscription des véhicules et des RSU -Authentification mutuelle V2I Renouvellement de clé -Communication sécurisée et authentifiée V2V -Mécanisme de révocation	Le protocole ne garantit pas l'authentification U2V, mais seulement l'authentification V2V et V2I.
Jiang et al. [25]	2021	-Fonction de hachage -ECC -PUF	-Utilisateur -Centre de données -Véhicule	- Enregistrement de l'utilisateur Enregistrement véhicule - Connexion et d'authentification -Mise à jour du mot de passe -Mise à jour des données biométriques	Authentification ultra-légère, créer trois clés de session entre les trois acteurs. nous avons identifié une vulnérabilité majeure dans ce protocole, le rendant vulnerable à une attaque de divulgation de clé secrète.

Wang et al. [26]	2022	-Fonction de hachage -ECC	-Utilisateur -Autorité -Véhicule	-Configuration - Enregistrement -Authentification -Mise à jour du mot de passe	Authentification utilise un coût computationnel élevé, ce qui la rend vulnérable aux attaques DoS.
Yang et al. [102]	2023	-Fonction de hachage -ECC XoR	-Autorité -Véhicule -RSU	Initialisation et d'enregistrement du système - Authentication and key agreement	Le protocole ne garantit pas l'authentification U2V, mais seulement l'authentification V2I.
He et al. [103]	2024	-fuzzy extractor -PUF Discrete logarithm problem	-Autorité -Véhicule -RSU	Initialisation du système - Enregistrement et phase d'accord de clé authentifié	Le protocole ne garantit pas l'authentification U2V, mais seulement l'authentification V2I et V2V.

TABLE 2.2 : Mécanismes d'authentification existants pour l'IoV : étude comparative

2.2.1.1 Analyse de la Sécurité des protocoles existants dans l'IoV :

Suite à l'analyse des divers schémas d'authentification dans le cadre de l'IoV, on examine la sécurité de ces schémas en ce qui concerne leur protection contre différentes attaques et vulnérabilités. Les techniques cryptographiques appliquées, les techniques d'analyse de sécurité et les contraintes de chaque schémas sont résumées dans le tableau présenté. Nous présentons une étude approfondie des paramètres de sécurité et des capacités de résistance aux attaques des protocoles examinés.

- **Techniques de sécurité et limitations :** La plupart des protocoles identifiés utilisent des techniques cryptographiques comme les fonctions de hachage, l'ECC, ainsi que des méthodes plus spécifiques comme les PUF. Bien que ces protocoles offrent un certain niveau de sécurité, ils présentent plusieurs limites. Par exemple, le protocole de Wang et al. [26] engendre un coût computationnel élevé, ce qui le rend vulnérable aux attaques par déni de service (DoS). Par ailleurs, Ma et al. [21] ont souligné que même s'ils sont efficaces contre certaines attaques, ils ont des lacunes lorsqu'il s'agit d'attaques de traçabilité et anonymat de l'utilisateur.
- **Fonctionnalités de sécurité et résistance aux attaques :** Les protocoles sont examinés pour leur capacité à faire face à différentes attaques courantes dans IoV. Par exemple, le protocole proposé par Vasudev et al. [22] manque d'une telle analyse, ce qui laisse des incertitudes quant à sa robustesse face aux attaques par usurpation d'identité. L'intégration de PUF dans les protocoles, comme proposé par Jiang et al. [25], permet une meilleure résistance contre les attaques par usurpation et clonage. Cependant nous avons identifié des failles essentielles, en particulier dans la gestion des clés secrètes. D'autre part, des schémas comme celui présenté par Srinivas et al. [48] utilisent des mécanismes cryptographiques à moindre coût. Cependant, une étude approfondie de ce schéma a mis en évidence une faiblesse qui met en risque l'authentification entre les utilisateurs légitimes et les dispositifs IoT intégrés aux véhicules.

2.2.1.2 Analyse des Performances des protocoles existants dans l'IoV :

Afin d'évaluer les coûts de calcul associées à la phase d'authentification et d'échange de clés, et sur la base des résultats expérimentaux antérieurs appliqués dans [28], les fonctions cryptographiques utilisées dans les précédents schémas d'authentification pour l'IoV sont répertoriées dans le tableau tableau 2.3 avec leurs temps d'exécution estimés. Le tableau 2.5 compare les schémas d'authentification en termes de coût calcul, telles que celles menées par Wang et al. [20], montrent que l'utilisation répétée, même pour des opérations cryptographiques à faible coût, rend le protocole plus complexe. Selon Chen et al. [19], la complexité des opérations a également un impact sur la performance, car elle ne prend pas en considération l'intégration de la mobilité élevée des véhicules, ce qui restreint leur utilisation pratique dans des environnements réels et dynamiques. Bien que Jiang et al. [25] semblent être l'une des meilleures propositions en ce qui concerne les performances, notre analyse approfondie a révélé une vulnérabilité. Il serait crucial d'améliorer la méthode de gestion des clés

secrètes, éventuellement en renforçant la complexité de leur établissement et en intégrant des étapes de vérification supplémentaires afin de identifier toute tentative d'exploitation.

Cryptographic Operation	Symbol	Device	Server
Multiplication scalaire	T_{ecm}	5.9 ms	2.6 ms
Addition de points	T_{eca}	0.35 ms	0.14 ms
Appairage bilinéaire	T_{bp}	9.23 ms	3.78 ms
Exponentiation modulaire	T_e	7.86 ms	2.34 ms
Chiffrement/déchiffrement symétrique	T'_{enc}/T'_{den}	0.079 ms	0.041 ms
Fonction de hachage	T_h	0.026 ms	0.011 ms
PUF (128-bit arbiter)	T_p	0.12 ms	-
Reproduction de l'extracteur flou	T_{fe}	3.28 ms	-
Retrieval algorithm	T_{Ret}	3.31 ms	-
Message authentication code	T_{mac}	2.9 ms	1.23 ms

TABLE 2.3 : Opérations cryptographiques et leurs coûts en temps pour l'IoV [28]

2.2.1.3 Synthèse sur la sécurité et la performance des protocoles existants dans l'IoV :

Les études comparatives mettent en évidence une amélioration significative des protocoles de sécurité pour l'IoV, tout en mettant en lumière des défis persistants. Bien que si certains schémas donnent une importance particulière à l'amélioration des performances, il est primordial de ne pas mettre en jeu la sécurité au profit d'une complexité de calcul réduite. Par exemple, Jiang et al. [25] offrent l'une des propositions les plus performantes, mais notre analyse approfondie a mis en évidence une vulnérabilité critique. Il est indispensable de renforcer la gestion des clés secrètes, notamment en augmentant la complexité de leur établissement et en intégrant des étapes supplémentaires de vérification pour détecter toute tentative d'exploitation. Un autre protocole pertinent a été repéré, abordant la collecte de Big Data dans les systèmes de transport intelligents basés sur l'IoT. Ce protocole, proposé par Srinivas et al. [48] assure une sécurité complète à travers différentes phases, garantissant l'intégrité et la confidentialité des données, tout en permettant une gestion flexible et dynamique des dispositifs IoT dans un environnement de système de transport intelligent (ITS). Une étape essentielle est la vérification des informations d'identification des dispositifs IoT : La Cloud-Getway effectue régulièrement une vérification de la validité des informations d'identification des nœuds IoT et surveille leur comportement afin de détecter toute activité illicite. Cependant, Une étude détaillée de ce schéma a mis en démonstration une vulnérabilité qui impacte l'authentification entre les utilisateurs légitimes et les dispositifs IoT installés dans les véhicules.

Protocole	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
Wazid et al. [15]	✓	✓	✓	✓	×	×	-	✓	✓	-	✓	-	-	✓	✓	✓	✓	×	✓	×
Ying et al. [16]	✓	✓	✓	×	×	×	✓	✓	✓	✓	×	-	×	✓	×	×	×	-	×	×
Mohit et al. [17]	×	×	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	×	✓
Zhou et al. [18]	✓	✓	✓	✓	×	×	-	✓	-	-	×	-	×	-	×	-	×	×	×	×
Chen et al. [19]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	×	×	×	✓	×
Ma et al. [21]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	×	✓
Vasudev et al. [22]	×	✓	×	✓	×	✓	×	✓	✓	✓	✓	×	✓	✓	✓	✓	×	×	×	×
Othman et al. [24]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Jiang et al. [25]	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Wang et al. [26]	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

A1 : Usurpation d'identité **A2** : Attaque Man-In-The-Middle (MITM) **A3** : Attaque par rejeu **A4** : Attaque DoS **A5** : Ecoute clandestine **A6** :Protection de clé secrète à long terme **A7** : Dispositif volé **A8** : Attaque clé session **A9** : Attaque interne **A10** : Attaque on/hors ligne par devinette de mot de passe **A11** : Anonymat **A12** : Spoofing **A13** : Attaque physique **A14** : Traçabilité **A15** : Attaque par dictionnaire **A16** : Attaque non synchronisée **A17** : Sécurité avant et arrière **A18** : Analyse de Big Data **A19** : AVISPA/Scyther/ProVerif **A20** : Identification du nœud IoT

TABLE 2.4 : Comparaison des mécanismes d'authentification pour l'IoV : fonctionnalité et sécurité

Scheme	Vehicle/User	RSU	TA/CS	VS	Total Cost
Wazid et al. [15]	$8T_h \approx 0.208$	$8\,T_h \approx 0.208$	-	$8\,T_h \approx 0.208$	$24\,T_h \approx 0.624$ ms
Ying et al. [16]	$T_e + 5\,T_h + T_{sen} \approx 8.069$	-	$T_e + 5\,T_h + T_{sen} \approx 8.069$	-	$2\,T_e + 10\,T_h + 2\,T_{sen} \approx 16.138$ ms
Mohit et al. [17]	$7\,T_h \approx 0.182$	-	$9\,T_h \approx 0.234$	$4\,T_h \approx 0.104$	$20\,T_h \approx 0.52$ ms
Zhou et al. [18]	$7T_{ecm} + 27T_{eca}4T_h \approx 42.104$	-	-	-	$7\,T_{ecm} + 27\,T_h + 47\,T_h \approx 42.104$ ms
Chen et al. [19]	$3T_e + 7T_h + T_{sem} \approx 23.841$	$1\,T_h \approx 0.026$	$3T_e + 4\,T_h + T_{sen} \approx 23.763$	-	$6T_e + 12T_h + 2T_{sen} \approx 47.63$ ms
Wang et al. [20]	$7\,T_h + 4\,T_{ecm} + 2T_{bp} + T_e \approx 50.102$	-	-	-	$7\,T_h + 4\,T_{ecm} + 2T_{bp} + T_e \approx 50.102$ ms
Ma et al. [21]	$4\,T_h + 3T_{ecm} \approx 17.804$	-	$11\,T_h + 10\,T_{ecm} \approx 59.286$	-	$15\,T_h + 13\,T_{ecm} \approx 77.09$ ms
Srinivas et al. [48]	$17\,T_h + T_{fe} + 5T_{ecm} \approx 39.42$	-	$10\,T_h + +3T_{ecm} \approx 71.13$	$8\,T_h + 4T_{ecm} \approx 23.8$	$35\,T_h + T_{fe} + 11T_{ecm} + 2T_{eca} \approx 71.13$ ms
Othman et al. [24]	$T_{Ret} + 5T_h + T_p + 2T_{mac} \approx 6.429$	$T_{bp} + 2T_h + T_{ecm} \approx 15.182$	$3T_h + T_{ecm} \approx 5.978$	-	$T_{Ret} + 10T_h + T_p + 2T_{mac} + T_{bp} \approx 27.589$ ms
	$2T_{Ret} + 6T_h + 2T_{mac} + 2T_{sec} \approx 12.655$	-	-	-	$2T_{Ret} + 6T_h + 2T_{mac} + 2T_{sec} \approx 12.655$ms
Jiang et al. [25]	$T_p + 13\,T_h + 2T_{ecm} + + \approx 13.38$	-	$19T_h + T_{ecm} \approx 2.809$	$T_p + 12\,T_h + T_{enc} \approx 7.336$	$2T_p + 44\,T_h + 4\,T_{ecm} + +3T_{enc} \approx 23.55$
Wang et al. [26]	$4T_{ecm} + 2\,T_{eca} + 6\,T_h \approx 24.456$	-	$6T_{ecm} + 2\,T_{eca} + 5\,T_h \approx 23.23$	-	$10T_{ecm} + 4\,T_{eca} + 11\,T_h \approx 47.686$

TABLE 2.5 : Comparaison des mécanismes d'authentification pour l'IoV : analyse des coûts de calcul

2.2.2 Les protocoles d'authentification existants dans l'IoD

Dans cette partie, nous examinerons différentes recherches sur la sécurité de la communication entre les utilisateurs et les drones, en analysant les vulnérabilités et la robustesse des protocoles d'authentification proposés dans l'Internet des Drones (IoD). Selon Derhab et al. [30], une étude a été réalisée sur les exigences de sécurité de l'IoD, mettant en lumière différentes attaques qui pourraient perturber ses services, ainsi que les mesures possibles pour assurer des communications sécurisées. Plus précisément, dans le contexte de l'authentification et de l'échange de clés dans le domaine de l'IoD, Wazid et al. [31], ont examiné les difficultés associées à la création de mécanismes d'authentification pour garantir la sécurité des communications dans IoD. En se référant à plusieurs recherches publiées depuis 2019 sur les protocoles d'authentification des utilisateurs dans l'IoD, nous classifions ces protocoles selon des algorithmes cryptographiques employés et du nombre de facteurs appliqués. Ces protocoles utilisent différentes primitives cryptographiques (fonctions de hachage, opérations XOR, concaténation, cryptosystèmes symétriques, cryptographie sur courbes elliptiques) et combinent deux ou trois facteurs (appareil mobile, mot de passe, biométrie) afin d'atteindre à la fois la sécurité et l'efficacité énergétique tout en réduisant les coûts de communication et de calcul. Srinivas et al. [32], Wazid et al. [33], Zhang et al. [34], et Yu et al. [35] ont proposé des protocoles qui se concentrent seulement sur des opérations XOR et des fonctions de hachage, ce qui permet de diminuer les coûts de calcul. Zhang et al. utilisent une méthode d'authentification à deux facteurs (Appareil mobile et mot de passe), tandis que Srinivas et al., Wazid et al. et Yu al. ont inclus l'extracteur flou comme troisième méthode pour authentifier les biométriques de l'utilisateur. Cependant, les protocoles de Srinivas et al., Wazid et al., et Zhang et al. sont vulnérables à des attaques telles que le vol d'appareil mobile et la fuite de secret éphémère [36]. Le schéma de Yu et al. présente des limites en ce qui concerne sa capacité à s'adapter uniquement à un utilisateur mobile et nécessite des améliorations en termes de scalabilité [37].

Akram et al. [38], Zhang et al. [39], ainsi que Tanveer et al. [40], ont proposé des protocoles qui combinent des fonctions de hachage et des cryptosystèmes symétriques pour transférer plusieurs secrets en utilisant un seul paramètre (un texte chiffré), tout en mettant en place une authentification à trois facteurs. Cependant, ces protocoles présentent un coût intermédiaire et sont vulnérables à diverses failles. Prenons l'exemple suivant. Selon Liu et al. [41], le protocole d'Akram et al. présente un manque de confidentialité persistante (PFS) et est exposé à des attaques telles que l'attaque par rejeu, le vol de vérificateur et la recapture de drones.

Enfin, Tanveer et al. [42], Ullah Jan et al. [67] et Berini et al. [43] ont présenté l'intégration de ECC, en combinaison avec des opérations XOR et des fonctions de hachage. Tanveer et al. [42] mettent en œuvre une authentification à trois facteurs, intégrant également des systèmes de cryptographie symétriques afin de rendre leur protocole plus résistante aux attaques de sécurité. Toutefois, ce protocole entraîne des coûts de calcul élevés et nécessite des améliorations en termes de scalabilité [37]. tandis que [43] et [67] utilisent une authentification à deux facteurs, lors de l'analyse des protocoles Ullah Jan

et al. [67] et Berini et al. [43], nous avons identifié des vulnérabilités, notamment celle décrite par Berini et al. [43], qui propose un protocole léger mais vulnérable aux attaques d'initiés privilégiés. Ces attaques permettent de capturer l'identité lors de la phase d'enregistrement. De plus, l'attaque par vol de dispositif mobile peut récupérer le nombre aléatoire stocké en texte clair, et Ai utilise des attaques par analyse de puissance, où Ai est défini comme hash(hash(identité, nombre aléatoire) et hash(identité, mot de passe)). L'attaquant devine un mot de passe PwdA pour l'utilisateur et vérifie la légitimité de Ai. Si la supposition est correcte, l'attaque réussit à déterminer le mot de passe correct de l'utilisateur. Sinon, l'attaquant continue à essayer d'autres mots de passe. Il est évident que l'attaquant peut finalement deviner le mot de passe correct et obtenir les informations secrètes sensibles de l'utilisateur en utilisant des attaques d'initiés privilégiés et le vol de dispositif mobile. Concernant le protocole de Ullah Jan et al. [67], il est vulnérable à l'écoute clandestine, nécessitant la mise en place de mesures contre l'écoute pendant la phase d'authentification. De plus, l'utilisation de fonctions XOR sur les paramètres peut aider à déterminer les paramètres secrets.

TABLE 2.6 : Mécanismes d'authentification existants pour l'IoD : étude comparative

Schéma	Année	Techniques appliquées	Facteurs	Analyse de sécurité	Limites
Srinivas et al. [32]	2019	Fonction de hachage, XOR	Deux facteur	Modèle oracle aléatoire et Analyse de sécurité informelle	Ne protège pas contre Problèmes de scalabilité, Attaque de traçabilité, Usurpation basée sur le Stolen Verifier.
Wazid et al. [33]	2019		Trois facteurs	Vérification de sécurité formelle (AVISPA) et Analyse de sécurité informelle	Ne protège pas contre "Attaque de traçabilité, Attaque du Stolen Verifier, Usurpation d'identité utilisateur, Attaque de réinitialisation de serveur, Divulgation de clé de session
Zhang et al. [34]	2020			Modèle oracle aléatoire Analyse de sécurité informelle	Ne protège pas contre Attaque par falsification
Tanveer et al. [40]	2020	Hachage, cryptosystèmes		Vérification de sécurité formelle (Scyther) et BAN logic Analyse de sécurité informelle	Ne protège pas contre Attaque de traçabilité
Zhang et al. [39]	2021			Modèle oracle aléatoire et Analyse de sécurité informelle	Le coût de calcul et de communication est très élevé
Akram et al. [38]	2022	symétriques		Modèle oracle aléatoire et Analyse de sécurité informelle	Ne protège pas contre Attaques du Stolen Verifier ou capture par drone
Tanveer et al. [42]	2022	ECC, Hachage, XOR		Évaluation de la sécurité (Scyther) et Real-Or-Random et Analyse de sécurité informelle	Ne protège pas contre Préservation dynamique de la vie privée
Yu et al. [47]	2023		Deux Facteur	Vérification de sécurité formelle (AVISPA) et sécurité informelle	Le coût de communication est très élevé
Berini et al. [43]	2023			Vérification de sécurité formelle (AVISPA) et sécurité informelle	Vulnérable aux privileged-insider attacks
Ullah Jan et al. [67]	2024			Analyse de sécurité informelle	Vulnérable aux eavesdropping attacks

2.2.2.1 Analyse de la Sécurité des protocoles existants dans l'IoD :

L'authentification dans l'IoD est un élément essentiel pour assurer la confidentialité, l'intégrité et la disponibilité des données. Les différents protocoles d'authentification analysés dans cette étude utilisent diverses méthodes et techniques pour sécuriser les échanges d'informations. Cependant, malgré la diversité des approches, la sécurité de ces mécanismes est souvent mise à l'épreuve par des menaces variées telles que les attaques MITM, les attaques par rejeu, et les attaques internes. Les techniques utilisées, les méthodes d'analyse de sécurité et les limites de chaque protocole sont présentées dans le tableau tableau 2.6, tandis que le tableau tableau 2.7 fournit une comparaison de ces protocoles face à des attaques spécifiques.

- **Techniques de sécurité et limitations :** Les techniques de sécurité utilisées par les protocoles étudiés varient considérablement. Par exemple, Wazid et al. [33] utilisent une combinaison de hachage et de trois facteurs d'authentification, ce qui permet d'améliorer la sécurité en augmentant le nombre de facteurs nécessaires pour valider une authentification. De même, Tanveer et al. [42]. Toutefois, certaines limitations apparaissent : ces protocoles restent vulnérables à des attaques telles que l'attaque par "Stolen Verifier" ou la falsification, qui peuvent compromettre la confidentialité des utilisateurs et l'intégrité des données. En outre, les coûts de calcul et de communication élevés dans certains protocoles, comme celui de Ullah Jan et al., peuvent être un frein dans des environnements IoD avec des ressources limitées.
- **Fonctionnalités de sécurité et résistance aux attaques :** Les protocoles analysés offrent diverses fonctionnalités de sécurité, en fonction des menaces qu'ils visent à contrer. Par exemple, Wazid et al. [33] ont mis en place une résistance aux attaques de type MITM et aux attaques par rejeu, et ont intégré des mécanismes pour protéger les clés de session. Toutefois, ces protocoles ne sont pas à l'abri de certaines vulnérabilités, comme les attaques de "Stolen Verifier", où un attaquant pourrait potentiellement obtenir des informations sensibles sur le processus d'authentification. D'autres protocoles, comme celui de Tanveer et al. [42], se distinguent par leur capacité à résister aux attaques de traçabilité, protégeant ainsi la vie privée des utilisateurs dans les environnements IoD. Cependant, malgré ces avancées, aucun protocole n'est totalement invulnérable. Par exemple, le protocole de Berini et al. [43] est efficace contre les attaques externes, mais reste vulnérable aux attaques internes et physiques, ce qui peut poser des problèmes dans des systèmes où des acteurs malveillants internes sont présents.

Protocole	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17
Wazid et al. [33]	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	×	✓	×	✓
Srinivas et al. [32]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	×	✓	×	✓
Zhang et al. [34]	✓	×	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓	×	×	✓
Tanveer et al. [40]	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	×	✓	×	✓	×	×
Nikooghadam al. [44]	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	×	✓	×	✓	×	×
Hussain et al. [36]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓	×	✓	×	✓
Zhang et al. [39]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓	✓	✓
Tanveer et al. [42]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓	✓	✓
Hussain et al. [45]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
Akram et al. [38]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
Yu et al. [47]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
Berini et al. [43]	✓	✓	✓	×	✓	×	✓	✓	✓	×	×	✓	✓	✓	✓	✓	✓
Ullah Jan et al. [67]	✓	×	✓	×	✓	×	✓	✓	✓	×	×	✓	✓	✓	✓	✓	✓

P1 : “Attaque MITM” ; **P2** : “Attaque sur la sécurité de la clé de session” ; **P3** : “Attaque par rejeu” ; **P4** : “Attaque interne” ; **P5** : “Attaque DoS” ; **P6** : “Attaque par devinette de mot de passe” ; **P7** : “Écoute clandestine” ; **P8** : “Anonymat” ; **A9** : “Protection de la clé secrète à long terme” ; **A10** : “Appareil volé” ; **A11** : “Attaque physique” ; **A12** : “Usurpation d’identité” ; **A13** : “Attaque non synchronisée” ; **A14** : “Traçabilité” ; **A15** : “Sécurité avant et arrière” ; **A16** : “Validation des informations d’identification du nœud” ; **A17** : “ d’AVISPA/Scyther/Tamarin”

TABLE 2.7 : Comparaison des mécanismes d’authentification pour l’IoD : fonctionnalité et sécurité

2.2.2.2 Analyse des Performances des protocoles existants dans l'IoD :

Une analyse approfondie des coûts de calcul et de communication est nécessaire pour évaluer les protocoles d'authentification dans l'IoD, car ces éléments ont un impact direct sur leur performance et leur application. D'après les études précédentes publiées dans [42], le tableau tableau 2.8 présente les fonctions cryptographiques employées dans les schémas d'authentification pour l'IoD, ainsi que leurs estimations de temps d'exécution.

- **Coûts de Calcul :** En examinant le tableau 2.2.2.2, on remarque que les coûts de calcul varient significativement d'un protocole à l'autre. Par exemple, le protocole proposé par Tanveer et al. [42] a un coût de calcul total de 21.9 ms ms, ce qui est relativement élevé par rapport à celui de Wazid et al. [33] qui est de 5,669 ms. Les protocoles qui impliquent des opérations cryptographiques plus complexes, comme ceux de Tanveer et al. [42] affichent des coûts plus élevés, parce qu'ils intègrent des techniques de sécurité plus sophistiquées, comme le chiffrement symétrique. Bien que cela rende ces protocoles plus sûrs, cela peut aussi augmenter leur complexité.
- **Coûts de Communication :** En examinant le tableau 2.10, on remarque que les coûts de communication varient également selon les protocoles. Le protocole de Tanveer et al. [40] présente le coût de communication le plus faible, avec un total de 1376 bits, ce qui en fait une option intéressante pour des applications où la bande passante est limitée. À l'inverse, le protocole de Ullah Jan et al. [67] utilise 3070 bits, ce qui indique un échange de données plus important entre les parties impliquées. Cette différence souligne l'importance de choisir un protocole qui parvient à un bon équilibre entre la sécurité et l'efficacité, en tenant compte des besoins spécifiques de l'application IoD. tableau 2.7

Opération cryptographique	**Notation**	**Drone**	**Serveur**
Multiplication de point ECC	T_{sm}	2,42 ms	0,745 ms
Addition de point ECC	T_{sa}	0,134 ms	0,003 ms
Dé(chiffrement) symétrique	$T_{enc/den}$	0,425 ms	0,07 ms
Fonction de hachage	T_h	0,381 ms	0,063 ms
Reproduction d'extracteur flou	T_{fe}	2,42 ms	0,745 ms

TABLE 2.8 : Temps d'opérations cryptographiques sur le drone et le serveur

Scheme	User	S	Drone	Total Cost
Wazid et al. [33]	$7T_h \approx 2.667$	$5T_h \approx 0.315$	$7T_h \approx 2.667$	$19T_h \approx 5.669$ ms
Srinivas et al. [32]	$14T_h + 7T_{fe} \approx 7.754$	$9T_h \approx 0.567$	$7T_h \approx 2.667$	$30T_h + 7T_{fe} \approx 10.988ms$
Zhang et al. [34]	$10T_h \approx 3.810$	$7T_h \approx 0.441$	$7T_h + 2.667$	$24T_h \approx 6.918ms$
Tanveer et al. [40]	$6T_h + 2T_{enc/dec} + T_{fe} \approx 5.556$	$2T_h + 2T_{enc/dec} \approx 0.267$	$3T_h + 2T_{enc/dec} \approx 1.993$	$11T_h + 6T_{enc/dec} + T_{fe} \approx 7.815$ ms
Nikooghadam al. [44]	$2T_{sm} + 6T_h \approx 7.126$	$2T_{sm} + 5T_h \approx 1.805$	$8T_h \approx 3.048$	$4T_{sm} + 19T_h \approx 11.979$ ms
Hussain et al. [36]	$15T_h + T_{fe} \approx 8.135$	$9T_h + 2T_{enc/dec} \approx 0.707$	$7T_h \approx 2.667$	$31T_h + 2T_{enc/dec} + T_{fe} \approx 11.509ms$
Tanveer et al. [42]	$6T_h + 3T_{enc/dec} + 3T_{sm} + T_{fe} \approx 13.241$	$2T_h + 3T_{enc/dec} + 3T_h \approx 1.826$	$3T_h + 2T_{enc/dec} + 2T_{sm} \approx 6.833$	$11T_h + 8T_{enc/dec} + 5T_{sm} + T_{fe} \approx 21.9$ ms
Hussain et al. [45]	$2T_{sm} + 8T_h + T_{fe} \approx 10.308$	$T_{sm} + 7T_h \approx 1.186$	$6T_h \approx 2.286$	$3T_{sm} + 21T_h + T_{fe} \approx 13.780$ ms
Akram et al. [38]	$9T_h \approx 3.429$	$7T_h + enc/dec \approx 0.581$	$8T_h + T_{fe} \approx 5.468$	$29T_h + 2T_{fe} \approx 6.667$ ms
Yu et al. [47]	$12T_h + T_{fe} \approx 6.992$	$9T_h \approx 0.567$	$8T_h + T_{fe} \approx 5.468$	$29T_h + 2T_{fe} \approx 13.027$ ms
Berini et al. [43]	$9T_h + 2T_{sm} \approx 9.031$	$6T_h + T_{sm} \approx 1.23$	$6T_h \approx 2.286$	$21T_h + 3T_{sm} \approx 12.44$ ms
Ullah Jan et al. [67]	$9T_h + T_{sm} \approx 5.849$	$4T_h \approx 0.252$	$3T_h + T_{sm} \approx 3.56$	$16T_h + 3T_{sm} \approx 9.661$ ms

TABLE 2.9 : Comparaison des mécanismes d'authentification pour l'IoD : analyse des coûts de calcul

Messages échangés (in bits)	User → S	S → Drone	Drone → User	Total Cost (in bits)
Wazid et al. [33]	672	512	512	1696
Srinivas et al. [32]	672	512	352	1536
Zhang et al. [34]	672	480	320	1472
Tanveer et al. [40]	544	416	416	1376
Nikooghadam al. [44]	832	992	512	2336
Hussain et al. [36]	672	672	672	2016
Tanveer et al. [42]	704	672	480	1856
Hussain et al. [45]	992	672	544	2208
Akram et al. [38]	896	768	512	2176
Berini et al. [43]	960	960	544	2464
Ullah Jan et al. [67]	960	1664	448	3070

TABLE 2.10 : Comparaison des mécanismes d'authentification pour l'IoD : analyse des coûts de communication

2.2.2.3 Synthèse sur la sécurité et la performance des protocoles existants dans l'IoD :

Les protocoles présentés précédemment ne garantissent pas un équilibre entre la sécurité et la consommation d'énergie des drones. De plus, ils reposent sur l'hypothèse que l'utilisateur doit connaître l'identité du drone opérant dans la zone où il souhaite collecter des informations en temps réel. Sans cette connaissance préalable, l'utilisateur ne peut pas authentifier le drone et, par conséquent, ne peut pas accéder aux informations de cette zone. Cependant, une limitation majeure de ces travaux est que les drones sont souvent limités par leur autonomie et peuvent être remplacés par d'autres drones. Il serait donc plus efficace qu'un utilisateur demande une authentification d'accès à une zone spécifique, plutôt que de devoir authentifier chaque drone individuellement. Si l'utilisateur a accès à cette zone, il pourrait alors s'authentifier avec n'importe quel drone affecté par la station de contrôle.

Un deuxième axe de recherche concernant les drones est lié aux récentes réglementations de la FAA (Federal Aviation Administration) [62] et de l'Agence européenne de la sécurité aérienne (EASA) [63], qui ont introduit une réglementation connue sous le nom d'Identification à distance (RemoteID). RemoteID exige que tous les drones diffusent périodiquement des messages indiquant leur identité. La conformité aux réglementations RemoteID est devenue obligatoire à partir de septembre 2022. Puisque cette réglementation est récente, les études sur leur authentification sont limitées. De plus, de nombreux domaines impliquent souvent plusieurs drones travaillant ensemble pour accomplir des tâches dépassant les capacités des unités individuelles. Il est donc nécessaire d'authentifier l'ensemble des drones (essaims de drones).

En 2022, Ardin et al. [68] ont présenté un système d'authentification et de transfert pour les essaims

de drones. Leur protocole permet à un nouveau drone de rejoindre un essaim existant, mais ne prend pas en compte la vérification de l'autorité d'accès pour surveiller chaque zone de l'espace aérien. Un autre travail d'Alkadi et Shoufan [69] propose un protocole décentralisé pour la gestion du trafic, offrant divers services de sécurité tels que la confidentialité et l'intégrité. Leur principal objectif est de sécuriser les interactions entre les entités, mais ils négligent l'anonymat et l'authenticité des drones.

Outre les avancées scientifiques, plusieurs systèmes d'identification de drones commerciaux émergent sur le marché. Parmi eux, on trouve ScaleFlyt de Thales [64], BLIP (Broadcast Location and Identification Platform) d'Unifly [65], et SIAM (Secure Airspace Integrated Management Tool) de RelmaTech [66]. Ces solutions utilisent les fonctionnalités d'authentification et d'anonymat de la technologie cellulaire LTE, qui n'est pas le mécanisme de communication actuellement proposé par les réglementations RemoteID et le groupe de travail DRIP (Drone Remote Identification Protocol) de l'IETF.

Cet examen des recherches existantes montre clairement que l'utilisation des drones dans la vie réelle pose plusieurs défis en matière de sécurité : la protection des données échangées entre les utilisateurs et les drones, ainsi que la gestion de l'authentification entre les drones et les systèmes aéronautiques.

Chapitre 3

Démarche et méthodologie de l'ensemble du travail de recherche

L'objectif de cette thèse est de créer des solutions afin d'améliorer la sécurité des communications dans le domaine de l'Internet des véhicules (IoV) et des drones (IoD). Dans la section 1.3., trois axes de recherche principaux ont été identifiés pour réaliser cet objectif. Nous présentons dans ce chapitre la méthodologie utilisée tout au long de notre étude, en soulignant l'harmonie entre les objectifs établis et la structure des chapitres qui suivent. Dans l'ensemble, nos travaux sont structurés en deux grandes volets : Le développement d'un protocole d'authentification pour l'IoV et d'un deuxième volet pour l'IoD.

3.1 Méthodologie pour la conception du protocole d'authentification dans IoV/IoD

La figure 3.1 illustre les étapes clés du processus de conception d'un protocole d'authentification. Ce processus commence par la définition des modèles réseau pour les deux contextes, IoV et IoD, afin de comprendre les spécificités et les besoins de chaque environnement. Ensuite, un modèle d'attaques est défini en utilisant des cadres théoriques reconnus, comme les modèles de Canetti-Krawczyk et de Dolev-Yao, pour identifier les vulnérabilités potentielles. Parallèlement, un modèle d'authentification est mis en place, en intégrant des principes tels que l'authentification mutuelle, l'anonymat, la génération de clés et l'intraçabilité. Des contre-mesures adaptées, telles que les techniques cryptographiques, la biométrie ou les PUF (Physical Unclonable Functions), sont ensuite sélectionnées. La conception du protocole comprend plusieurs phases : pré-déploiement, connexion, authentification, génération de clés, et mise à jour des paramètres. Enfin, des analyses de sécurité formelles, utilisant des outils comme Scyther, Tamarin ou AVISPA, et des évaluations des performances, portant sur la consommation énergétique et la surcharge computationnelle, permettent de valider la robustesse et l'efficacité du

protocole développé.

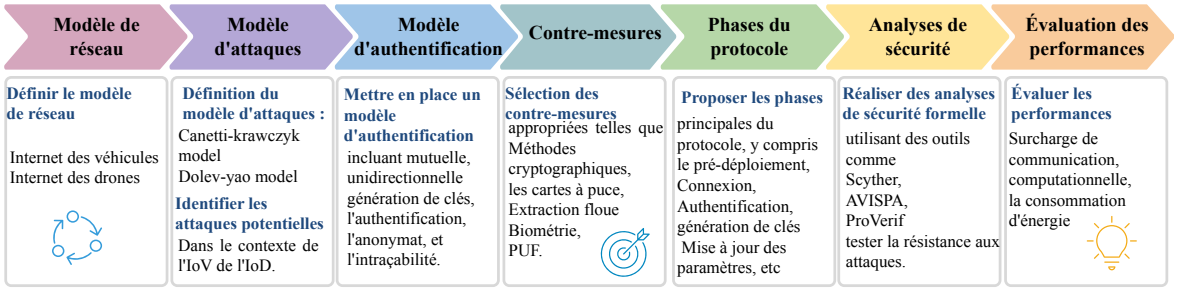


FIGURE 3.1 : Processus de conception d'un protocole d'authentification pour l'IoV/IoD

3.2 Volet 1 : Conception du protocole d'authentification pour IoV

3.2.1 Modèle de réseau de l'IoV

L'Internet des Véhicules (IoV) représente un cadre technologique innovant qui intègre les réseaux sans fil, les technologies mobiles, l'infrastructure cloud et les objets IoT pour connecter de manière fluide différents acteurs du système de transport intelligent, tels que les véhicules, les usagers, les unités de bord de route (RSUs) et les plateformes cloud. L'objectif principal est de permettre une communication transparente et en temps réel entre ces entités pour optimiser les services et la gestion de la mobilité.

3.2.2 Les Modèles de menace :

Avec l'augmentation de la mobilité des véhicules et la transmission massive de données, la sécurité devient un enjeu majeur. La technologie sans fil, qui est essentielle à l'IoV, devient une cible privilégiée pour les cyberattaques. De plus, les dispositifs IoV (comme les RSU, les véhicules, etc.) sont fréquemment situés dans des espaces publics, ce qui les rend plus exposés aux attaques physiques. De cette manière, les pirates peuvent attaquer les dispositifs IoV en utilisant des attaques physiques et voler sans difficulté les informations confidentielles stockées dans l'appareil.

En outre , on trouve plusieurs modèles de menace bien connus, tels que :

- **Modèle de menace de Dolev–Yao (DY) :** Dans ce modèle [53], deux entités communicantes dans le réseau peuvent échanger des informations via un canal non sécurisé, et les entités terminales sont supposées non fiables. Par conséquent, un attaquant A peut intercepter tous les messages échangés, mais il peut également modifier ou falsifier les messages capturés.
- **Modèle de menace de Canetti et Krawczyk (CK) :** Le modèle de menace CK bien connu [54] est plus puissant que le modèle de menace DY, qui est utilisé dans les schémas d'authentification

récents. Selon le modèle de menace CK, en plus des capacités considérées dans le modèle de menace DY, un attaquant a la capacité de compromettre les états de session et les informations confidentielles, telles que les clés secrètes.

Un protocole d'authentification robuste est donc indispensable pour garantir une communication sécurisée entre les différentes entités.

3.2.3 Protocole d'authentification et ses défis

Un protocole d'authentification doit permettre de vérifier l'identité des entités communicantes pour éviter tout accès malveillant et assurer une transmission sécurisée des informations via des clés de session. Cependant, les systèmes classiques utilisant des paires bilinéaires ou le chiffrement par identité présentent des limitations dans le contexte de l'IoV, en particulier en ce qui concerne la protection de l'identité et de la vie privée des utilisateurs.

3.2.4 Technique cryptographique

Il existe plusieurs techniques de cryptographie, parmi lesquelles on trouve la fonction physique uncloneable (PUF). En tant que technologie légère et performante, les PUF sont devenues un sujet de recherche et de développement très prisé ces dernières années.

Un PUF utilise les variations imprévisibles et incontrôlables du mécanisme de fabrication des circuits intégrés comme identifiant unique pour chaque dispositif matériel. Ces variations, qui ne peuvent pas être reproduites même sur la même ligne de production, confèrent à chaque PUF un caractère distinct et inimitable. Même un attaquant connaissant la structure du dispositif ne pourra pas reproduire un circuit cryptographique PUF avec les mêmes variations aléatoires du circuit [49]. Ainsi, la réponse du PUF à une entrée spécifique reste imprévisible et difficile à reproduire, garantissant une protection élevée des informations [50].

Cependant, lorsque les conditions environnementales changent, la réponse du PUF peut différer pour la même demande. Des éléments externes, tels que la température, peuvent perturber la stabilité du signal de sortie du PUF. Par conséquent, des variations peuvent entraîner des réponses incorrectes, différentes de la réponse initiale [51]. Ces variations causées par les modifications environnementales peuvent être corrigées en utilisant la méthode de l'extracteur flou [52].

Le fuzzy extractor, qui peut extraire des informations stables et répétables à partir de la réponse du PUF, consiste en une paire d'algorithmes. Les deux algorithmes sont la fonction de génération $\mathrm{Gen}(.)$ et la fonction de reproduction $\mathrm{Rep}(.,.)$, qui sont définies comme suit :

1) Gen : Pour la paire de défi-réponse (C_i, R_i) générée par $R_i = \mathrm{PUF}(C_i)$, $\mathrm{Gen}(.)$ produit un tuple constitué d'une chaîne aléatoire (référée comme la clé secrète) X_i et des données auxiliaires P_i, c'est-à-dire, $\mathrm{Gen}(R_i) = (X_i, P_i)$.

2) Rep : Étant donné une sortie du PUF, notée comme la réponse actuelle R_i', la clé originale K_i est ensuite récupérée en utilisant les données auxiliaires P_i, c'est-à-dire, $\text{Rep}(R_i', P_i) = X_i$.

3.2.5 Améliorations proposées dans IoV

Les systèmes d'authentification classiques utilisant des paires bilinéaires ou le chiffrement par identité sont très performants en termes de calcul, et il est compliqué de garantir la sécurité de l'identité et de la vie privée des utilisateurs s'ils sont directement intégrés dans l'environnement IoV. Dans cette situation, il est nécessaire de concevoir des mécanismes d'authentification légers, adaptés à l'environnement IoV.

Jiang et al. [25] ont présenté un protocole d'authentification mutuelle léger basé sur les PUF, qui génère trois clés de session : véhicule-utilisateur, véhicule-centre de données, et utilisateur-centre de données. Nous avons réalisé une analyse cryptographique approfondie qui démontre plusieurs vulnérabilités dans le protocole tel que écoute clandestine. Un attaquant peut extraire la clé de session partagée entre deux parties légitimes, à savoir le capteur de véhicule et l'utilisateur, ou entre le capteur de véhicule et le centre de données. Par conséquent, cela compromet la confidentialité des messages en raison de la simplicité avec laquelle un adversaire peut déduire la clé de session. De plus, des erreurs dans le calcul des clés de session sont présentes, sans oublier l'utilisation inappropriée des horodatages. Pour que ce protocole puisse être utilisé, il est nécessaire de proposer une amélioration pour remédier à tous les problèmes mentionnés. Cette solution est développée dans l'article intitulé "Security analysis on Three-factor authentication protocol using physical unclonable function for IoV.", présenté au chapitre 4.

Un autre travail de Srinivas et al. [48], qui n'utilise pas les PUF, propose l'ajout d'une phase supplémentaire nommée "Phase de Validation des Crédentiels des Dispositifs IoT", a été introduite. Elle est particulièrement nécessaire lorsque le cloud-gateway (CG) vérifie périodiquement que les crédentiels stockés dans un nœud IoT sont toujours valides après son déploiement dans le réseau. De plus, cette phase permet également de s'assurer que les nœuds IoT fonctionnent conformément aux attentes du CG, afin de détecter tout comportement malveillant des nœuds IoT. Cependant, après une analyse approfondie de ce protocole, nous avons découvert qu'il présente une vulnérabilité affectant l'authentification entre les utilisateurs légitimes et les dispositifs IoT installés dans les véhicules. Nous avons proposé une amélioration de ce protocole, détaillée dans l'article intitulé "Enhancement of a User Authentication Scheme for Big Data Collection in IoT-Based Intelligent Transportation System", et expliquée dans le chapitre 5.

3.2.6 Évaluation de la sécurité

Pour démontrer la sécurité prouvée, plusieurs approches de vérification de la sécurité peuvent être utilisées. Comme l'illustre la figure 3.2, notre recherche adopte plusieurs modèles de menace, ainsi que des analyses informelles de sécurité.

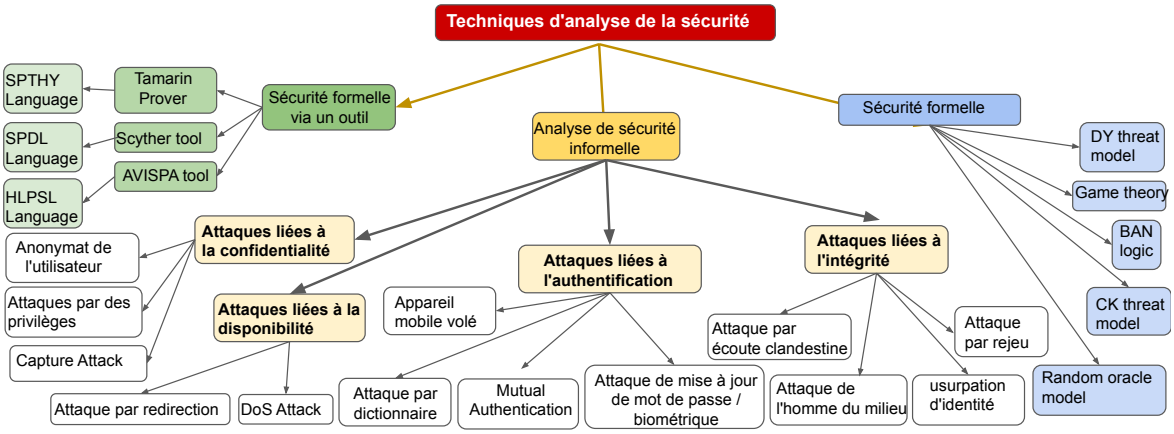


FIGURE 3.2 : Techniques de vérification de la sécurité

3.2.7 Évaluation des performances

L'analyse des performances d'un protocole est essentielle pour vérifier son adéquation dans un environnement IoV/IoD, souvent limité par des contraintes de calcul et de bande passante. Afin d'analyser ces performances, on utilise principalement deux indicateurs : le coût total de calcul et de communication.

Le coût de calcul, mesuré en millisecondes, correspond au temps nécessaire pour que le véhicule, l'utilisateur et le serveur effectuent les opérations définies par le protocole d'authentification. C'est un indicateur crucial pour mesurer la capacité du système embarqué à accomplir rapidement les tâches de sécurité requises. Il garantit que le protocole d'authentification ne perturbe pas la mobilité rapide des véhicules, permettant ainsi une authentification en temps réel sans compromettre leur fonctionnement ou leur mission. Par ailleurs, il est crucial de considérer la durée totale d'exécution du protocole, notamment lors des phases d'authentification.

Le coût de communication, mesuré en bits, représente la quantité totale de données échangées sur le canal pour l'ensemble des interactions du protocole. Grâce à l'analyse de ce flux de données, il est possible d'évaluer l'influence du protocole sur l'exploitation des ressources réseau. Il est nécessaire de minimiser le nombre de bits échangés à chaque interaction afin d'assurer la performance du protocole et d'optimiser la consommation des ressources du réseau.

3.3 Volet 2 : Conception du protocole d'authentification pour IoD

L'Internet des drones (IoD), en tant qu'architecture de coordination automatisée des drones, offre des solutions pour pallier les problèmes auxquels l'Internet des véhicules (IoV) est confronté, tels que les défis liés à la couverture limitée des stations de base et des unités en bordure de route. Grâce à la

technologie IoD, les drones peuvent être déployés pour surveiller des situations critiques, effectuer des ajustements rapides en fonction des conditions en temps réel, et ainsi améliorer la gestion des flux, que ce soit pour le trafic routier ou d'autres usages.

3.3.1 Étude de cas dans l'IoD

La mise en œuvre de l'IoD n'est pas une tâche facile, car elle présente de nombreuses contraintes qui peuvent poser de graves menaces à la vie si elles ne sont pas gérées de manière adéquate. Étant donné la nature ouverte des réseaux IoD, la sécurité et la préservation de la vie privée des utilisateurs demeurent des objectifs essentiels. Différentes catégories d'attaques ont déjà été rapportées dans ces réseaux, ce qui a mené de nombreux chercheurs à proposer divers schémas d'authentification utilisant différentes techniques cryptographiques. Bien que ces schémas puissent partiellement répondre à l'objectif visé, une analyse approfondie met en évidence que les études réalisées sur l'authentification des messages pour des applications de sécurité explicites sont très restreintes. Ces recherches contiennent des défis tels que les coûts élevés de calcul et de communication, les authentifications redondantes ou erronées, les vulnérabilités aux attaques et les écoutes clandestines, ce qui nécessite une exploration plus approfondie.

Par ailleurs, l'adoption massive des drones a également soulevé des préoccupations majeures en matière de sécurité et de protection de la vie privée. En effet, des incidents d'accès non autorisé par des drones amateurs ont récemment été signalés par des opérateurs d'infrastructures critiques, telles que des zones militaires et des aéroports, entraînant de graves problèmes de sécurité, de confidentialité et de sûreté. Pour cette raison, la Federal Aviation Administration (FAA) et les autorités européennes ont récemment établi une réglementation connue sous le nom d'Identification à Distance (RemoteID). Cette réglementation exige que tous les drones diffusent périodiquement des messages rapportant leur identité et des détails sur la station de base. La conformité à la réglementation RemoteID est devenue obligatoire à partir de septembre 2022. Bien que répondant aux exigences des opérateurs d'infrastructures critiques, la réglementation RemoteID a suscité des préoccupations notables au sein de la communauté des drones. Récemment, des communautés de drones amateurs ont également déposé une plainte contre la FAA, invoquant des problèmes de confidentialité liés à la mise en œuvre obligatoire de RemoteID.

Pour répondre à ces enjeux, ce travail se concentre sur deux objectifs principaux, détaillés ci-dessous.

1. Assurer une communication sécurisée entre un utilisateur et un drone situé dans une zone spécifique, tout en tenant compte de la durée de vie du drone, de ses contraintes de calcul, de communication, ainsi que de ses limitations énergétiques, est un défi majeur. Cet objectif est traité dans l'article intitulé "LASeR : Lightweight and Secure Remote User Authentication Protocol for Internet of Drones", présenté dans le chapitre 6.

2. Garantir une authentification anonyme et la vérification de l'autorité d'accès pour les essaims de

drones, tout en prenant en compte les exigences de la récente réglementation FAA RemoteID, constitue un défi significatif. Cette problématique est abordée dans notre article intitulé "2AS-DS : Anonymous Authentication Scheme Based on Physical Unclonable Function for Drone Swarms", présenté dans le chapitre 7.

3.3.2 Évaluation de la sécurité

Pour l'évaluation de la sécurité, nous utilisons les deux modèles de menace : Dolev-Yao et Canetti-Krawczyk, détaillés dans la section 3.1.2. De plus, nous appliquons trois outils d'analyse de sécurité formelle : Scyther, AVISPA et Tamarin.

En outre, nous considérons les capacités suivantes que l'attaquant A pourrait avoir :

- Le smartphone peut être perdu ou volé, ce qui le rend donc peu fiable, et peut également être physiquement capturé par A. De plus, A a la capacité d'extraire tous les paramètres stockés sur un appareil mobile via des attaques d'analyse de puissance [55].
- A est capable de deviner hors ligne l'identité de l'utilisateur et le mot de passe [56].
- L'utilisateur légitime peut également agir en tant qu'attaquant.
- A peut s'infiltrer en tant qu'initié et obtenir le vérificateur du serveur de la station au sol.
- A peut capturer physiquement le drone et extraire les informations qu'il contient.

3.3.3 Outils de Vérification de Sécurité : AVISPA, Scyther et Tamarin

AVISPA Tool : AVISPA est un outil indépendant qui simule les applications et protocoles de sécurité en utilisant le langage de spécification de protocole de haut niveau (HLPSL) et détermine si les procédures de sécurité proposées sont sûres, non sûres ou indécises pour la communication. Sous les hypothèses du modèle DY (décrit en Section 3.2), AVISPA prouve la sécurité du cadre proposé pour des objectifs prédéfinis de confidentialité et d'authentification. De plus, AVISPA analyse la sécurité contre les attaques liées à la confidentialité et à l'authentification, telles que les attaques de falsification, les attaques de l'homme du milieu, le vol d'identité et les attaques par rejeu. Les résultats de simulation d'AVISPA indiquent si le schéma proposé est sûr et protégé contre de telles attaques ou non.

Le langage HLPSL orienté rôles comporte deux types de rôles : les rôles de base représentent divers participants au protocole, tandis que les rôles de composition représentent différents scénarios incorporant les rôles de base. HLPSL fonctionne en construisant un rôle de base pour chaque nœud de communication. Chaque rôle inclut des transitions pour ses différents états, mises en œuvre à travers deux rôles essentiels : session et environnement. Un intrus est souvent représenté par l'un des participants du protocole. En d'autres termes, il adopte l'un des rôles de base autorisés, désigné par

rôle (i). La spécification du schéma écrite en HLPSL est transformée en un format intermédiaire via le traducteur HLPSL2IF. Le format intermédiaire est ensuite transmis via l'un des quatre moteurs de SPAN pour produire le format de sortie.

Le format de sortie comprend les parties suivantes :

-SUMMARY (Résumé) : indique si le schéma analysé est sûr ou non. Parfois, cette section rapporte que l'analyse est indécise, ce qui se produit lorsque le moteur sélectionné ne prend pas en charge les spécifications du schéma.

-DETAILS (Détails) : explique les raisons du résultat, qu'il s'agisse d'un schéma trouvé sûr, vulnérable à une attaque, ou d'une analyse indécise.

-PROTOCOL (Protocole) : désigne le schéma ciblé, écrit en spécifications HLPSL.

-GOAL (Objectif) : indique l'objectif analytique à réaliser par AVISPA.

-BACK-END (Moteur) : décrit le moteur sélectionné pour l'analyse de sécurité : l'un de TA4SP, CL-AtSe, OFMC, ou SATMC. cette partie contient des commentaires et des statistiques pertinentes.

Dans le cadre de la thèse, nous avons utilisé deux moteurs, OFMC et CL-AtSe, qui sont efficaces pour détecter les attaques et évaluer les protocoles de sécurité. Ces moteurs garantissent que le schéma proposé est protégé contre les attaques courantes, telles que les attaques de l'homme du milieu et les attaques par rejeu. Les deux autres moteurs, SATMC et TA4SP, ne prennent pas en charge les opérations du schéma proposé, donc leurs résultats de simulation ne sont pas pris en compte.

Simulateurs formels de sécurité Scyther Tool : Scyther est un outil efficace pour la vérification, la falsification et l'analyse des protocoles de sécurité afin de détecter les attaques potentielles et les vulnérabilités. Il peut analyser les protocoles de sécurité en utilisant les assertions de Scyther avec une approche de vérification de modèle non borné et une technique de recherche d'états symboliques en arrière. L'outil Scyther utilise le langage de description des protocoles de sécurité (SPDL) pour spécifier et analyser les protocoles de sécurité. Le langage d'entrée de Scyther permet de spécifier de nombreuses propriétés de sécurité sous forme d'un ensemble d'assertions. Par exemple, on peut affirmer qu'une certaine valeur est confidentielle dans la spécification de rôle, c'est-à-dire une assertion de secret, ou que certaines propriétés doivent être respectées par les deux partenaires de communication, c'est-à-dire une assertion d'authentification. Pour l'authentification, Scyther fournit quatre assertions différentes, notamment la vivacité (Alive), l'accord faible (Weakagree), l'accord (Niagree) et la synchronisation (Nisynch) pour détecter les attaques par rejeu, réflexion, et l'homme du milieu, entre autres. Ici, la synchronisation est utilisée pour examiner les attaques par suppression de rejeu, ce qui est une propriété strictement plus forte que l'accord dans le modèle standard d'intrus. L'outil Scyther est le seul actuellement capable de vérifier la synchronisation parmi les outils existants. Dans l'outil Scyther, en plus du modèle standard de Dolev-Yao, 9 autres modèles d'adversaires couramment utilisés, tels que le modèle CK, le modèle eCK, etc., sont entièrement pris en charge pour démontrer la sécurité des protocoles d'authentification et d'échange de clés. En particulier, tous les protocoles que nous avons

proposés sont vérifiés en utilisant l'outil Scyther.

Simulateurs formels de sécurité Tamarin prover : Tamatin est un outil puissant pour la modélisation symbolique et l'analyse des protocoles de sécurité. Il prend en entrée un modèle de protocole de sécurité, qui inclut les actions effectuées par les agents dans différents rôles, tels que l'initiateur du protocole, le répondeur et le serveur de clés de confiance. L'outil nécessite également des spécifications de l'adversaire et des propriétés souhaitées du protocole. TAMARIN peut être utilisé pour générer automatiquement des preuves démontrant la capacité du protocole à maintenir ses propriétés spécifiées, même en présence de plusieurs instances des rôles du protocole fonctionnant en parallèle avec les actions de l'adversaire. l'outil Tamarin est utilisé pour vérifier différentes propriétés de sécurité d'un système :

- **Disponibilité** : Tamarin peut vérifier la validité du schéma pour un nombre illimité de sessions. Cela signifie que l'outil peut démontrer que le protocole fonctionne correctement, même lorsque de nombreuses sessions sont établies simultanément.
- **Confidentialité** : Tamarin vérifie cet aspect avec le lemme User SK secrecy, ce qui signifie que l'outil prouve que la clé secrète d'un utilisateur reste protégée.
- **Authentification** : L'outil utilise le lemme User auth pour vérifier que les parties authentifiées sont réellement celles qu'elles prétendent être, assurant ainsi que les utilisateurs sont correctement identifiés.
- **Non-répudiation** : La non-répudiation est assurée par l'authentification et la fraîcheur de la communication.
- **Détection des attaques MITM** : L'outil utilise des règles pour vérifier que les canaux de communication sont sécurisés, ce qui inclut la détection des attaques MitM et assure que les communications sont protégées.
- **Détection des attaques par rejeu** : Cela est vérifié via Injective agreement, une règle qui garantit la fraîcheur du message. Elle permet de s'assurer que le message n'est pas un ancien message réutilisé, et qu'il est toujours valide et actuel.

3.3.4 Évaluation des performances

Dans cette section, nous évaluons les performances en tenant compte du coût de calcul, du coût de communication, que nous avons détaillé dans la section 3.1.3, ainsi que de la consommation énergétique, un facteur crucial pour les drones qui sont des dispositifs limités en termes d'énergie.

Calcul de la consommation énergétique : Nous analysons la consommation énergétique liée aux différentes fonctions cryptographiques effectuées par les drones. Pour ce faire, l'énergie consommée

est calculée à l'aide de la formule [57] suivante :

$$\text{Énergie} = cur \times vol \times \text{temps}$$

où le courant (*cur*) est exprimé en ampères (A) et le voltage (*vol*) en volts (V). Dans notre cas, nous considérons les spécifications du Raspberry Pi 3, utilisé ici comme référence pour les drones, avec un courant de 2500 mA et un voltage de 3,3 V. Le paramètre T représente le temps nécessaire pour exécuter une primitive cryptographique spécifique. Cette analyse permet de mieux comprendre l'impact énergétique des opérations cryptographiques sur les drones et d'optimiser leur efficacité en termes de consommation d'énergie.

3.4 Conclusion

Les travaux de cette thèse, développés au cours des quatre chapitres suivants, ont permis d'atteindre les objectifs fixés dans la section 1.3. Le premier objectif spécifique a été accompli grâce aux recherches menées dans le cadre du premier volet, détaillées dans les chapitres 4 et 5. Quant au deuxième et au troisième objectif spécifique, ils ont été atteints à travers le deuxième volet, dont les résultats sont exposés dans les chapitres 6 et 7. L'ensemble de ces objectifs spécifiques focalisés vers l'objectif principal : améliorer la sécurité dans l'Internet des objets, plus précisément IoV et IoD.

Chapitre 4

Analyse de la sécurité et amélioration du protocole d'authentification à trois facteurs utilisant fonction physique non clonable pour l'Internet des véhicules.

Dans ce chapitre, nous abordons l'impact de l'avènement de l'IoT sur l'Internet des véhicules (IoV). L'IoV a permis le développement de nombreuses applications intelligentes pour la gestion et la sécurité du trafic, en offrant aux véhicules la capacité de gérer de manière autonome les situations imprévues grâce au partage de ressources telles que des informations critiques. Toutefois, en tant que technologie reposant sur les infrastructures de réseaux sans fil, l'IoV présente plusieurs vulnérabilités, le rendant ainsi susceptible à diverses attaques menaçant la sécurité et la confidentialité des utilisateurs.

Nous analysons dans ce chapitre le protocole de Jiang et al. [25], mettant en évidence les limites des garanties de sécurité, notamment en ce qui concerne la possibilité pour un attaquant de déduire les clés de session partagées entre le capteur du véhicule et le centre de données, ainsi qu'entre le capteur du véhicule et l'utilisateur. En réponse à ces failles, nous proposons une amélioration du protocole afin de renforcer la sécurité et d'éliminer les vulnérabilités identifiées.

4.1 Introduction

L'Internet des Véhicules (IoV) a dominé les systèmes de transport intelligents (ITS) en raison de nombreuses caractéristiques, telles que la compatibilité avec les appareils portables, la grande échelle du réseau, la capacité de traitement élevée et les systèmes à topologie dynamique, etc. Il est particulièrement crucial pour les véhicules autonomes en raison de leur capacité à prendre des décisions de

conduite correctes. L'IoV met l'accent sur l'échange d'informations entre les véhicules, les unités routières et les humains. Son principal objectif est de faciliter l'obtention d'informations en temps réel sur le trafic, d'améliorer le confort de voyage et de préserver la commodité des déplacements. L'IoV dépend des technologies informatiques actuelles ainsi que des communications sans fil, ce qui offre une opportunité à un adversaire d'extraire, de supprimer, d'insérer ou de modifier des données pendant la communication. Les véhicules se déplacent fréquemment d'une zone à une autre, ce qui entraîne la création et l'arrêt continuel de nouvelles connexions. Cette mobilité rapide des véhicules nécessite une authentification à chaque nouvelle connexion, car l'authentification est la première ligne de défense pour garantir la sécurité, la confidentialité et l'intégrité des données. À partir de la section 2.2, où nous avons examiné les différents problèmes de sécurité et d'authentification dans un environnement IoV, nous avons mené une analyse comparative des performances des protocoles existants. Il en ressort que le protocole proposé par Jiang et al. est extrêmement léger et permet la création de trois clés de session entre les principaux acteurs : le véhicule, l'utilisateur, et le centre de données. Jiang et al. ont affirmé que leur protocole était résistant à divers types d'attaques. Cependant, nous démontrons ici que le schéma de Jiang et al. présente des vulnérabilités, notamment qu'un attaquant peut extraire la clé de session partagée entre deux parties légitimes, à savoir le capteur du véhicule et l'utilisateur, ainsi que le capteur du véhicule et le centre de données. De plus, des erreurs dans le calcul des clés de session ont été identifiées. Nous proposons donc une amélioration pour remédier à toutes ces failles de sécurité.

Le reste du chapitre est organisé comme suit : la section 4.2 présente une brève discussion sur le modèle de système, ainsi que sur la procédure d'authentification et d'échange de clés. La section 4.3 examine le schéma de Jiang et al., tandis que la section 4.4 expose le problème du schéma de Jiang et al. et propose des corrections. La section 4.5 décrit notre cryptanalyse des schémas de Jiang et al. Enfin, la section 4.6 présente notre amélioration du protocole de Jiang et al.

4.2 Le modèle de système

Le réseau IoV (Internet of Vehicles) peut être considéré comme un système composé de composants hétérogènes, y compris un grand nombre de nœuds de véhicules et d'infrastructures. Grâce à son réseau, l'IoV garantit la connexion, l'interaction et l'échange de données entre ses composants. Ces interactions entre les différents composants permettent aux véhicules d'accéder à des informations sur le trafic, ce qui améliore réellement la sécurité routière et l'efficacité des transports [58]. Jiang et al. [25] ont proposé une authentification à trois facteurs utilisant une fonction physiquement in-clonable (PUF) pour l'IoV est présentée. Ses principaux composants sont le Centre de Données (DC), le Capteur de Véhicule (VSk), et l'Utilisateur (Ui). Les principales tâches du DC sont l'enregistrement et la gestion des données de trafic en temps réel collectées par le VSk, qui est déployé dans le véhicule. Pour gérer le véhicule ou planifier l'itinéraire, chaque Ui reçoit respectivement des données en temps réel et des données statistiques du VSk et du DC. Comme illustré à la figure 4.1, l'ensemble de l'authentification

comprend quatre étapes. Après le processus d'authentification, les fournisseurs peuvent partager une clé de session privée entre le Capteur de Véhicule et le Centre de Données, entre le Capteur de Véhicule et l'Utilisateur, ainsi qu'entre le Centre de Données et l'Utilisateur, afin d'assurer la confidentialité et l'intégrité des données transmises par cryptage et décryptage des communications ultérieures.

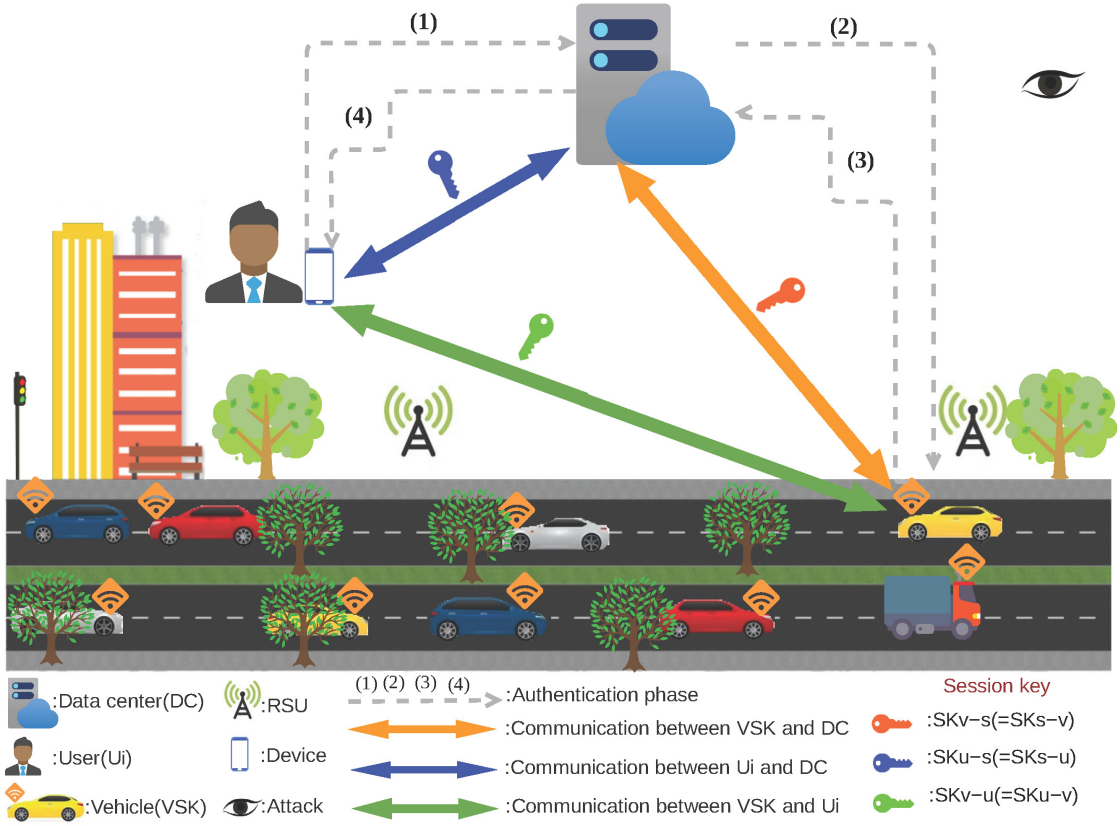


FIGURE 4.1 : Modèle de système et phase d'authentification

TABLE 4.1 : Notations utilisées dans notre protocole amélioré pour l'IoV

Symbole	Description
UID_i, ID_s, VID_k	L'identification de U_i, DC et du véhicule, respectivement.
PW_i, BIO_i	Le mot de passe et les données biométriques de U_i.
UF_i, VF_k	Le PUF de U_i et VS_k, respectivement.
(UC_i, UR_i), (VC_j, VR_j)	Le couple challenge-réponse du PUF de U_i et VS_k.
S, s	La paire de clés publique/privée de DC.
VPk	La clé publique de VS_k.
Gen$(\cdot)$, Rep$(\cdot)$	Fonction de génération probabiliste et de reproduction déterministe.
$h(\cdot)$	La fonction de hachage cryptographique unidirectionnelle.
T_i	Les horodatages utilisés dans la phase de connexion et d'authentification.
$\|$	L'opération de concaténation.
$\oplus$	L'opération XOR bit à bit.

4.3 Revue du Schéma de Jiang et al.

Le schéma proposé par Jiang et al. utilise une sécurité à trois facteurs : ils ont combiné les PUF, les données biométriques et le mot de passe, ces trois facteurs étant principalement implémentés du côté utilisateur. Du côté des capteurs de véhicules, un extracteur flou est utilisé pour traiter le bruit de la réponse des PUF. Du côté utilisateur, un engagement flou est introduit pour traiter le bruit des réponses des PUF et des données biométriques. Trois types de parties sont impliquées dans ce schéma.

- **Utilisateur (U_i)** : Reçoit des données en temps réel des capteurs de véhicules et des données statistiques du centre de données pour la gestion des véhicules ou la planification des itinéraires.
- **Centre de données (DC)** : Gère et stocke les données collectées par les capteurs des véhicules.
- **Capteur de véhicule (VS_k)** : Déployé dans le véhicule pour collecter des données de trafic en temps réel.

Ce protocole comprend cinq phases principales, mais dans notre analyse de sécurité de ce schéma, nous nous intéressons aux trois phases principales : la configuration du système, l'enregistrement, et la connexion et l'authentification. Les notations clés utilisées sont listées dans le tableau tableau 4.1.

4.3.1 Configuration du Système

Dans cette phase, DC choisit d'abord une valeur secrète s et un point P sur la courbe elliptique $y^2 = x^3 + ax + b$, où a, $b \in \mathbb{Z}_p$ satisfont $4a^3 + 27b^2 \neq 0 (\text{mod } p)$, p est un grand nombre premier. Ensuite, DC calcule sa propre clé publique $S = s \cdot P$.

4.3.2 Phase d'Enregistrement

4.3.2.1 Enregistrement de l'Utilisateur

Pour rejoindre le système IoV, un utilisateur doit s'enregistrer auprès de DC pour être légitime, comme indiqué dans la figure **??**. Voici les étapes :

1. U_i saisit l'identifiant UID_i dans le dispositif, et le dispositif envoie l'identification de l'utilisateur et la demande d'enregistrement $\langle UID_i, \text{Req}_{\text{reg}} \rangle$ à DC via un canal sécurisé.

2. Dès réception du message de U_i, DC recherche dans sa base de données pour vérifier si U_i est déjà enregistré. Sinon, il génère un défi UC_i pour U_i, et le transmet à U_i. Sinon, il envoie un message pour alerter que l'utilisateur est déjà enregistré.

3. Après avoir reçu UC_i, U_i sélectionne le mot de passe PW_i, l'entre dans le dispositif et y imprime les données biométriques BIO_i. Ensuite, le dispositif de l'utilisateur calcule $UR_i = UF_i(UC_i)$ et utilise la technologie de fusion basée sur l'engagement flou pour fusionner les données biométriques de l'utilisateur avec celles du dispositif $w = BIO_i \oplus UR_i$. De plus, le dispositif choisit une valeur secrète K_i et calcule $UK_i = h(K_i)$, $W_i = UC_i \oplus h(UID_i \parallel PW_i)$, et $V_i = h(h(PW_i \parallel UK_i) \bmod l)$. Enfin, le dispositif envoie UK_i à DC.

4. DC stocke l'élément $\{UID_i, UK_i\}$ dans sa base de données pour une authentification ultérieure.

4.3.2.2 Enregistrement des Capteurs de Véhicules

L'identité du véhicule VID_k est utilisée comme défi du PUF dans cette procédure.

1. Le capteur du véhicule VS_k envoie VID_k et une demande d'enregistrement à DC pour l'enregistrement.

2. Dès réception de la demande, DC recherche également dans sa base de données pour vérifier si le véhicule est enregistré. Si ce n'est pas le cas, il envoie un message à VS_k pour demander des données de vérification à stocker. Sinon, il rejette la demande.

3. VS_k prend VID_k comme défi du PUF et extrait la sortie du PUF $VR_k = VF_k(VID_k)$. Ensuite, VS_k utilise un extracteur flou pour traiter le bruit de la réponse du PUF et générer des données d'assistance $(VK_k, VP_k) = \text{Gen}(VR_k)$ et calcule $VQ_k = VK_k \cdot P$. Enfin, VS_k transmet $\{VP_k, VQ_k\}$ à DC.

4. DC stocke l'élément $\{VID_k, VP_k, VQ_k\}$ dans sa base de données pour une authentification ultérieure.

4.3.3 Phase de Connexion et d'Authentification

Pour accéder à DC et VS_k, un utilisateur U_i se connecte d'abord et obtient l'autorisation du DC, puis établit des clés de session avec DC et VS_k, respectivement.

1. U_i saisit d'abord l'identifiant UID_i', le mot de passe PW_i', et les données biométriques BIO_i' dans le dispositif. Le dispositif récupère d'abord le défi PUF $UC_i' = W_i \oplus h(UID_i' \parallel PW_i')$ et utilise le PUF intégré pour calculer la réponse $UR_i' = UF_i(UC_i')$. Par la suite, le dispositif utilise la technologie de fusion biométrique basée sur l'engagement flou pour fusionner les données biométriques de l'utilisateur avec celles du dispositif $w' = BIO_i' \oplus UR_i'$, $K_i' = \text{Decoder}(UP_i \oplus w')$ et calcule la valeur secrète $UK_i' = h(K_i')$ pour vérifie localement si la valeur calculée de $h(h(PW_i' \parallel UK_i') \mod l)$ est égale à la valeur stockée V_i.

2. Si la valeur est égale, l'appareil de l'utilisateur sélectionne un nombre aléatoire UN_i pour calculer sa clé publique temporaire $UQ_i = UN_i \cdot P$ et la clé temporaire $B = UN_i \cdot S = (B_x, B_y)$ partagée avec le centre de données (DC). Ensuite, l'appareil commence à calculer les pseudo-identités de l'utilisateur et du véhicule, $DUID_i = UID_i' \oplus B_x$ et $DVID_k = VID_k \oplus B_y$, et génère un message de vérification $V_1 = h(UID_i' \parallel ID_s \parallel VID_k \parallel UK_i' \parallel T_1 \parallel B_x)$, où VID_k est l'identité du véhicule auquel U_i souhaite accéder et T_1 est l'horodatage actuel. Enfin, l'appareil envoie le message $M_1 = \langle UQ_i, DUID_i, DVID_k, V_1, T_1 \rangle$ à DC.

3. Après réception du message de l'appareil de U_i, DC vérifie d'abord la validité temporelle du message. Ensuite, la clé temporaire $B' = s \cdot UQ_i = (B_x', B_y')$ est calculée, et les identifications de l'utilisateur et du véhicule sont restaurées $UID_i' = DUID_i \oplus B_x'$ et $VID_k' = DVID_k \oplus B_y'$. DC récupère ensuite UK_i dans la base de données en fonction de UID_i'. Si U_i est introuvable, la session est immédiatement terminée. Sinon, DC calcule $h(UID_i' \parallel ID_s \parallel VID_k' \parallel UK_i \parallel T_1 \parallel B_x')$. Si la valeur calculée est égale à celle stockée, DC récupère VP_k et VQ_k dans la base de données et calcule $J = s \cdot VQ_k = (J_x, J_y)$. Ensuite, DC calcule la pseudo-identité de U_i, $DUID_i' = UID_i' \oplus h(VID_k' \parallel ID_s \parallel J_x \parallel T_2)$, ainsi que le matériel de clé $DUK_i' = h(UID_i' \parallel UK_i \parallel T_2) \oplus h(J_x \parallel T_2)$. Enfin, DC envoie le message $M_2 = \langle DUID_i', DUK_i, DSK_s, V_2, T_2 \rangle$ au capteur du véhicule V_Sk.

4. Après réception du message, V_{Sk} vérifie immédiatement la validité temporelle du message. Si cela est satisfait, V_{Sk} utilise VID_k' pour calculer $VR_k' = VF_k(VID_k')$. Ensuite, V_{Sk} utilise l'extracteur flou pour traiter le bruit de la réponse PUF, $(VK_k', VP_k') = Gen(VR_k')$, et calcule la clé temporaire $J' = VK_k' \cdot S = (J_x', J_y')$. Ensuite, V_{Sk} récupère l'identité de l'utilisateur $UID_i'' = DUID_i' \oplus h(VID_k' \parallel ID_s \parallel J_x' \parallel T_2)$ et calcule $h(VID_k \parallel ID_s \parallel UID_i'' \parallel J_x' \parallel J_y' \parallel T_2)$ pour juger si la valeur calculée est égale à l'identité reçue. Si elles sont égales, l'identité de *DC* est authentifiée. Sinon, V_{Sk} refuse *DC* et termine la session.

5. V_{Sk} calcule la clé de session $SK_{V-U} = h(h(DUK_i \oplus h(J_x' \parallel T_2)) \parallel h(VID_k' \parallel T_3 \parallel VK_k'))$ partagée avec U_i et *DC*. Ensuite, V_{Sk} calcule la réponse pseudo $DVR_k = VR_k' \oplus J_x'$, la valeur de vérification

$V_3 = h(ID_s \| VID'_k \| J'_y \| SK_{V-S} \| T_3)$, et $V_4 = h(UID''_i \| VID'_k \| VK'_k \| SK_{V-U} \| T_3)$, où V_4 sera envoyé à U_i par DC. Enfin, V_{Sk} envoie le message à DC.

6. Après réception du message de V_{Sk}, DC vérifie la validité temporelle du message. Ensuite, DC calcule $VR''_k = DVR_k \oplus J_x$, et $VK''_k = Rep(VP_k, VR''_k)$. Ensuite, DC utilise la valeur secrète calculée de V_{Sk} et sa propre valeur secrète s pour calculer la clé de session partagée $SK_{S-V} = h(h(ID_s \| s \| T_2) \| h(VID'_k \| VK''_k \| T_3))$ et la valeur V_3. Ensuite, DC vérifie l'identité de V_{Sk} et la validité de la clé en jugeant si la valeur calculée est égale à V_3. Si elles sont égales, DC calcule la clé de session $SK_{S-U} = h(h(ID_s \| T_4 \| s) \| h(UID'_i \| UK_i \| T_4))$ partagée avec U_i et une valeur de vérification $V_5 = h(ID_s \| UID'_i \| B'_y \| SK_{S-U} \| T_4)$. Enfin, DC envoie le message $M_4 = \langle AVK_k, DVK_k, DSK_k, V_4, V_5, T_2, T_3, T_4 \rangle$ à U_i.

7. Après réception du message, U_i vérifie d'abord la validité temporelle du message. Si c'est le cas, U_i calcule la clé de session $SK_{U-S} = h((DSK_s \oplus h(UK'_i \| T_4)) \| h(UID'_i \| UK'_i \| T_4))$ partagée avec DC et $h(ID_s \| UID'_i \| B_y \| SK_{U-S} \| T_4)$ pour déterminer la validité de DC et de la clé de session. Si la valeur calculée est égale à V_5, U_i récupère la valeur secrète de V_{Sk}, $VK''_k = AVK_k \oplus B_y$, et calcule la clé de session partagée $SK_{U-V} = h(h(UID'_i \| UK'_i \| T_4) \| (DVK_k \oplus h(T_4 \| UK'_i)))$ ainsi que le message de vérification $h(UID'_i \| VID_k \| VK''_k \| SK_{V-U} \| T_3)$, et confirme l'identité de V_{Sk} et la validité de la clé de session En étant sûr que la valeur calculée correspond à V_4.

4.4 Énoncé du Problème du Schéma de Jiang et al.

Dans notre analyse du protocole proposé par Q. Jiang et al., nous avons relevé des erreurs dans le calcul des clés de session, ainsi qu'une utilisation inappropriée des horodatages. Afin d'organiser clairement notre travail, les corrections apportées au protocole sont présentées dans la section 4.4.1, tandis que la cryptanalyse du protocole d'authentification est détaillée dans la section 4.4.2.

Lors de la phase d'authentification du protocole de Q. Jiang et al. (section 4.3.3), nous avons identifié des erreurs concernant la description des clés de session, notamment entre VS_k et DC, ainsi qu'entre VS_k et U_i.

1) Démonstration que $SK_{V-S} \neq SK_{S-V}$:

À l'étape 5 : VS_K calcule incorrectement la clés de session

$$SK_{V-S} = h(h(DSK_s \oplus h(J'_y \| T_2)) \| h(VID'_k \| VK'_k \| T_3))$$

À l'étape 6 : DC calcule

$$SK_{S-V} = h(h(ID_s \| s \| T_2) \| h(VID'_k \| VK''_k \| T_3))$$

La vérification des clés de session entre le capteur de véhicule et le centre de données montre que l'utilisation de valeurs incorrectes conduit finalement à la génération de fausses clés de session, telles que $SK_{V-S} \neq SK_{S-V}$, comme démontré ci-dessous.

$SKV-S \stackrel{?}{=} SKS-V$

$h\left(h(DSK_s \oplus h(J_y''||T2))||h(VID_k'||VKK_k''||T3)\right) \stackrel{?}{=} h\left(h(ID_s||s||T2)||h(VID_k'||VKK_k''||T3)\right)$

$h\left(h(h(ID_s||s||T2) \oplus h(J_y||T2) \oplus h(J_y''||T2))||h(VID_k'||VKK_k''||T3) \stackrel{?}{=} h\left(h(ID_s||s||T2)||h(VID_k'||VKK_k''||T3)\right)\right.$

$h\left(h(h(ID_s||s||T2) \oplus \emptyset)||h(VID_k'||VKK_k''||T3)\right) \stackrel{?}{=} h\left(h(ID_s||s||T2)||h(VID_k'||VKK_k''||T3)\right)$

$h\left(h(h(ID_s||s||T2))||h(VID_k'||VKK_k''||T3)\right) \neq h\left(h(ID_s||s||T2)||h(VID_k'||VKK_k''||T3)\right)$

2) Démonstration que $SK_{V-U} \neq SK_{U-V}$:

À l'étape 5 : VS_K calcule incorrectement la clés de session

$$SK_{V-U} = h(h(DUK_i \oplus h(j_x' \parallel T_2)) \parallel h(VID_k' \parallel T_3 \parallel VK_K'))$$

À l'étape 7 : Ui calcule incorrectement la clé de session

$$SK_{U-V} = h(h(UID_i' \parallel UK_i' \parallel T_4) \parallel (DVK_k \oplus h(T_4 \parallel UK_i')))$$

La vérification des clés de session entre l'utilisateur et le capteur de véhicule. Si on utilisent la valeur incorrecte, on généreront finalement de fausses clés de session, telles que $SK_{V-U} \neq SK_{U-V}$, comme démontré ci-dessous.

$SKV-U \stackrel{?}{=} SKU-V$

$h\left(h\left(DUK_i \oplus h(J_x''||T2)\right)||h(VID_k''||T3||VKK_k')\right) \stackrel{?}{=} h\left(h(UID_i''||UK_i''||T4)\right)||(DVK_k \oplus h(T4||UK_i''))$

$h\left(h\left(UID_i''||UK_i||T2\right) \oplus \emptyset\right)||h(VID_k''||T3||VKK_k')) \stackrel{?}{=} h\left(h(UID_i''||UK_i||T4)\right)||\left(h(VID_k''||T3||VKK_k') \oplus \emptyset\right)$

$h\left(h(h(UID_i''||UK_i||T2))||h(VID_k''||T3||VKK_k')\right) \neq h\left(h(UID_i''||UK_i||T4)||h(VID_k''||T3||VKK_k')\right)$

4.4.1 Corrections au protocole de Q. Jiang et al.

Après une analyse approfondie et plusieurs tests, nous avons trouvé que les valeurs correctes devraient être lues comme suit :

$$SK_{V-U} = h((DUK_i \oplus h(j_x' \parallel T_2)) \parallel h(VID_k' \parallel T_3 \parallel VK_K'))$$

$$SK_{V-S} = h((DSK_s \oplus h(J_y' \parallel T_2)) \parallel h(VID_k' \parallel VK_k' \parallel T_3))$$

$$SK_{U-V} = h(h(UID_i' \parallel UK_i' \parallel T_2) \parallel (DVK_k \oplus H(T_4 \parallel UK_i')))$$

1) Démonstration qu'après correction, $SK_{V-S} = SK_{S-V}$:

$h\left(\mathrm{DSKs} \oplus h\left(J_y' \parallel T2\right)\right) \parallel h\left(\mathrm{VIDk}' \parallel \mathrm{VKK}' \parallel T3\right) \stackrel{?}{=} h\left(h\left(\mathrm{IDs} \parallel s \parallel T2\right) \parallel h\left(\mathrm{VIDk}' \parallel \mathrm{VKK}' \parallel T3\right)\right)$

$h\left(h\left(\mathrm{IDs} \parallel s \parallel T2\right) \oplus \oslash\right) \parallel h\left(\mathrm{VIDk}' \parallel \mathrm{VKK}' \parallel T3\right) \stackrel{?}{=} h\left(h\left(\mathrm{IDs} \parallel s \parallel T2\right) \parallel h\left(\mathrm{VIDk}' \parallel \mathrm{VKK}' \parallel T3\right)\right)$

$h(h(\text{IDs} \parallel s \parallel T2)) \parallel h(\text{VIDk}' \parallel \text{VKK}' \parallel T3) = h(h(\text{IDs} \parallel s \parallel T2)) \parallel h(\text{VIDk}' \parallel \text{VKK}' \parallel T3)$

2) Démonstration qu'après correction, $SK_{V-U} = SK_{U-V}$:

$h(\text{DUKi} \oplus h(J'_x \parallel T2)) \parallel h(\text{VIDk}' \parallel T3 \parallel \text{VKK}') \stackrel{?}{=} h(h(\text{UIDi}' \parallel \text{UKi}' \parallel T2) \parallel (\text{DVKk} \oplus h(T4 \parallel \text{UKi}')))$

$h(h(\text{UIDi}' \parallel \text{UKi}' \parallel T2) \oplus \oslash) \parallel h(\text{VIDK}' \parallel T3 \parallel \text{VKK}') \stackrel{?}{=} h(h(\text{UIDi}' \parallel \text{UKi}' \parallel T2) \parallel h(\text{VIDK}' \parallel T3 \parallel \text{VKK}'))$

$h(h(\text{UIDi}' \parallel \text{UKi}' \parallel T2)) \parallel h(\text{VIDK}' \parallel T3 \parallel \text{VKK}') = h(h(\text{UIDi}' \parallel \text{UKi}' \parallel T2)) \parallel h(\text{VIDK}' \parallel T3 \parallel \text{VKK}')$

4.5 Cryptanalyse du protocole de Jiang et al.

Dans cette section, nous détaillons la cryptanalyse du protocole d'authentification du Jiang et al.[25]. La cryptanalyse a révélé que le protocole analysé contient une véritable faiblesse de sécurité, telle que la vulnérabilité liée à l'attaquant qui extrait la clé partagée entre le capteur du véhicule et le centre de données ainsi qu'entre le capteur du véhicule et l'utilisateur. La cryptanalyse détaillée du protocole de Jiang et al. est présentée dans la section suivante.

4.5.1 Modèle d'attaque

Les étapes suivantes résument certaines capacités d'un adversaire $A*$:

- Tous les messages du canal non sécurisé peuvent être capturés par $A*$ pour collecter des informations (écoute clandestine).
- $A*$ peut être à l'intérieur ou à l'extérieur du système. Si $A*$ est à l'intérieur, les valeurs enregistrées dans la mémoire des parties honnêtes peuvent être extraites par celui-ci [59].

Nous avons supposé que $A*$ connaît uniquement l'identité du véhicule (VID_k), puis nous avons détaillé l'analyse de sécurité du protocole de Jiang et al., où nous montrons que le protocole est vulnérable à la déduction de la clé de session SK_{V-S} et SK_{V-U} via une attaque par écoute clandestine [60].

4.5.2 L'attaque de déduction d'une clé de session entre le capteur du véhicule et le centre de données.

Après l'authentification réussie, VS_K et DC peuvent partager une clé de session privée, afin d'assurer la confidentialité et l'intégrité des données transmises par le chiffrement et le déchiffrement des communications ultérieures [61].

$$SK_{V-S} = SK_{S-V} = h(h(ID_s \parallel s \parallel T_2) \parallel h(VID'_k \parallel VK'_k \parallel T_3))$$

Pour déduire une clé de session, un adversaire doit suivre ces étapes :

1. Étape SK_{V-S} 1 : $A*$ intercepte les messages transmis sur le canal public et extrait facilement les valeurs : $DVID_K$ dans le message $M1$, DSK_s dans le message $M2$ et AVK_K dans le message $M4$.

2. Étape SK_{V-S} 2 : $A*$ peut facilement dériver l'original B_y et VK_K en utilisant les opérations de XOR. Formellement, cela peut être présenté comme suit :
   $B_y* = VID_k \oplus DVID_k = VID_k \oplus (VID_k \oplus B_y) = (VID_k \oplus VID_k) \oplus B_y = \varnothing \oplus B_y$.
   $VK_K* = B_y* \oplus AVK_K = B_y* \oplus (B'_y \oplus VK''_K) = (B_y* \oplus B'_y) \oplus VK''_K = \varnothing \oplus VK''_K$.

3. Étape SK_{V-S} 3 : Ensuite, $A*$ calcule J, en utilisant la multiplication de points de l'ECC et la clé publique de DC (S)
   $J* = VK_k*.S = VK_k*.(s.P) = s.(VK_k*.P) = s.VQ_k = (J_x*, J_y*)$.

4. Étape SK_{V-S} 4 : $A*$ peut intercepter le message M_2 et extraire T_2 transmis sur le canal public. Ensuite, il peut facilement calculer $h(J_y* \parallel T_2)*$. Après avoir calculé les valeurs ci-dessus, $A*$ peut facilement dériver $h(ID_s \parallel s \parallel T_2)$
   $h(ID_s \parallel s \parallel T_2)* = DSK_s \oplus h(J_y* \parallel T_2)* = (h(ID_s \parallel s \parallel T_2) \oplus h(J_y \parallel T_2)) \oplus h(J_y* \parallel T_2)* = h(ID_s \parallel s \parallel T_2) \oplus (h(J_y \parallel T_2) \oplus h(J_y* \parallel T_2)*) = h(ID_s \parallel s \parallel T_2) \oplus \varnothing$.

5. Étape SK_{V-S} 5 : $A*$ peut intercepter le message M_3 et extraire T_3 transmis sur le canal public. Ensuite, il peut facilement calculer h$(VID_K \parallel VK_K* \parallel T_3)*$.

6. Étape SK_{V-S} 6 : Après les calculs ci-dessus, $A*$ peut facilement calculer la clé de session entre VS_K et DC.
   $SK_{V-S}* = SK_{S-V}* = h(\ h(ID_s \parallel s \parallel T_2)* \parallel h(VID_k \parallel VK_k* \parallel T_3)*\ )$

Cette attaque est finalement illustrée dans figure 4.2

4.5.3 L'attaque de déduction d'une clé de session entre le capteur du véhicule et l'utilisateur

Après l'authentification réussie, les fournisseurs de VS_K et U_i peuvent partager une clé de session privée, afin d'assurer la confidentialité et l'intégrité des données transmises par le chiffrement et le déchiffrement des communications ultérieures [**?**].

$$SK_{V-U} = SK_{U-V} = h(h(UID'_i \parallel UK_i \parallel T_2) \parallel h(VID'_k \parallel VK'_k \parallel T_3))$$

Pour déduire une clé de session, un adversaire doit suivre ces étapes.

1. Étape SK_{V-U} 1 : $A*$ intercepte les messages transmis sur le canal public et extrait facilement les valeurs : $DVID_K$ dans le message $M1$, DUK_i dans le message $M2$ et AVK_K dans le message $M4$.

2. Étape SK_{V-U} 2 : $A*$ peut facilement dériver l'original B_y et VK_K en utilisant les opérations de XOR. Formellement, cela peut être présenté comme suit :

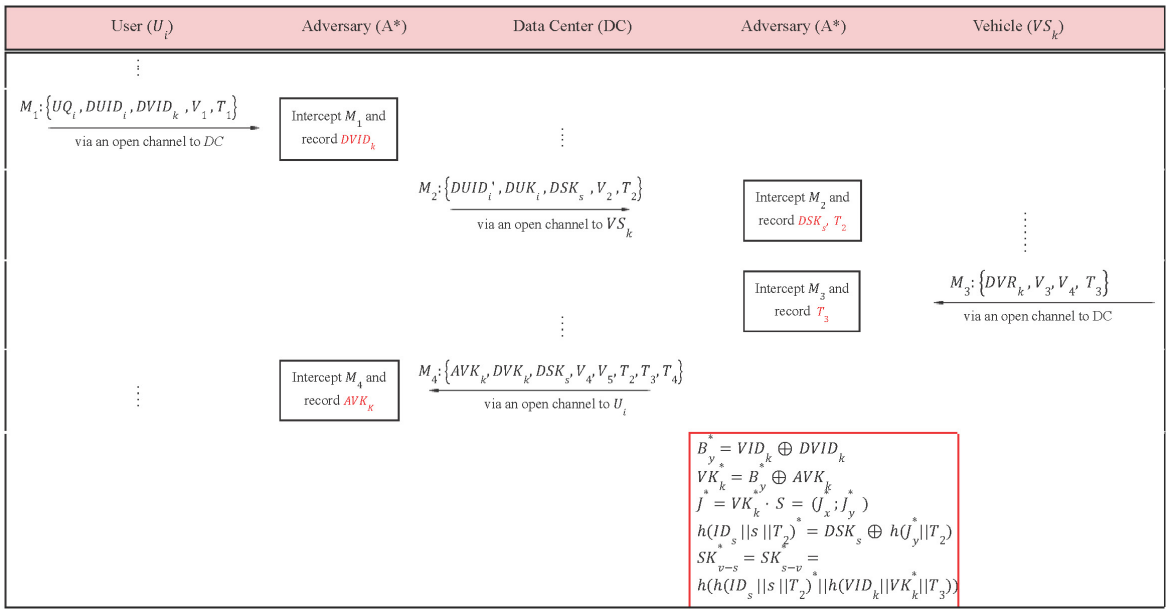


FIGURE 4.2 : Illustration de la dérivation d'une clé de session entre le capteur du véhicule et le centre de données dans le schéma de Jiang et al.

$B_y{*} = VID_k \oplus DVID_k = VID_k \oplus (VID_k \oplus B_y) = (VID_k \oplus VID_k) \oplus B_y = \varnothing \oplus B_y.$
$VK_K{*} = B_y{*} \oplus AVK_K = B_y{*} \oplus (B'_y \oplus VK''_K) = (B_y{*} \oplus B'_y) \oplus VK''_K = \varnothing \oplus VK''_K.$

3. Étape SK_{V-U} 3 : Ensuite, A^* calcule J, en utilisant la multiplication de points de l'ECC et la clé publique de DC (S)
$J^* = VK_k{*} \,.\, S = VK_k{*} \,.\, (s \,.\, P) = s \,.\, (VK_k{*} \,.P) = s.VQ_k = (J_x{*}, J_y{*}).$

4. Étape SK_{V-U} 4 : A^* peut intercepter le message M_2 et extraire T_2 transmis sur le canal public. Ensuite, il peut facilement calculer $h(J_x{*} \parallel T_2)^*$. Après avoir calculé les valeurs ci-dessus, A^* peut facilement dériver $h(UID'_i \parallel UK_i \parallel T_2)$.
$h(UID'_i \parallel UK_i \parallel T_2)^* = DUK_i \oplus h(J_x{*} \parallel T_2)^* = (h(UID'_i \parallel UK_i \parallel T_2) \oplus h(J_x \parallel T_2)\,) \oplus h(J_x{*} \parallel T_2)^* = h(UID'_i \parallel UK_i \parallel T_2) \oplus (h(J_x \parallel T_2) \oplus h(J_x{*} \parallel T_2)^*) = h(UID'_i \parallel UK_i \parallel T_2) \oplus \varnothing.$

5. Étape SK_{V-U} 5 : A^* peut intercepter le message M_3 et extraire T_3 transmis sur le canal public. Ensuite, il peut facilement calculer h$(VID_K \parallel T_3 \parallel VK_K{*})^*$.

6. Étape SK_{V-U} 6 : Après les calculs ci-dessus, A^* peut facilement calculer la clé de session entre VS_K et U_i.
$SK_{V-U}{*} = SK_{U-V}{*} = h(\, h(UID'_i \parallel UK_i \parallel T_2)* \parallel h(VID_k \parallel T_3 \parallel VK_k*)* \,)$

Cette attaque est également illustrée dans la figure 4.3.

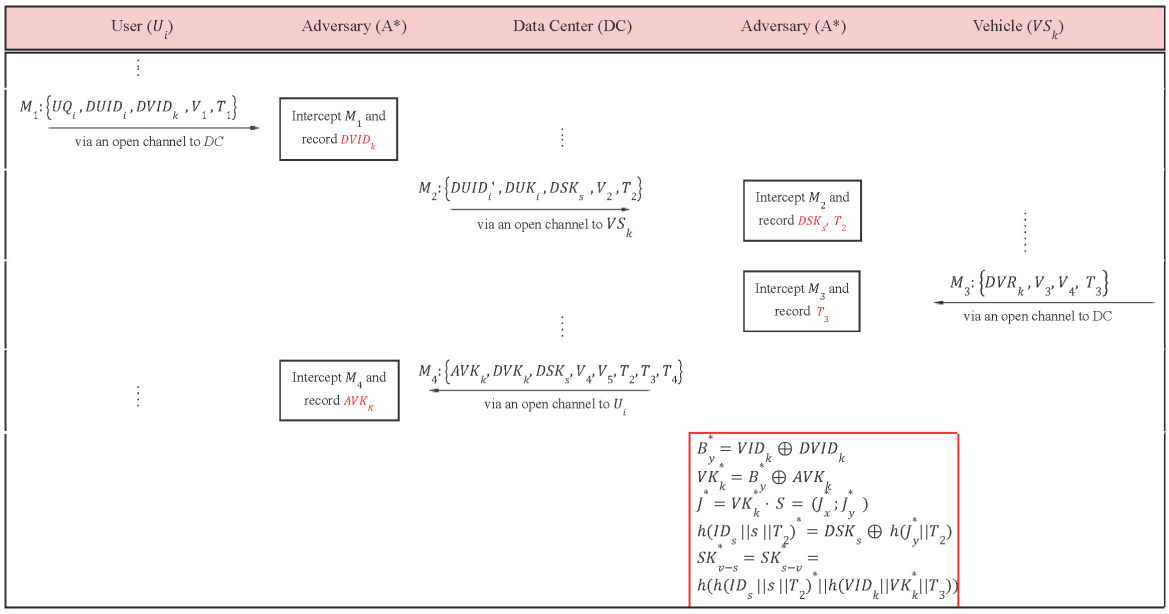


FIGURE 4.3 : Illustration de la dérivation d'une clé de session entre le capteur du véhicule et l'utilisateur dans le schéma de Jiang et al.

4.6 Amélioration du Schéma

La cause fondamentale des faiblesses du schéma de Jiang et al. réside dans le fait que l'adversaire a accès à la communication publique et peut intercepter les messages transmis. L'objectif principal de l'adversaire en interceptant ces messages est d'obtenir les clés de session. Dans cette section, nous proposons une amélioration du schéma de Jiang et al. pour surmonter les problèmes de sécurité identifiés dans le schéma analysé de Jiang et al.

Dans notre schéma amélioré de Jiang et al., nous modifions la phase d'authentification pour renforcer la sécurité. Les modifications suivantes dans la phase d'authentification sont nécessaires pour se protéger contre une attaque par dérivation des clés de session :

1. Au cours de l'Étape 2, après une vérification réussie, l'utilisateur (U_i) choisit un secret aléatoire UN_i pour calculer UQ_i=$UN_i \cdot P$ et B=$UN_i \cdot S$=(B_x,B_y), la clé temporaire partagée avec le Centre de Données. U_i peut en outre calculer une autre clé temporaire
   C= $(UN_i + h(UID_i \,||ID_s\,||T_1)) \cdot S \bmod l$ =(C_x,C_y) partagée avec le Centre de Données, où S est la clé publique du Centre de Données. Ensuite, l'appareil commence à calculer les identités pseudo de l'utilisateur et du véhicule $DUID_i$=$UID_i' \oplus B_x$,

   $DVID_k$=$VID_k \oplus C_x$, et V_1=h($UID_i' \,||\, ID_s \,||\, VID_k \,||\, UK_i' \,||\, T_1 \,||B_x \,||C_x$).

   Enfin, l'appareil transmet le message M_1=(UQ_i, $DUID_i$, $DVID_k$, V_1, T_1) au Centre de Données via un canal public.

2. Après réception du message M_1 de U_i, le Centre de Données vérifie d'abord l'actualité de M_1. Si c'est le cas, le Centre de Données calcule la clé temporaire B'=$s \cdot UQ_i$ =(B'_x, B'_y) et l'identification de l'utilisateur UID'_i=$DUID_i \oplus B_x$. Sinon, le Centre de Données calcule une autre clé temporaire partagée avec l'utilisateur. C'= $(h(UID'_i \,||ID_s\, ||T_1) \cdot P + UQ_i) \cdot s \bmod l$= (C'_x, C'_y), où s est la clé privée du Centre de Données. Ensuite, il calcule l'identification du véhicule VID'_k=$DVID_k \oplus C'_x$, et obtient UK_i en recherchant dans la base de données selon UID'_i. S'il n'est pas reconnu, cela signifie que U_i est un utilisateur non autorisé, et le Centre de Données mettra fin immédiatement à la session. Sinon, le Centre de Données vérifie la validité de $V_1 \stackrel{?}{=}$ h(UID'_i || ID_s || VID'_k || UK_i || T_1 ||B'_x ||C'_x). Si c'est le cas, le message M_1 de l'utilisateur est accepté par le Centre de Données. Le reste du protocole reste inchangé.

La clé temporaire partagée (C) entre l'utilisateur (U_i) et le Centre de Données (DC) calculée dans la phase d'authentification améliorée est équivalente, et cela est démontré comme suit :

$$
\begin{aligned}
C \text{ de } U_i &= (UN_i + h(UID_i \,||ID_s\, ||T_1)) \cdot S \bmod l \\
&= (UN_i + h(UID_i \,||ID_s\, ||T_1)) \cdot (s \cdot P) \bmod l \\
&= (UN_i \cdot P + h(UID_i \,||ID_s\, ||T_1) \cdot P) \cdot s \bmod l \\
&= (UN_i \cdot P + h(UID_i \,||ID_s\, ||T_1) \cdot P) \cdot s \bmod l \\
&= (UQ_i + h(UID_i \,||ID_s\, ||T_1) \cdot P) \cdot s \bmod l
\end{aligned}
$$

$$
\begin{aligned}
C \text{ de } DC &= (h(UID'_i \,||ID_s\, ||T_1) \cdot P + UQ_i) \cdot s \bmod l \\
&= (UQ_i + h(UID'_i \,||ID_s\, ||T_1) \cdot P) \cdot s \bmod l
\end{aligned}
$$

L'adversaire A* est incapable de relier les messages (M_1, M_2, M_3, et M_4) puisque les composants des messages sont tous uniques et dynamiques grâce à l'utilisation de marqueurs temporels et de secrets aléatoires, ce qui rend très difficile pour un adversaire A* d'extraire les clés de session. Ainsi, dans notre amélioration, il est évident que les clés de session sont protégées.

U_i	DC	VS_k
input UID_i, PW_i, imprint BIO_i	.....	
$UC_i = W_i \bigoplus (h(UID_i \| PW_i))$	check timeliness T_1	
$UR_i = UF_i(U_C)$	$B = s \cdot UQ_i = (B_x, B_y)$	
$w = BIO_i \bigoplus UR_i$	$UID_i = DUID_i \bigoplus B_x$	
$K_i^* = Decoder(UP_i \bigoplus w))$	$C = (h(UID_i \| ID_s \| T_1) \cdot P + UQ_i) \cdot s$	
$UK_i = h(K_i)$	$VID_k = DVID_k \bigoplus C_x$	
$V_i = h(h(PW_i \| UIK_i))$	obtain UID_k by UK_i	
select UN	$V_1 = h(UID_i \| ID_s \| VID_k \| UK_i \| T_1 \| B_x)$	check timeliness of T_2
$UQ_i = UN_i \cdot P$	get VID_k by VP_k, VQ_k and $J = s \cdot VQ_k$	$VR_k = VF(VID_k)$
$B = UN_i \cdot S = (B_x, B_y)$	$DUID_i = UID_i \bigoplus h(VID_k \| ID_s \| J_x \| T_2)$	$(VP_k, VK_k) = Gen(VR_k)$
$DUID_i = UID_i \bigoplus B_x$	$DUK_i = h(UID_i \| UK_i \| T_2) \bigoplus h(J_x \| T_2)$	$J = V_K \cdot S, S = (J_x, J_y)$
$C_i = (UN + h(UID_i \| ID_s \| T_1) \cdot S = (C_x, C_y)$	$DSK_s = h(ID_s \| s \| T_2) \bigoplus h(J_y \| T_2)$	$UID_i = DUID_i \bigoplus h(VID_k \| ID_s \| J_x \| T_2)$
$DVID_k = VID_k \bigoplus C_x$	$V_2 = h(VID_k \| ID_s \| UID_i \| J_x \| J_y \| T_2)$	$V_2 = h(VID_k \| ID_s \| UID_i \| J_x \| J_y \| T_2)$
$V_1 = h(UID_1 \| ID_s \| VID_k \| UK_i \| T_1 \| B_x \| C_x)$	$M_2 =< DUID_i, DUK_i, DSK_s, V_2, T_2 >$	$SK_{v-u} = h(h(DUK_i \bigoplus h(J_x \| T_2)) \| h(VID_k \| T_3 \| VK_k))$
$M_1 = (UQ_i, DUID_i, DVID_k, V_1, T_1)$ to DC	.....	$SK_{v_s} = h(h(DSK_s \bigoplus h(J_y \| T_2)) \| h(VID_k \| VK_k \| T_3))$
	check timeliness of T_3	$DVR_k = VR_k \bigoplus J_x$
	$VR_k = DVR_k \bigoplus J^x$	$V_3 = h(ID_s \| VID_k \| J_y \| SK_{v-s} \| T_3)$
	$VK_k = Rep(VP_k, VR_k)$	$V_4 = h(UID_1 \| VID_1 \| SK_{w_s} \| T_3)$
	$SK_{s-v} = h(h(ID_s \| s \| T_2) \| h(VID_k \| VK_k \| T_4)$	$M_3 =< DVR_k, V_3, V_4, T_3 >$
	$V_3 = h(ID_s \| VID_k \| SK_{s-v} \| T_3)$	
check timeliness of T_4	$SKs - u = h(h(ID_s \| T_4 \| s) \| h(UID_i \| UK_i \| T_4))$	
$SK_{u-s} = h(DSK_s \bigoplus h(UK_i \| T_4)) \| h(UID_i \| UK_i \| T_4))$	$V_5 = h(ID_s \| UID_i \| SK_{s-u} \| T_4)$	
$V_5 = h(ID_s \| UID_i \| B_y \| SK_{u-s} \| T_4)$	$AVK_k = B_y \bigoplus VK_k$	
$VK_k = AVK_k \bigoplus B_y$	$DVK_k = h(VID_k \| T_3 \| VK_k) \bigoplus h(T_4 \| UK_i)$	
$VK_k = AVK_k \bigoplus By$	$DSK_s = h(ID_s \| T_4 \| s) \bigoplus h(UK_i \| T_4)$	
$SKu - v = h(h(UIDi \| UK_i \| T4) \| (DVK_k \bigoplus h(T_4 \| UK_k)$	$M_4 =< AVK_k, DVk_k, DSk_k, V_4, V_5, T_2, T_3, T_4 >$	
$V_4 = h(UID_i \| VID_k \| VK_k \| SK_{u-v} \| T_3)$		

FIGURE 4.4 : Phase d'authentification de notre protocole améliorée

4.7 Analyse des Performances

Afin d'évaluer les coûts de calcul et de communication associés à notre protocole amélioré, nous nous sommes basés sur les résultats expérimentaux antérieurs appliqués dans [28].

Le tableau 4.2 comparatif présente les coûts de calcul de deux schémas, celui de "Jiang et al." et "Notre protocole". Le schéma de Jiang et al. se distingue par un coût total inférieur, soit 23.55 ms, comparé à 40.60 ms pour notre protocole. Cette différence est principalement due à un temps d'exécution utilisateur plus court (13.38 ms contre 25.306 ms) et un temps d'exécution TA/CS significativement plus rapide (2.809 ms contre 8.02 ms). En revanche, les deux schémas affichent un temps identique pour la capture de véhicule, soit 7.336 ms. Bien que notre protocole présente un temps de calcul global plus élevé, il a permis de résoudre la vulnérabilité existante tout en maintenant un temps de traitement faible pour la capture de véhicule, ce qui garantit que le protocole reste applicable pour l'Internet des véhicules.

Scheme	User	TA/CS	VS	Total Cost
Jiang et al. [25]	13.38 ms	2.809 ms	7.336 ms	23.55
Notre protocole	25.306 ms	8.02 ms	7.336 ms	40.60

TABLE 4.2 : Comparaison des coûts de calcul entre le protocole de Jiang et al. et notre amélioration

En ce qui concerne le coût de communication, nous avons conservé le même nombre de bits transmis. Le protocole se compose de quatre messages : le premier, envoyé par l'utilisateur au centre de données, contient 1152 bits ; le deuxième, un message de 832 bits, est échangé entre le centre de données et le capteur de véhicule ; le troisième message, une réponse du capteur de véhicule au centre de données, contient 672 bits ; et le dernier message, envoyé par le centre de données à l'utilisateur, contient 992 bits. Ainsi, le nombre total de bits échangés s'élève à 3648 bits.

4.8 Conclusion

Dans ce chapitre, nous avons détaillé la cryptanalyse du protocole d'authentification proposé par Q. Jiang et al. [25], ou nous avons démontré que les clés de session pouvaient être déduites par un adversaire. Par conséquent, il est clair que le schéma de Q. Jiang et al. souffre d'une faiblesse majeure en termes de confidentialité des messages, permettant à un attaquant potentiel de déduire la clé de session partagée entre le véhicule et le centre de données, ainsi qu'entre le véhicule et l'utilisateur.

Pour remédier à ces failles de sécurité, nous avons proposé une amélioration du protocole. Bien que notre nouveau protocole entraîne un temps de calcul global plus élevé, il corrige la vulnérabilité existante tout en maintenant un temps de traitement relativement faible pour véhicule, ce qui garantit son applicabilité à l'Internet des véhicules. Ainsi, ce protocole est particulièrement adapté aux systèmes

nécessitant des réponses rapides, comme les véhicules connectés. Cependant, il pourrait ne pas être idéal pour les environnements Big Data, car l'échange de 4 messages, avec des tailles de messages pouvant atteindre 3648 bits, peut augmenter la charge sur le réseau, ce qui pourrait entraîner des ralentissements dans systèmes. En particulier, lorsque le centre de données reçoit une demande d'authentification de l'utilisateur, il doit retransmettre la demande et attendre la réponse du véhicule pour la renvoyer à l'utilisateur, ce qui peut poser des problèmes de performance dans les environnements à grande échelle.

Dans le chapitre suivant, nous aborderons cette problématique et proposerons des solutions pour l'améliorer.

Chapitre 5

Amélioration du schéma d'authentification des utilisateurs pour la collecte de Big Data dans un système de transport intelligent basé sur l'IoT.

Dans ce chapitre, nous présentons les recherches récentes sur l'Internet des véhicules, en particulier les systèmes de transport intelligents (ITS), où une grande quantité de données doit être collectée et envoyée. Toutefois, la communication au sein des ITS est confrontée à de nombreux problèmes, ce qui conduit à diverses attaques. Récemment, Srinivas et al. ont proposé un protocole de sécurité adapté à la collecte de Big Data dans les ITS basés sur l'IoT, appelé UAP-BCIoT [48]. Cependant, après notre analyse de sécurité, nous avons découvert une faille. Dans ce chapitre, nous montrons comment cette faille affecte l'accès des utilisateurs aux données fournies par les dispositifs IoT déployés sur les véhicules. De plus, nous proposons des améliorations pour résoudre le problème d'authentification du schéma.

5.1 Introduction

D'ici 2050, plus de 70 % de la population mondiale devrait résider dans des zones urbaines, une urbanisation rapide qui engendrera de nombreux défis organisationnels et techniques. Pour gérer efficacement cette croissance, plusieurs pays ont adopté le concept de villes intelligentes. Ces villes s'appuient sur des technologies avancées, notamment les systèmes de transport intelligents (ITS). L'ITS utilise des innovations telles que l'Internet des objets (IoT), les systèmes cyber-physiques et l'analyse des Big Data pour optimiser les services urbains, en particulier le transport. La collecte et

l'analyse de Big Data dans les ITS jouent un rôle essentiel dans l'amélioration de la qualité de vie des citoyens, en facilitant une gestion plus efficace et sécurisée du trafic et des infrastructures de transport. Le protocole UAP-BCIoT, récemment proposé par Srinivas et al. [48], a été spécialement conçu pour les systèmes de transport intelligents basés sur l'IoT. Il présente l'avantage de n'utiliser que trois messages, contrairement à d'autres protocoles qui en nécessitent quatre, comme celui décrit dans le quatrième chapitre. Cette réduction du nombre de messages diminue le volume de données transmises, allégeant ainsi la charge sur le nœud Cloud-Gateway.

Cependant, la protection des dispositifs critiques utilisés dans ces systèmes est une priorité. En effet, toute intrusion pourrait entraîner des dysfonctionnements graves, voire dangereux, dans les véhicules connectés. Par conséquent, il est impératif de concevoir des mécanismes de communication sécurisés pour minimiser les risques de cyberattaques. Lors de la conception d'un système ITS, il est essentiel de prendre en compte la résilience face à plusieurs types d'attaques, telles que les intrusions internes, l'usurpation d'identité, les attaques par rejeu, ou les attaques de type homme du milieu. L'authentification est généralement perçue comme une solution clé pour sécuriser les communications dans ces systèmes tout en protégeant l'identité des véhicules. Un schéma d'authentification adapté aux ITS doit également garantir trois caractéristiques essentielles : l'intraçabilité, l'authentification mutuelle et l'anonymat.

Le protocole UAP-BCIoT propose une approche innovante pour répondre à ces enjeux. Il établit un cadre de protection pour les dispositifs IoT intelligents au sein des systèmes de transport. Le modèle de réseau pour schéma UAP-BCIoT est présenté dans la figure 5.1. Dans ce schéma, le centre de confiance pour l'enregistrement des Big Data (BRC) est chargé d'enregistrer les nœuds Cloud-Gateway semi-fiables (CG), les utilisateurs, ainsi que les dispositifs IoT. Le nœud Cloud-Gateway joue un rôle crucial en assurant l'authentification mutuelle entre l'utilisateur et un nœud IoT. Une fois l'authentification réussie, une clé de session secrète est échangée, permettant à l'utilisateur légitime d'accéder aux informations en temps réel, tout en garantissant la sécurité et l'intégrité des données. Bien que le protocole UAP-BCIoT soit conçu pour protéger contre plusieurs attaques connues, il présente une faille affectant l'authentification entre les utilisateurs légitimes et les dispositifs IoT déployés dans les véhicules. Ce problème peut compromettre l'accès des utilisateurs aux données critiques fournies par ces dispositifs. Dans ce chapitre, nous démontrons l'impact de cette faille et proposons des améliorations pour renforcer le schéma d'authentification. Notre analyse comparative montre que la version améliorée du protocole UAP-BCIoT offre de meilleures performances en termes d'efficacité, de sécurité et de fiabilité par rapport à la version originale.

Le reste de ce chapitre est organisé comme suit. La section 5.2 présente et analyse le schéma UAP-BCIoT. Nous discuterons des raisons pour lesquelles cette faille est apparue et de son impact sur les affirmations de l'article dans la section 5.3. Nous proposerons une version améliorée de l'UAP-BCIoT dans la section 5.4. Une analyse détaillée des coûts de communication et de calcul pour l'UAP-BCIoT amélioré est présentée dans la section 5.5. Enfin, nous résumerons ce travail dans la section 5.6.

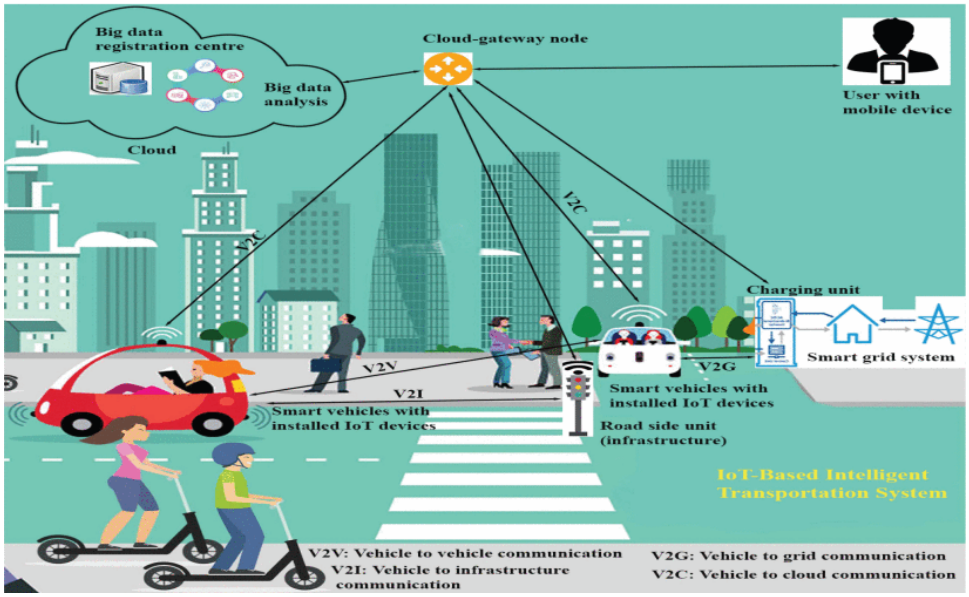


FIGURE 5.1 : Modèle de réseau du schéma UAP-BCIoT

5.2 Revues du schéma UAP-BCIoT

Dans cette section, nous passons en revue le schéma UAP-BCIoT récemment présenté par Srinivas et al. [48]. Afin de faciliter la description et l'analyse du schéma UAP-BCIoT, nous examinons le modèle de réseau qu'ils utilisent. La Figure 5.1 représente les entités typiques d'un ITS, où les utilisateurs accèdent aux dispositifs IoT via Internet public. Ces entités comprennent principalement les utilisateurs, le nœud passerelle cloud et les nœuds IoT, mais incluent également le BRC, les véhicules intelligents, les unités de recharge de véhicules et les unités en bordure de route. Dans ce modèle de réseau, divers types de communications sont présents : communication véhicule-à-véhicule, véhicule-à-infrastructure, véhicule-au-réseau et véhicule-au-cloud. UAP-BCIoT se concentre sur la communication entre des utilisateurs autorisés qui peuvent accéder directement aux données des nœuds IoT installés dans les véhicules intelligents via le nœud CG. Leur schéma comprend différentes phases. Les notations couramment utilisées tout au long de ce chapitre sont répertoriées dans le Tableau 5.1.

5.2.1 Phase d'initialisation du système

Le BRC sert d'entité autorisée dans le réseau, responsable de la récupération des différents paramètres pour chaque nœud déployé. De plus, la passerelle cloud autorisée (CG) sélectionne ses propres clés privées et publiques ainsi que d'autres paramètres.

TABLE 5.1 : Notations utilisées dans le protocole UAP-BCIOT

Notations	**Élucidations**
BRC	Centre d'enregistrement de données de confiance
CG_k	k^{me} passerelle cloud semi-fiable
IN_j	j^{me} dispositif IoT intelligent
U_i, PW_i	i^{me} utilisateur et son mot de passe
ID_{CG_k}, ID_{IN_j}	Identification de CG_k et IN_j, respectivement
X_{CG_k}	Clé secrète de CG_k
P	Générateur de point ECC
G_{pub}=$G_{pri}.P$	Clé publique calculée par CG_k
x, y	Nombres arbitraires générés par U_i et IN_j, respectivement
TS_1, TS_2, TS_3	Horodatages

5.2.2 Phase d'enregistrement des dispositifs IoT

Le BRC effectue cette phase hors ligne. Le BRC calcule le secret $IC_{j_1} \leftarrow h(SK_{CG-BRC} \parallel ID_{IN_j})$ pour chaque nœud IoT (ID_{IN_j}). Le BRC enregistre (ID_{IN_j}, IC_{j_1}) dans la mémoire de IN_j avant son déploiement dans l'environnement ITS.

5.2.3 Phase d'enregistrement de l'utilisateur

Pour s'enregistrer et devenir un utilisateur légitime, U_i exécute les étapes suivantes :

1. Initialement, l'utilisateur U_i est libre de choisir son identité (ID_i) et son mot de passe (PW_i) et détermine ses informations biométriques (BIO_i). Ensuite, U_i calcule Gen(BIO_i)$\leftarrow$(σ_i, τ_i),

   $MID_i \leftarrow$ h($ID_i \| b_i$) et $MPW_i \leftarrow$ h($PW_i \| \sigma_i$) où b_i est un nombre aléatoire généré par MD_i. Après cela, U_i envoie $\{MID_i, MPW_i\}$ avec une requête d'enregistrement à CG_k via un canal sécurisé.

2. À la réception du message de requête, CG_k calcule $G_1 \leftarrow (g_{pri} \cdot h(MID_i)) \cdot P$, $G_2 \leftarrow G_1 \oplus h(MPW_i \parallel MID_i)$ et $G_3 \leftarrow G_1 \oplus h(X_{CG_k})$. Ensuite, CG_k envoie $\{G_1, G_2, G_3, h(\cdot), \mathrm{P}\}$ à U_i via un canal sécurisé.

3. À la réception du message de réponse, U_i calcule $L_i \leftarrow b_i \oplus h(ID_i \parallel \sigma_i \parallel PW_i)$,

   $G_2^* \leftarrow G_2 \oplus h(b_i \parallel \sigma_i \parallel PW_i)$, $G_3^* \leftarrow G_3 \oplus h(\sigma_i \parallel b_i \parallel PW_i)$, $G_4 \leftarrow h(G_1 \parallel PW_i \parallel b_i \parallel \sigma_i)$.

4. Enfin, U_i conserve $\{L_i, G_2^*, G_3^*, G_4, h(\cdot), Gen(\cdot), Rep(\cdot), \tau_i, \mathrm{P}\}$ dans son dispositif mobile.

5.2.4 Phase de connexion et d'authentification

Supposons qu'à un moment donné, un utilisateur souhaite surveiller les données d'un nœud IoT spécifique intégré dans un véhicule intelligent. Pour garantir la confidentialité de l'utilisateur et maintenir l'authenticité des données acquises, il est essentiel que les utilisateurs et les nœuds IoT confirment mutuellement leurs identités et établissent une clé secrète partagée avant d'échanger des données. La Figure 5.2 illustre le processus complet de la phase de connexion et d'authentification du UAP-BCIoT.

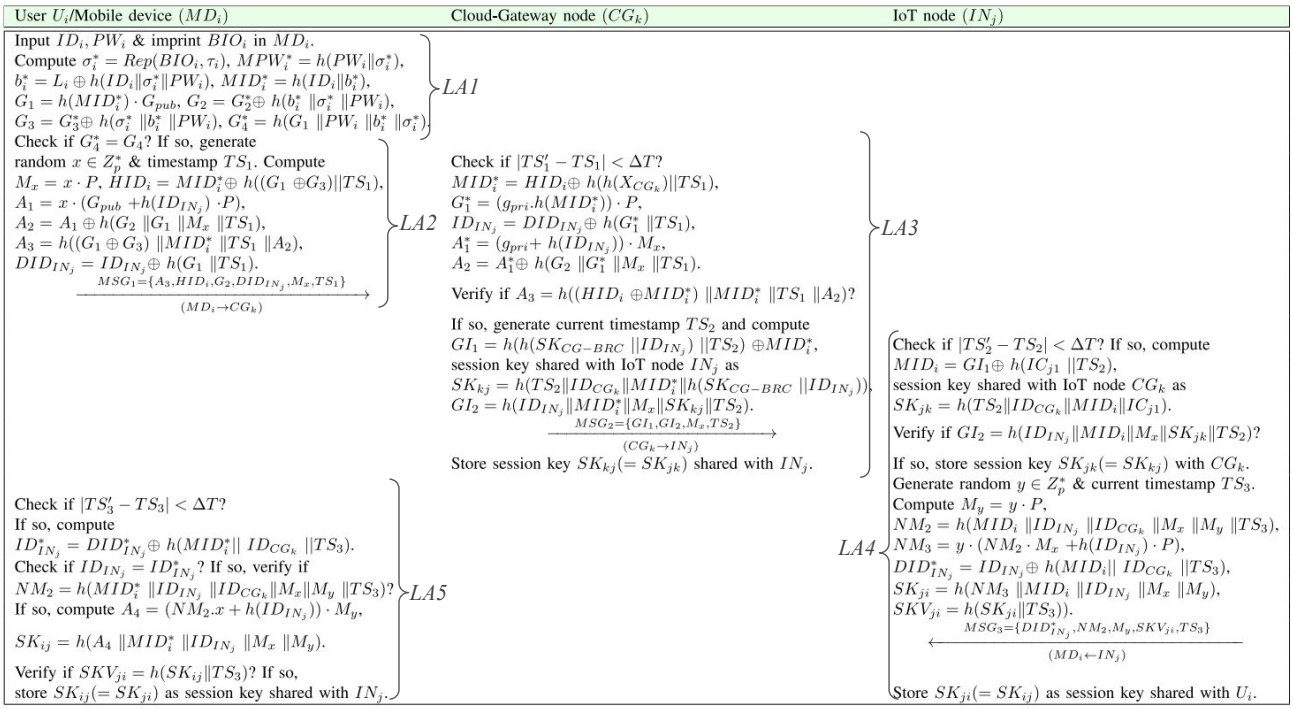


FIGURE 5.2 : Phase d'authentification du schéma UAP-BCIOT

5.3 Énoncé du problème

La phase d'authentification du protocole UAP-BCIoT est divisée en cinq étapes. Le processus d'authentification est réalisé par le dispositif passerelle-cloud (CG_k) pour valider les utilisateurs participants et déterminer si un nœud IoT (IN_j) fonctionne correctement ou non dans le réseau. Le problème apparaît aux étapes LA2 et LA3.

L'étape *LA*2 de la phase d'authentification du manuscrit considéré est la suivante. L'appareil mobile (MD_i) vérifie G_4. Si correct, MD_i confirme que les informations d'identification saisies par l'utilisateur (U_i) sont valides. U_i sélectionne l'identité du dispositif IoT auquel il souhaite accéder, puis génère un

nonce aléatoire (x) et un horodatage (TS_1) pour calculer les valeurs suivantes :

$$M_x \leftarrow x \cdot P$$

$$HID_i \leftarrow MID_i^* \oplus h((G_1 \oplus G_3) \parallel TS_1)$$

$$A_1 \leftarrow x \cdot (G_{pub} + h(ID_{IN_j}) \cdot P)$$

$$A_2 \leftarrow A_1 \oplus h(G_2 \parallel G_1 \parallel M_x \parallel TS_1)$$

$$A_3 \leftarrow h((G_1 \oplus G_3) \parallel MID_i^* \parallel TS_1 \parallel A_2)$$

$$DID_{IN_j} \leftarrow ID_{IN_j} \oplus h(G_1 \parallel TS_1)$$

L'appareil mobile envoie le message de requête (MSG_1) au Cloud-Passerelle semi-fiable (CG_k) via un canal public.

$$MSG_1 \leftarrow \{A_3, HID_i, G_2, DID_{IN_j}, M_x, TS_1\}.$$

À l'étape $LA3$, CG_k vérifie le délai de transmission maximum après réception du message de requête MSG_1. Si la vérification est réussie, CG_k calcule :

$$MID_i^* \leftarrow HID_i \oplus h(h(X_{CG_k}) \parallel TS_1)$$

$$G_1^* \leftarrow (g_{pri} \cdot h(MID_i^*)) \cdot P$$

$$ID_{IN_j} \leftarrow DID_{IN_j} \oplus h(G_1^* \parallel TS_1)$$

$$A_1^* \leftarrow (g_{pri} + h(ID_{IN_j})) \cdot M_x$$

$$A_2 \leftarrow A_1^* \oplus h(G_2 \parallel G_1^* \parallel M_x \parallel TS_1)$$

Après ce calcul, CG_k vérifie :

$A_3 \stackrel{?}{=} h((HID_i \oplus MID_i^*) \parallel MID_i^* \parallel TS_1 \parallel A_2)$

Cette vérification échoue systématiquement, ce qui entraîne l'annulation de la session, même si le message reçu de l'utilisateur est légitime. La raison de cet échec réside dans une faille de la description de la valeur de vérification (A_3) calculée par U_i et vérifiée par CG_k, qui ne sont pas équivalentes, ce qui est justifié par ce qui suit :

A_3 de U_i= $h((G_1 \oplus \underline{G_3}) \parallel MID_i^* \parallel TS_1 \parallel A_2)$

$=h((G_1 \oplus \underline{(G_1 \oplus h(X_{CGK}))}) \parallel MID_i^* \parallel TS_1 \parallel A_2)$

$=h((G_1 \oplus G_1 \oplus h(X_{CGK})) \parallel MID_i^* \parallel TS_1 \parallel A_2)$

$=h((0 \oplus h(X_{CGK})) \parallel MID_i^* \parallel TS_1 \parallel A_2)$

$=h(h(X_{CGK}) \parallel MID_i^* \parallel TS_1 \parallel A_2)$

A_3 de CG_k= $h((\underline{HID_i} \oplus MID_i^*) \parallel MID_i^* \parallel TS_1 \parallel A_2)$

$= h((\ \underline{(MID_i^* \oplus h((G_1 \oplus G_3) \parallel TS_1))} \oplus MID_i^*) \parallel MID_i^* \parallel TS_1 \parallel A_2)$

$= h(((h((G_1 \oplus G_1 \oplus h(X_{CGK}) \oplus 0) \parallel TS_1))) \parallel MID_i^* \parallel TS_1 \parallel A_2)$

$= h(((h((0 \oplus h(X_{CGK})) \parallel TS_1))) \parallel MID_i^* \parallel TS_1 \parallel A_2)$

$= h(h(h(X_{CGK}) \parallel TS_1) \parallel MID_i^* \parallel TS_1 \parallel A_2)$

$\neq A_3$ de U_i

Le problème mentionné ci-dessus a rendu A_3 calculé par U_i et vérifié par CG_k non équivalents. Sans cette équivalence, les principales affirmations de l'article, telles que l'authentification d'un utilisateur légitime par le nœud Cloud-Gateway, ne sont pas valides.

5.4 Solution proposée

Dans cette section, nous proposons une version améliorée du schéma UAP-BCIoT pour surmonter ces défauts. Deux solutions possibles sont proposées dans les sous-sections suivantes.

5.4.1 Solution côté Cloud-Gateway

Le dispositif cloud-gateway (CG_k) effectue le processus d'authentification pour valider les utilisateurs participants. CG_k vérifie la validité du participant U_i en vérifiant A_3 après réception de la requête de connexion MSG_1.
Nous avons montré que A_3 calculé par U_i et vérifié par CG_k ne sont pas équivalents, ce qui a conduit à la suspension d'un utilisateur légitime. Néanmoins, notre analyse indique qu'il existe un moyen de garantir l'équivalence entre A_3 reçu d'un utilisateur et A_3 vérifié par CG_k. Pour cette raison, nous ajustons l'Étape $LA2$ de l'Éq. (HID_i) et l'Étape $LA3$ de l'Éq. (MID_i^*) comme suit :

$$HID_i \leftarrow MID_i^* \oplus (G_1 \oplus G_3)$$

$$MID_i^* \leftarrow HID_i \oplus h(X_{CG_k})$$

De cette manière, la valeur de vérification A_3 calculée par U_i et celle calculée par CG_k sont équivalentes, ce qui est justifié par ce qui suit :
A_3 de $CG_k = h((\ \underline{HID_i} \oplus MID_i^*) \parallel MID_i^* \parallel TS_1 \parallel A_2)$
$= h((\ \underline{(MID_i^* \oplus (G_1 \oplus G_3))} \oplus MID_i^*) \parallel MID_i^* \parallel TS_1 \parallel A_2)$
$= h((MID_i^* \oplus (G_1 \oplus \underline{G_3}\ ) \oplus MID_i^*) \parallel MID_i^* \parallel TS_1 \parallel A_2)$
$= h((G_1 \oplus \underline{G_1 \oplus h(X_{CGK})} \oplus 0) \parallel MID_i^* \parallel TS_1 \parallel A_2)$
$= h((0 \oplus h(X_{CGK})) \parallel MID_i^* \parallel TS_1 \parallel A_2)$
$= h(h(X_{CGK}) \parallel MID_i^* \parallel TS_1 \parallel A_2)$
$= A_3$ de U_i (UAP-BCIoT)

Avec cette première solution (côté CG_k), le problème d'authentification des utilisateurs légitimes vers les dispositifs IoT souhaités est résolu.

5.4.2 Solution côté utilisateur

Les utilisateurs externes autorisés peuvent accéder directement aux données des capteurs IoT déployés dans les véhicules via le message de requête de connexion (MSG_1) au dispositif cloud-gateway. Dans la section précédente, nous avons indiqué le problème d'égalité. Dans cette sous-section, nous fournissons le correct A_3 pour le message de requête de connexion. Plus de détails ci-dessous :

$$A_3 \leftarrow h(h((G_1 \oplus G_3) \parallel TS_1) \parallel MID_i^* \parallel TS_1 \parallel A_2)$$

De cette manière, la vérification A_3 calculée par U_i et celle calculée par CG_k sont équivalentes, ce qui est justifié par ce qui suit :

A_3 de $U_i = h(h(G_1 \oplus \underline{G_3} \parallel TS_1) \parallel MID_i^* \parallel TS_1 \parallel A_2)$
$= h(h(G_1 \oplus \underline{(G_1 \oplus h(X_{CGK}))} \parallel TS_1) \parallel MID_i^* \parallel TS_1 \parallel A_2)$
$= h(h(G_1 \oplus G_1 \oplus h(X_{CGK}) \parallel TS_1) \parallel MID_i^* \parallel TS_1 \parallel A_2)$
$= h(h(0 \oplus h(X_{CGK}) \parallel TS_1) \parallel MID_i^* \parallel TS_1 \parallel A_2)$
$= h(h(h(X_{CGK}) \parallel TS_1) \parallel MID_i^* \parallel TS_1 \parallel A_2)$
$= A_3$ de CG_k (UAP-BCIoT)

Avec cette deuxième solution (côté U_i), le problème d'authentification vers les dispositifs IoT souhaités est résolu.

5.5 Analyse des performances

Dans cette section, nous effectuons une comparaison détaillée de UAP-BCIoT avec la solution CG_k-side et la solution U_i-side concernant les caractéristiques de sécurité, les coûts de communication et les coûts de calcul. Pour cela, nous avons basé notre analyse sur les résultats expérimentaux de [48].

5.5.1 Comparaison des mesures de sécurité

Srinivas et al. ont prouvé la sécurité de leur protocole conçu (UAP-BCIoT) contre diverses attaques de sécurité, ainsi que ses mesures de sécurité supérieures et ses nombreuses fonctionnalités par rapport à des schémas similaires (Li et al. [70], Porambage et al. [71] et Banerjee et al. [72]). Notre version améliorée côté CG_k (A) résout le problème d'authentification des utilisateurs mais contient une vulnérabilité liée à une attaque par rejeu. Cependant, notre version améliorée côté U_i (B), qui ajoute des fonctions de hachage au protocole UAP-BCIoT, assure un niveau de sécurité et de fonctionnalités équivalent tout en corrigeant la faille d'authentification des utilisateurs dans le UAP-BCIoT.

TABLE 5.2 : Comparaison de la sécurité et des fonctionnalités

Protocol	Li [70]	Por [71]	Ba [72]	UAP-BCIoT [48]	Enhanced A UAP-BCIoT	Enhanced B UAP-BCIoT
A1	✓	×	✓	✓	✓	✓
A2	✓	×	×	✓	✓	✓
A3	✓	–	✓	✓	✓	✓
A4	✓	×	✓	✓	✓	✓
A5	✓	×	×	✓	×	✓
A6	✓	×	✓	✓	✓	✓
A7	✓	×	✓	✓	✓	✓
A8	✓	×	✓	✓	✓	✓
A9	×	×	×	✓	✓	✓
A10	×	×	×	✓	✓	✓
A11	✓	✓	✓	×	✓	✓
A12	✓	✓	✓	✓	✓	✓
A13	✓	✓	×	×	✓	✓

A1 :"Privileged-insider attck" ; A2 :"User anonymity" ; A3 :"Stolen mobile /smart card" ;
A4 : "Main-in-the-middle attack" ; 5 : "Replay attck" ; A6 :"Impersonation attcks" ;
A7 :"DoS attack" ; A8 :"IoT node capture attack ; A9 : "IoT node credential validation" ;
A10 : "Big Data analytics" ; A11 : "Mutual authentication" ; A12 : "Traceability" ;
A13 :"Accessibility × does not resist an attack or not supports an attribute"
✓ "resists an attack or supports an attribute" ; – : "Not applicable in a scheme"

Nous considérons que la deuxième solution (B) est la plus appropriée. Le Tableau 5.2 offre une comparaison complète des caractéristiques de sécurité et des fonctionnalités de l'UAP-BCIoT amélioré par rapport à trois autres schémas. Il est évident que l'UAP-BCIoT amélioré propose des fonctionnalités plus étendues que le UAP-BCIoT [48] et offre des mesures de sécurité supérieures par rapport aux trois autres schémas.

5.5.2 Comparaison des coûts de calcul

Les résultats expérimentaux listés dans le Tableau 5.3 sont utilisés pour calculer le temps d'exécution de diverses opérations cryptographiques. Le coût de calcul pour UAP-BCIoT, UAP-BCIoT amélioré A et B sont respectivement de 225,20, 224,56 et 225,52 millisecondes. Dans la section précédente, nous avons décidé de choisir la version B pour éviter les attaques de sécurité. Le UAP-BCIoT amélioré (B) garantit la sécurité et corrige la faille de l'UAP-BCIoT, avec des coûts de calcul légèrement plus élevés par rapport à UAP-BCIoT.

TABLE 5.3 : Temps d'exécution pour diverses opérations utilisées par UAP-BCIoT

Notations	**Description**	**Execution Time**
T_h	Hash Function	0.32 ms
T_{sed}	Symmetric en/decryption	8.7 ms
T_{ecm}	ECC point multiplication	17.1 ms
T_{eca}	ECC point addition	4.4 ms
T_{fe}	Fuzzy Extractors	17.1 ms

5.5.3 Comparaison des coûts de communication

Pour le calcul du coût de communication, il est supposé que l'identité, le timestamp, le résultat de la fonction de hachage (SHA-1), le certificat (signature utilisant ECDSA) et un nonce aléatoire nécessitent respectivement 160, 32, 160, 320 et 160 bits. Dans l'UAP-BCIoT amélioré, les trois messages échangés $MSG_1 \leftarrow \{A_1, HID_i, G_2, DID_{IN_j}, M_x, TS_1\}$, $MSG_2 \leftarrow \{GI_1, GI_2, M_x, TS_2\}$ et $MSG_3 \leftarrow \{DID^*_{IN_j}, NM_2, M_y, SKV_{ji}, TS_3\}$ nécessitent respectivement 1312 bits, 672 bits et 832 bits. Le Tableau 5.4 présente un résumé du nombre de messages échangés et des coûts de communication pour tous les schémas. Il est clair à partir du Tableau 5.4 que le UAP-BCIoT amélioré (B) présente des coûts de calcul légèrement plus élevés par rapport à UAP-BCIoT, tout en corrigeant la faille de ce dernier.

TABLE 5.4 : Performance Comparison

Performance Parameters		Li et al. [70]	Banerjee et al. [72]	UAP-BCIoT [48]	Enhanced UAP-BCIoT (A)	Enhanced UAP-BCIoT (B)
Computation Costs (in ms)	User	70,96	47,04	125,14	124,82	125,46
	Gateway	19,34	45,1	42,12	41,8	42,12
	IoT node	18,38	18,04	57,94		
	Total Cost	$19T_h + 6T_{ecm}$ 108.68	$19T_h + 10T_{sed}$ $+T_{fe}$= 159.58	$35T_h + 11T_{ecm}$ $+T_{fe} + 2T_{eca}$= 225.20	$33T_h + 11T_{ecm}$ $+T_{fe} + 2T_{eca}$= 224.56	$36T_h + 11T_{ecm}$ $+T_{fe} + 2T_{eca}$= 225.52
Communication Costs (Bits)	MSG_1	800	192	1152	992	1312
	MSG_2	640	1088	672		
	MSG_3	640	1024	832		
	MSG_4	640	-			
	Total Cost	2720	2304	2656	2496	2816

5.6 Conclusion

Dans ce chapitre, nous avons proposé une amélioration du protocole d'authentification UAP-BCIoT existant [48]. Nous avons d'abord démontré que le schéma UAP-BCIoT présente une lacune au niveau de l'authentification côté utilisateur. Pour remédier à cette faiblesse, un schéma amélioré a été introduit. Bien que cette version engendre des coûts de communication et de calcul légèrement supérieurs à ceux de UAP-BCIoT, elle offre des fonctionnalités accrues ainsi qu'une sécurité renforcée par rapport aux autres schémas existants. Cette amélioration permet ainsi de garantir une meilleure protection des utilisateurs tout en maintenant une efficacité opérationnelle optimale.

Par ailleurs, l'IoV ouvre des perspectives prometteuses pour la diffusion en temps réel et l'amélioration de la qualité d'expérience des véhicules terrestres. Cependant, en cas de défaillance des communications, notamment lors d'urgences telles que des embouteillages, fournir des informations instantanées aux usagers reste un défi majeur. Les drones, ou véhicules aériens sans pilote (UAV), apparaissent comme une solution viable pour rétablir les communications d'urgence dans de telles situations critiques. Grâce à leur capacité à couvrir de vastes zones, à intervenir rapidement sur les problématiques urgentes, et à se déplacer à des vitesses bien supérieures à celles des véhicules terrestres, sans être contraints par les infrastructures routières, les drones se révèlent particulièrement efficaces pour la surveillance du trafic et la collecte de données routières.

Le prochain chapitre explorera donc l'intégration des drones afin d'assurer un suivi en temps réel, sécurisé et fiable de zones spécifiques.

Chapitre 6

LASeR : Protocole d'authentification à distance léger et sécurisé pour l'Internet des Drones

Dans ce chapitre, nous présentons l'intégration des drones de manière efficace et robuste afin d'assurer un suivi en temps réel, sécurisé et fiable de zones spécifiques. L'Internet des Drones (IoD) conduit à de nouveaux cas d'utilisation pratiques, tels que la surveillance améliorée du trafic. Toutefois, la communication sans fil utilisée pour l'échange d'informations entre les utilisateurs et les drones présente des vulnérabilités en raison de sa nature ouverte, exposant ainsi le système à divers défis de sécurité. Pour remédier à ce problème crucial dans les environnements IoD, nous proposons dans ce chapitre un protocole d'authentification et de négociation de clé léger, nommé LASeR, qui garantit un échange sécurisé entre les utilisateurs et les drones.

6.1 Introduction

L'automatisation du système de transport global ne peut pas se limiter aux seuls véhicules. D'autres éléments, tels que les tâches des équipes de terrain, la police de la circulation, les inspecteurs routiers ou encore les équipes de secours, doivent également être automatisés. Dans ce contexte, les drones intelligents peuvent jouer un rôle clé. Ces drones sont une technologie innovante pour surveiller les conditions de circulation et collecter des informations. Comparés aux dispositifs traditionnels comme les boucles de détection, les caméras de surveillance ou les capteurs à micro-ondes, les drones sont plus économiques et peuvent surveiller de longs segments de route de façon continue, ou se concentrer sur un segment particulier. De plus, en cas de catastrophe, les infrastructures de communication ou d'électricité peuvent être endommagées, compliquant le contrôle et la collecte de données sur le réseau de transport.

Afin de faciliter l'utilisation des drones, la transmission des données doit être à la fois rapide et sécurisée. Quel que soit l'usage envisagé, l'utilisateur distant doit recevoir des informations en temps réel sur une zone spécifique via un drone déployé sur place. Cependant, la transmission de ces données entre les drones et l'utilisateur se fait par des canaux ouverts, exposant les informations à divers risques de sécurité, tels que la capture de drones, les attaques de type homme du milieu, l'usurpation d'identité, les attaques internes, les attaques par rejeu, et bien d'autres menaces.

Par ailleurs, l'Internet des Drones (IoD) fait face à des contraintes de ressources en raison des capacités limitées en matière de calcul, de communication et d'énergie. Pour prolonger la durée de vie des infrastructures IoD, il est crucial de concevoir des protocoles de communication et de sécurité nécessitant un minimum de ressources. À cette fin, certains chercheurs se concentrent sur le développement de protocoles d'authentification et d'échange de clés pour assurer des communications sécurisées entre l'utilisateur et le drone. Cependant, il est important de souligner que ces protocoles restent vulnérables à divers risques de sécurité. De plus, certains ne sont pas adaptés aux environnements IoD à ressources limitées, en raison de leurs exigences élevées en matière de communication et de calcul.

L'équilibre entre l'efficacité énergétique et la sécurité représente un défi majeur. En effet, garantir un niveau de sécurité plus élevé nécessite généralement une consommation d'énergie accrue de la part des drones. Étant donné que les drones sont des objets mobiles qui se déplacent fréquemment, de nouvelles connexions sont continuellement établies puis rompues. L'authentification et l'échange de clés, qui constituent la première ligne de défense à chaque connexion, deviennent donc des processus essentiels.

C'est dans ce cadre que nous proposons un protocole d'authentification et d'établissement de clé sécurisé et léger basé sur l'ECC (Cryptographie à Courbe Elliptique). L'ECC offre le même niveau de sécurité que le RSA, mais avec des longueurs de clé plus courtes, ce qui le rend plus efficace. Il nécessite également moins de puissance de calcul et de ressources réseau, améliorant ainsi les performances, surtout dans des environnements à ressources limitées. De plus, l'ECC est utilisé pour établir un secret partagé entre deux parties via l'échange de clés Diffie-Hellman basé sur une courbe elliptique (ECDH). Chaque partie génère sa propre paire de clés et échange les clés publiques. Ensuite, elles utilisent leurs clés privées respectives et la clé publique reçue pour calculer un secret partagé, qui servira à créer une clé symétrique sécurisée. Les opérations ECC, qui reposent sur des opérations de groupe additives plutôt que sur des opérations multiplicatives complexes, permettent ainsi de trouver un équilibre optimal entre efficacité et sécurité.

6.1.1 Contribution

Dans ce travail, nous proposons un protocole d'authentification et d'échange de clés sécurisé et à faible consommation de ressources. Pour une meilleure synthèse, nos contributions sont répertoriées ci-dessous :

- Nous avons proposé un protocole d'authentification et d'échange de clés léger, appelé LASeR, utilisant des opérations cryptographiques légers telles que les fonctions de hachage, les opérations XOR et la cryptographie à courbe elliptique, ce qui améliore l'efficacité de notre protocole.
- Dans un système IoD, selon LASeR, un utilisateur enregistré peut avoir un accès direct à un drone dans une zone désignée, mais un accès limité ou restreint dans d'autres zones, en fonction de divers scénarios et types d'utilisateurs.
- L'utilisateur enregistré peut facilement mettre à jour ses mots de passe et/ou ses informations biométriques sur son appareil local à tout moment. De plus, si l'appareil mobile de l'utilisateur enregistré est perdu ou volé, il peut le remplacer par un nouvel appareil.
- Notre protocole LASeR permet une authentification mutuelle sécurisée tout en garantissant l'anonymat de l'utilisateur et l'intracabilité. De plus, par rapport aux protocoles existants, notre protocole démontre un équilibre exceptionnel entre efficacité et sécurité.
- Les résultats obtenus montrent également que LASeR minimise efficacement la consommation d'énergie lors de l'exécution des tâches de calcul et de communication.

Ce chapitre est organisé comme suit. La section 6.2 présente un aperçu des préliminaires et du modèle de réseau. Une description détaillée du protocole proposé est discutée dans la section 6.3. La section 6.4 montre comment le protocole est sécurisé contre diverses attaques. Une comparaison des performances est donnée dans la section 6.5. Enfin, la conclusion est présentée dans la section 6.6.

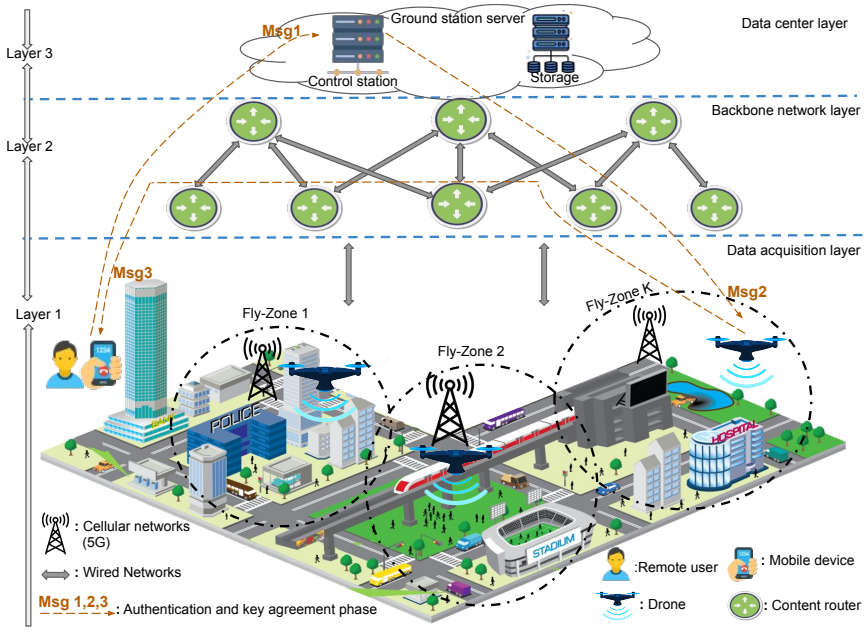


FIGURE 6.1 : Modèle de réseau (architecture) du protocole LASeR proposé

6.2 LASeR : Modèle de Système

Dans cette section, nous nous focalisons sur les modèles de réseau et de menaces associés au LASeR proposé, qui sont essentiels pour l'analyse de sécurité du protocole LASeR.

6.2.1 Modèle de Réseau

La figure 6.1 illustre le modèle de réseau du LASeR proposé, qui se compose de trois couches distinctes. À la Couche 1 (Couche d'Acquisition de Données), nous avons plusieurs zones de vol ($C_k|k = 1,2,3,\ldots,n$), où n est le nombre de zones, Utilisateur ($U_i|i = 1,2,3,\ldots,I$), où I est le nombre d'utilisateurs, et Drone ($D_j|j = 1,2,3,\ldots,J$), où J est le nombre de drones. Différents drones peuvent être déployés pour collecter et surveiller des informations sensibles dans les différentes zones ciblées. De plus, un D_j transmet les données recueillies à la couche supérieure (Couche 2) en utilisant des réseaux sans fil. Les routeurs de contenu gèrent le routage des données vers les serveurs à la Couche 3 (la couche du centre de données) en utilisant des technologies prometteuses (réseaux cellulaires 5G). La station de contrôle participe au processus d'authentification et permet l'accès des utilisateurs aux données IoD stockées sur le serveur s'ils sont autorisés à accéder à ces informations. Cependant, dans les applications sensibles au temps, l'Utilisateur Externe (U_i) souhaite connaître des informations sur une zone particulière (C_k). Pour cette raison, une communication directe entre U_i et D_j déployé dans C_k est nécessaire, avec l'assistance du Serveur de Station au Sol (GSS). En raison de la nature ouverte de la communication, elle est exposée à divers risques de sécurité. Une authentification utilisateur à distance sécurisée et un échange de clés sont nécessaires entre l'utilisateur et le drone.

6.2.2 Modèle de Menace

Pour démontrer la sécurité prouvable de LASeR, nous utilisons les modèles de menace suivants :

- Modèle de menace Dolev–Yao (DY) : Dans ce modèle [53], toute paire d'entités communicantes (utilisateur et drone) dans le réseau peut communiquer via un canal non sécurisé, et les entités de point de terminaison sont considérées comme non fiables. Par conséquent, un attaquant $\mathscr{A}$ peut capturer tous les messages échangés, mais il peut également modifier les messages capturés.
- Modèle de menace de Canetti et Krawczyk : Le modèle de menace bien connu CK [54] est plus puissant que le modèle de menace DY, qui est pris en compte dans les schémas d'authentification récents. Selon le modèle de menace CK, en plus des capacités considérées dans le modèle DY, $\mathscr{A}$ a la capacité de régler les états de session et les informations confidentielles, telles que les clés secrètes. Il est donc essentiel que même si les informations secrètes et les états de session sont compromises, ces informations ne puissent pas compromettre la confidentialité des identifiants d'autres entités. Ainsi, la confidentialité antérieure et ultérieure est maintenue dans la conception

de LASeR sous le modèle de menace CK.

En outre, nous considérons également les capacités suivantes que l'attaquant $\mathscr{A}$ pourrait avoir :

1. L'appareil mobile intelligent n'est pas complètement fiable et est susceptible d'être capturé physiquement par $\mathscr{A}$. De plus, $\mathscr{A}$ a la capacité d'extraire tous les paramètres stockés sur un appareil mobile par l'attaques par analyse de puissance [55].
2. $\mathscr{A}$ a la capacité de deviner l'identité de l'utilisateur et le mot de passe hors ligne [56].
3. L'utilisateur légitime peut également agir en tant qu'attaquant.
4. $\mathscr{A}$ peut infiltrer comme un insider et obtenir le vérificateur du serveur de la Station au Sol.
5. $\mathscr{A}$ peut capturer physiquement le drone et extraire les informations stockées à l'intérieur.

TABLE 6.1 : Liste des notations du protocole LASeR proposé.

Notation	Description
U_i, D_j	$i^{\text{ème}}$ utilisateur, $j^{\text{ème}}$ drone, respectivement
GSS	Serveur de station au sol pour tous les U_i et D_j
MD_i	Appareil mobile de U_i
ID_i, PW_i, BIO_i	Identité, mot de passe, et données biométriques de U_i, respectivement
$ID_{D_j}(Q_{D_j}, X_{D_j})$	Identité et clé publique/secrète de D_j, respectivement
$ID_{GSS}(Q_{GSS}, X_{GSS})$	Identité et clé publique/secrète de GSS, respectivement
E	Courbe elliptique de référence
a, b	Coefficients de la courbe elliptique
P	Un point de base dans $E_p(a,b)$
$k \cdot P$	Une multiplication de point sur la courbe elliptique ; $k \in Z_p^*, P \in E_p(a,b)$
SID_{D_j}	Clé secrète entre GSS et D_j
T_u, T_G, T_d	Horodatages utilisés dans la phase d'authentification de LASeR
T_u^*, T_G^*, T_d^*	Temps de réception des messages utilisés dans la phase d'authentification de LASeR
A_i, B_i	Secrets partagés entre U_i et GSS lors de la phase d'enregistrement.
	A_i est utilisé pour la connexion entre U_i et MD_i
	B_i est utilisé pour la vérification d'authentification entre MD_i et GSS.
ΔT	Délai de transmission maximal autorisé
b_r, u_r, r_D	Nombres aléatoires
SK_d, SK_u	La clé de session générée respectivement du côté de D_j et de U_i
$Gen(\cdot), Rep(\cdot), \tau_t$	Génération de clé par extracteur flou, algorithme de reproduction, et paramètre, respectivement
σ_i	Clé secrète biométrique de U_i
$h(\cdot), \|\|, \oplus$	Fonction de hachage cryptographique, concaténation, et opération XOR bit à bit, respectivement

6.3 Le Protocole LASeR Proposé

Le protocole Proposé (LASeR) comprend cinq phases : 1) pré-déploiement, 2) enregistrement de l'utilisateur mobile, 3) authentification et échange de clés, 4) mise à jour du mot de passe et des données biométriques, et 5) révocation. tableau6.1 répertorie les notations utilisées dans LASeR. Les

composants fondamentaux de LASeR sont la cryptographie à courbe elliptique, les opérations XOR bit à bit, et une fonction de hachage résistante aux collisions pour garantir un accès sécurisé des utilisateurs au drone. De plus, nous utilisons les horodatages actuels et des nombres aléatoires pour résister aux attaques par rejeu. En dehors de la technique d'extracteur flou (FE), spécifiquement utilisée pour la vérification biométrique locale du côté de l'utilisateur. Les sous-sections suivantes expliquent les différentes phases de LASeR.

6.3.1 Phase de Pré-déploiement

Le serveur de la station au sol (GSS) est l'entité de confiance. Le GSS garantit l'enregistrement sécurisé de tous les drones avant leur déploiement. Dans cette phase, le GSS attribue une identité unique à chaque zone de vol (CID_k) et enregistre chaque drone avant son déploiement. L'identité du drone (ID_{D_j}) est ensuite stockée dans le GSS. De plus, le drone reçoit des valeurs secrètes et publiques qui seront utilisées à des fins d'autorisation dans le futur. Pour atteindre cet objectif, le GSS sélectionne la clé secrète du drone (X_{D_j}), puis calcule Q_{GSS}=X_{GSS}·P, Q_{D_j}=X_{D_j}·P et SID_{D_j}=h($CID_k \| ID_{GSS} \| X_{GSS} \| ID_{D_j} \| X_{D_j}$), où X_{GSS}, Q_{D_j} et SID_{D_j} sont respectivement la clé secrète du GSS, la clé publique de D_j, et la clé secrète partagée entre GSS et D_j.

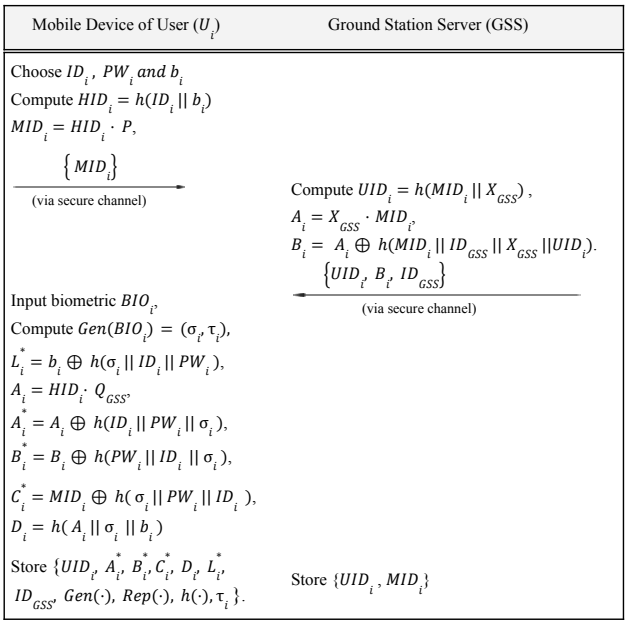


FIGURE 6.2 : Phase d'enregistrement des utilisateurs

6.3.2 Phase d'Enregistrement de l'Utilisateur (UR)

L'utilisateur initie cette phase pour s'enregistrer auprès du GSS, afin d'acquérir les identifiants fournis par le GSS. Ces identifiants serviront de preuve de sa légitimité auprès du GSS. L'utilisateur et le GSS

effectuent la procédure suivante pour accomplir cette phase :

- Étape UR-1 : U_i sélectionne d'abord un ID_i et un mot de passe PW_i uniques. U_i choisit ensuite aléatoirement un nombre b_i et calcule HID_i=$h(ID_i \| b_i)$, une pseudo-identité. Il calcule également et envoie de manière sécurisée MID_i=$HID_i \cdot P$ au GSS.
- Étape UR-2 : À la réception de MID_i de U_i, le GSS procède au calcul de UID_i=$h(MID_i \| X_{GSS})$, A_i=$X_{GSS} \cdot MID_i$, et B_i=$A_i \oplus h(MID_i \| ID_{GSS} \| X_{GSS} \| UID_i)$. Ensuite, il transmet de manière sécurisée $\{UID_i, B_i, ID_{GSS}\}$ avec les identifiants à l'utilisateur et stocke $\{UID_i, MID_i\}$ dans sa base de données.
- Étape UR-3 : Une fois que U_i reçoit $\{UID_i, B_i, ID_{GSS}\}$ du GSS, il procède à l'entrée de ses données biométriques BIO_i dans son appareil et calcule $Gen(BIO_i)$=(σ_i, τ_i) et A_i=$HID_i \cdot Q_{GSS}$. Ensuite, l'utilisateur masque b_i, A_i, B_i, MID_i dans $L_i^*, A_i^*, B_i^*, C_i^*$ respectivement, en utilisant son identité, son mot de passe, et sa clé biométrique, et calcule la valeur de vérification D_i. Enfin, l'utilisateur stocke ces valeurs dans MD_i.

Cette phase est résumée dans la figure 6.2.

6.3.3 Phase de Connexion et d'Authentification (LA)

Durant cette phase, avant d'obtenir des informations en temps réel sur une zone de vol spécifique, l'utilisateur doit s'authentifier et établir une clé secrète avec un drone, assisté par le GSS via un canal ouvert (non sécurisé). Il est supposé que U_i possède les identités CID_k des zones de vol auxquelles il a accès. Une explication détaillée de cette phase est fournie dans les étapes suivantes :

- Étape LA-1 : U_i saisit son identité ID_i et son mot de passe PW_i' dans l'interface de connexion accessible de MD_i. il fournit également ses informations biométriques BIO_i' à MD_i. Après réception des informations de U_i, MD_i calcule σ_i'=$Rep(BIO_i', \tau_i)$ et démasque b_i, A_i, B_i, MID_i, puis calcule la pseudo-identité HID_i, D_i', et vérifie si $D_i' \stackrel{?}{=} D_i$ est valide.
- Étape LA-2 : Après vérification réussie, U_i génère un nonce r_u et un horodatage T_u, puis calcule U_1=$A_i \oplus B_i$, U_2=$CID_k \oplus h(U_1 \| T_u)$, U_3=$r_u \oplus h(CID_k \| T_u \| U_1)$, représentant respectivement l'identité masquée de la zone de vol et le nonce masqué. Ensuite, le jeton d'authentification U_4=$h(U_1 \| CID_k \| r_u \| T_u)$ est calculé pour authentifier le message de demande transmis par l'utilisateur $(UID_i, U_2, U_3, U_4, T_u)$ au GSS.
- Étape LA-3 : À la réception, le GSS vérifie d'abord la fraîcheur de l'horodatage en comparant la différence de temps entre l'horodatage de réception et d'envoi $|T_u^* - T_u| \leq \triangle T$. Si la condition est remplie, le GSS vérifie si UID_i existe dans sa base de données et, si trouvé, récupère le MID_i correspondant. Le GSS calcule également U_1'=$h(MID_i \| ID_{GSS} \| X_{GSS} \| UID_i)$, puis démasque CID_k', r_u' et calcule aussi le jeton d'authentification U_4' pour le vérifier avec le jeton reçu $U_4' \stackrel{?}{=} U_4$.

- Étape LA-4 : Après vérification de l'authenticité du message de l'utilisateur, le GSS sélectionne un horodatage T_G et récupère l'identité ID_{D_j} associée au D_j déployé dans CID'_k demandé par l'utilisateur. Ensuite, le GSS calcule R_u=$r'_u \cdot P$, l, le jeton d'authentification utilisé par l'utilisateur pour vérifier indirectement l'authenticité du GSS. G_1 et G_3 contiennent des secrets partagés entre le GSS et D_j, G_2 comprend r'_u et l, masqués en utilisant G_1. De plus, G_4 est dérivé de la combinaison XOR de MID_i et G_3. Ensuite, le jeton d'authentification G_5=$h(SID_{D_j} \| r'_u \| R_u \| MID_i \| l \| T_u \| T_G)$ est calculé pour authentifier le message de demande transmis par le GSS (R_u, G_2, G_4, G_5, T_u, T_G) à D_j.

- Étape LA-5 : À la réception, D_j vérifie la ponctualité en évaluant la condition $|T^*_G - T_G| \leq \triangle$T. Si la condition est remplie, il procède au calcul de G'_1, et l'utilise pour démasquer les valeurs masquées de r'_u et l. Ensuite, il calcule G'_3 pour démasquer MID_i et aussi le jeton d'authentification G'_5 pour le vérifier avec le jeton reçu $G'_5 \stackrel{?}{=} G_5$.

- Étape LA-6 : D_j génère T_d et r_D, calcule $R_D = r_D \cdot P$ et D_1. Il utilise ensuite cette valeur pour masquer son identité (ID_{D_j}) dans D_2, puis calcule f, W_{D_j}, et Z_{D_j} pour dériver une clé de session SK_d. Ensuite, le jeton d'authentification D_3=$h(SK_d \| CID_k \| ID_{D_j} \| ID_{GSS} \| T_d)$ est calculé pour authentifier le message de réponse transmis par D_j à l'utilisateur (R_D, D_2, D_3, T_d).

- Étape LA-7 : À la réception, U_i vérifie d'abord la fraîcheur de l'horodatage en comparant la différence de temps entre l'horodatage de réception et d'envoi $|T^*_d - T_D| \leq \triangle$T. Si cela est satisfait, il calcule D_1 pour démasquer ID_{D_j}, l, f, W_{U_i}, et Z_{U_i} pour dériver une clé de session SK_u, et calcule aussi le jeton d'authentification D'_3 pour le vérifier avec le jeton reçu $D'_3 \stackrel{?}{=} D_3$. Si c'est le cas, cela suggère que la clé de session calculée par U_i (SK_u) et D_j (SK_d) est identique et cela est affirmé dans la section 4.4.

Les phases de connexion et d'authentification liées à LASeR sont résumées dans 6.3.

6.3.4 Validité des Clés de Session

L'équivalence des clés de session générées par U_i et D_j pendant la phase d'authentification et d'échange de clés peut être justifiée comme suit :

SK_u=$kdf(Z_{U_i} \| T_d \| T_u)$

=$kdf(W_{U_i}(R_D + lQ_{D_j}) \| T_d \| T_u)$

=$kdf((R_D + lQ_{D_j})W_{U_i} \| T_d \| T_u)$

=$kdf((R_D + lQ_{D_j})(r_u + fHID_i) mod\ q \| T_d \| T_u)$

SK_d=$kdf(Z_{D_j} \| T_d \| T_u)$

=$kdf(W_{D_j}(R_u + fMID_i) \| T_d \| T_u)$

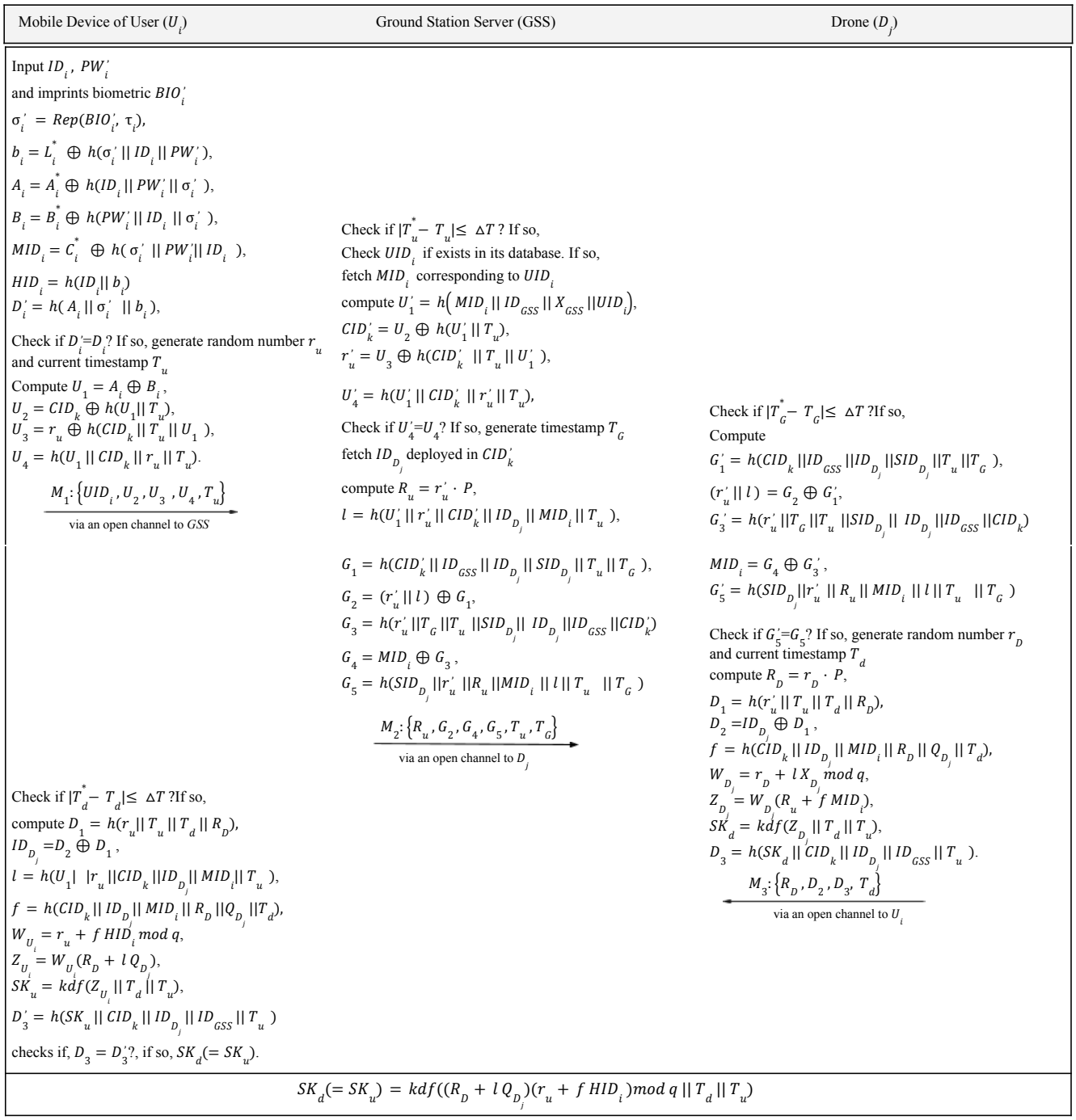


FIGURE 6.3 : Phase de connexion et d'authentification du LASeR proposé.

$=kdf(\ (r_D+lX_{D_j})(R_u+fMID_i) mod\ q\ ||T_d\ ||T_u)$

$=kdf(\ (r_D+lX_{D_j})(r_u+fHID_i)\cdot P\ mod\ q\ ||T_d\ ||T_u)$

$=kdf(\ (r_D+lX_{D_j})\cdot P\ (r_u+fHID_i) mod\ q\ ||T_d\ ||T_u)$

$=kdf(\ (R_D+lQ_{D_j})(r_u+fHID_i) mod\ q\ ||T_d\ ||T_u)$

$=SK_u$

6.3.5 Phase de mise à jour du mot de passe et/ou des données biométriques (PU)

À cette phase, l'utilisateur peut mettre à jour son mot de passe tandis que ses données biométriques restent inchangées, et ses anciennes informations biométriques sont considérées comme récentes ou nouvelles. Cependant, si l'utilisateur souhaite modifier ses données biométriques, il doit fournir de nouvelles informations biométriques localement sur son appareil mobile sans intervention du GSS, en suivant les étapes suivantes.

- Étape PU-1 : U_i entre son ID_i, ancien mot de passe PW_i, et enregistre l'ancienne donnée biométrique BIO_i' sur son appareil mobile. MD_i calcule $\sigma_i' = Rep(BIO_i', \tau_i)$, $A_i = A_i^* \oplus h(ID_i \parallel PW_i \parallel \sigma_i')$, $b_i = L_i \oplus h(\sigma_i' \parallel ID_i \parallel PW_i)$, et vérifie si $D_i \stackrel{?}{=} (A_i \parallel \sigma_i' \parallel b_i)$. Si l'opération réussit, l'appareil mobile informe l'utilisateur d'entrer un nouveau mot de passe PW_i^{new} et de nouvelles données biométriques BIO_i^{new}, puis passe à l'étape suivante.

- Étape PU-2 : U_i entre les nouvelles données biométriques BIO_i^{new}, le nouveau mot de passe PW_i^{new} et les fournit à MD_i. MD_i calcule $B_i = B_i^* \oplus h(PW_i \parallel ID_i \parallel \sigma_i')$, $MID_i = C_i \oplus h(\sigma_i' \parallel PW_i \parallel ID_i)$, $(\sigma_i^{new}, \tau_i^{new}) = Gen(BIO_i^{new})$, $L_i^{new} = b_i \oplus h(\sigma_i^{new} \parallel ID_i \parallel PW_i^{new})$, $A_i^{*new} = A_i \oplus h(ID_i \parallel PW_i^{new} \parallel \sigma_i^{new})$, $B_i^{*new} = B_i \oplus h(PW_i^{new} \parallel ID_i \parallel \sigma_i^{new})$, $C_i^{new} = MID_i \oplus h(\sigma_i^{new} \parallel PW_i^{new} \parallel ID_i)$, et $D_i^{new} = (A_i \parallel \sigma_i^{new} \parallel b_i)$.

- Étape PU-3 : Enfin, U_i met à jour les valeurs de $\{A_i^*, B_i^*, C_i^*, D_i, L_i^*, \tau_i\}$ avec $\{A_i^{*new}, B_i^{*new}, C_i^{new}, D_i^{new}, L_i^{new}, \tau_i^{new}\}$, respectivement.

Cette phase est résumée dans la figure 6.4.

6.3.6 Phase de révocation et de ré-enregistrement de l'utilisateur (RR)

Si le dispositif mobile MD_i d'un utilisateur légitime U_i est volé ou perdu, les étapes suivantes sont utilisées pour obtenir un nouveau dispositif mobile.

- Étape RR-1 : U_i doit se souvenir de la même identité ID_i et sélectionner un nouveau mot de passe PW_i. Ensuite, U_i génère un nonce aléatoire b_i' et calcule HID_i=$h(ID_i \parallel b_i')$, MID_i=$HID_i \cdot P$, et envoie MID_i au GSS de manière sécurisée.

- Étape RR-2 : Après avoir obtenu MID_i de U_i, le GSS calcule UID_i=$h(MID_i \parallel X_{GSS})$, A_i=$X_{GSS} \cdot MID_i$, et B_i=$A_i \oplus h(MID_i \parallel ID_{GSS} \parallel X_{GSS} \parallel UID_i)$, puis envoie $\{UID_i, B_i, ID_{GSS}\}$ à U_i de manière sécurisée.

- Étape RR-3 : Après avoir reçu $\{UID_i, B_i, ID_{GSS}\}$ du GSS, U_i saisit ses nouvelles données biométriques BIO_i^{new} dans le capteur de son appareil mobile et calcule $Gen(BIO_i^{new})$=$(\sigma_i^{new}$,

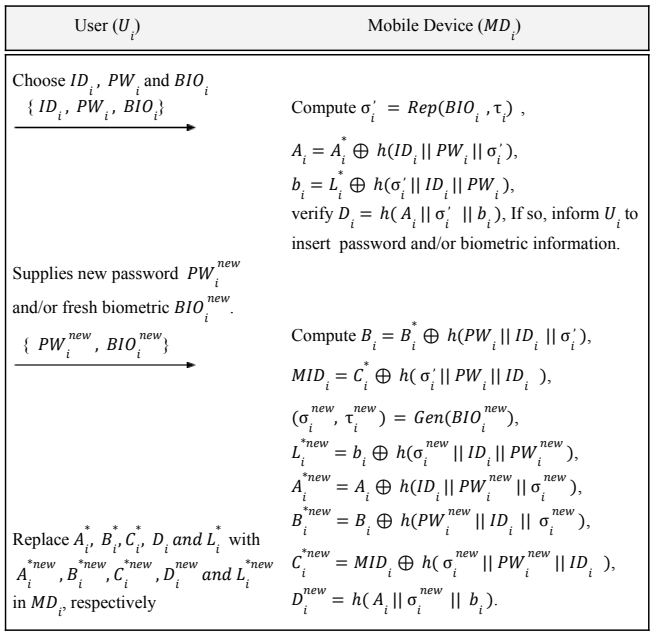


FIGURE 6.4 : Phase de mise à jour du mot de passe et/ou des données biométriques.

τ_i), L_i^{*new}=$b_i \oplus h(\sigma_i^{new} \parallel ID_i \parallel PW_i^{new})$, A_i=$HID_i \cdot Q_{GSS}$, A_i^{*new}=$A_i \oplus h(ID_i \parallel PW_i^{new} \parallel \sigma_i^{new})$, B_i^{*new}=$B_i \oplus h(PW_i^{new} \parallel ID_i \parallel \sigma_i^{new})$, C_i^{*new}=$MID_i \oplus h(\sigma_i^{new} \parallel PW_i^{new} \parallel ID_i)$, et D_i^{new}=$(A_i \parallel \sigma_i^{new} \parallel b_i')$. Enfin, U_i stocke $\{UID_i, A_i^{*new}, B_i^{*new}, C_i^{*new}, D_i^{new}, L_i^{*new}, ID_{GSS}, Gen(.), Rep(.), h(.), \tau_i\}$ dans MD_i.

Cette phase est résumée dans la figure 6.5.

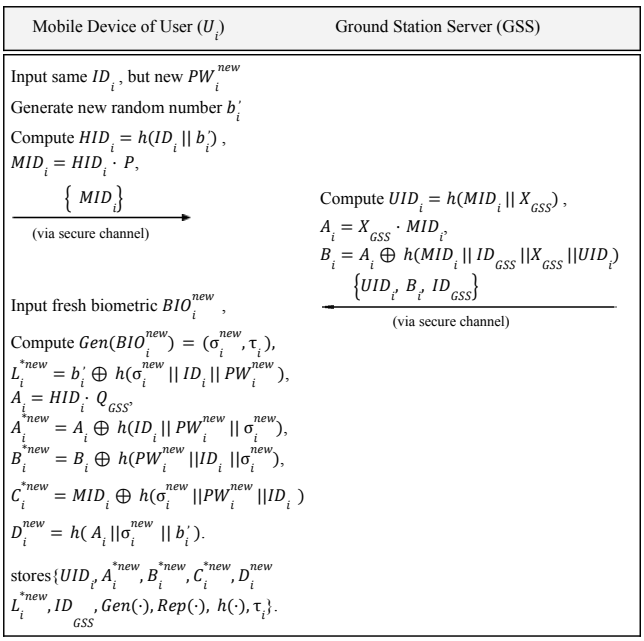


FIGURE 6.5 : Phase de révocation et de réenregistrement de l'utilisateur

6.4 Analyse de sécurité

Cette section présente une analyse formelle de LASeR à l'aide de trois outils de vérification, à savoir AVISPA, Scyther et Tamarin. De plus, des analyses de sécurité informelles sont effectuées sur LASeR afin de démontrer sa résistance contre différentes attaques malveillantes.

6.4.1 Vérification formelle : AVISPA

Dans cette section, nous utilisons l'outil AVISPA [73] pour fournir la preuve que LASeR reste sécurisé contre les adversaires actifs et passifs. Le protocole LASeR proposé est codé en langage HLPSL (High Level Protocol Specification Language), qui est un langage orienté rôle [74]. AVISPA est un outil de simulation automatisé pour vérifier la sécurité des protocoles Internet ainsi que des applications intégrant quatre moteurs principaux, à savoir (i) on-the-fly model-checker (OFMC) ; (ii) constraint-logic-based attack searcher (CL-AtSe) ; (iii) SAT-based Model-Checker (SATMC) ; and (iv) Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP). Il y a trois rôles de base qui doivent être spécifiés lors des phases de connexion, d'authentification et d'échange de clés de session de LASeR, à savoir l'utilisateur U_i, le serveur de la station au sol GSS, et le drone D_j. En plus des rôles de base, les rôles composites (la session, l'objectif et l'environnement) sont obligatoires. Nous avons choisi les moteurs OFMC et CL-AtSe, couramment utilisés, pour notre analyse. Cependant, étant donné que le protocole LASeR repose sur l'opération XOR au niveau des bits, nous ignorons les résultats de vérification des moteurs TA4SP et SATMC, car ils ne prennent pas

en charge les opérations XOR au niveau des bits.

Le protocole LASeR proposé est simulé à l'aide de l'outil SPAN (Security Protocol ANimator for AVISPA) et les résultats de vérification sont résumés dans la figure 6.6, qui montre que LASeR protège à la fois contre les attaques de type homme du milieu et de rejeu.

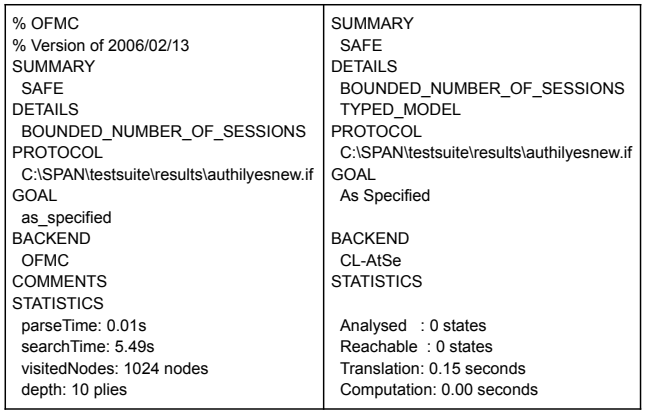


FIGURE 6.6 : Résultats de la simulation avec les backends CL-AtSe et OFMC.

6.4.2 Vérification formelle : Scyther

Pour fournir des preuves supplémentaires de la sécurité du protocole LASeR proposé, nous avons utilisé l'outil formel renommé appelé Scyther [75]. Parmi les outils disponibles, Scyther est le seul capable de vérifier la synchronisation, un aspect crucial dans les protocoles d'authentification [76]. Scyther est un outil puissant de vérification, d'analyse et de falsification des protocoles de sécurité pour identifier les vulnérabilités et les attaques potentielles, et utilise plusieurs modèles d'adversaires comme celui de Dolev-Yao (DY) et d'autres modèles, tels que Canetti-Krawczyk (CK), le modèle CK étendu (eCK), etc. Il prend spécifiquement en charge la révélation aléatoire, la révélation de la clé de session/clé à long terme, la révélation de l'état de session et le secret de transmission parfait, les attaques de clé de session connue, et les attaques de compromission de clé (éphémère). Le protocole LASeR proposé est codé en SPDL (Security-Protocol Description-Language), qui est similaire au langage de programmation Python. Nous avons pris en compte diverses revendications pour modéliser les propriétés de sécurité souhaitées, telles que Secret, Alive, Nisynch, Niagree et Weakagree. Par exemple, la revendication Secret est utilisée pour vérifier la confidentialité du terme d'entrée et Weakagree est utilisée pour valider la capacité du protocole à résister aux attaques d'usurpation d'identité.

La Figure 6.7 présente les résultats de l'analyse de sécurité du protocole LASeR à l'aide de Scyther. D'après la Figure 6.7, toutes les revendications sont "Aucune attaque dans la limite". Scyther permet de définir le nombre maximum d'exécutions. Cinq exécutions sont disponibles par défaut, donc si Scyther renvoie "Aucune attaque dans la limite", cela signifie qu'après avoir vérifié Scyther cinq fois ou moins, aucune attaque n'a été trouvée. Cependant, il est possible de détecter des attaques si le

nombre de limites de vérification est augmenté au-delà de cinq. Pour certains schémas, augmenter le nombre maximum d'exécutions peut conduire à la découverte d'une attaque. LASeR a été testé dans deux scénarios : "5" et "10" exécutions, obtenant des résultats similaires comme indiqué dans la figure 6.7.

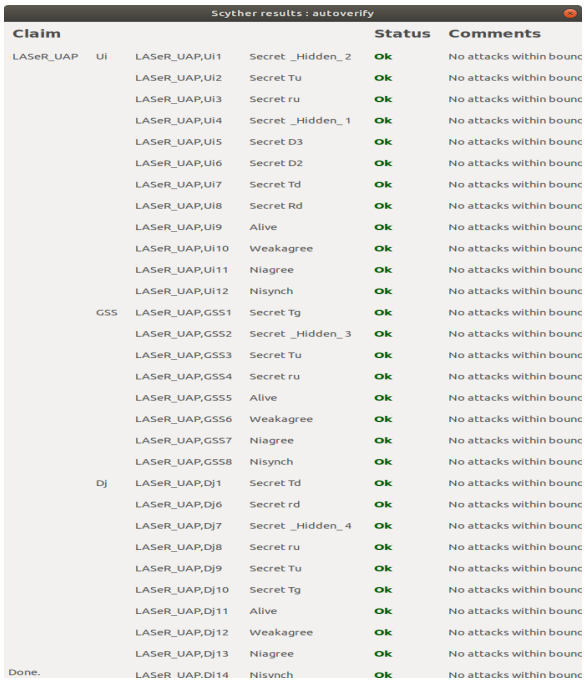


FIGURE 6.7 : Résultats de l'analyse de sécurité de LASeR avec Scyther.

6.4.3 Vérification formelle : Tamarin

Le vérificateur Tamarin utilise des méthodes symboliques pour la vérification des protocoles, en particulier la vérification de modèles symboliques. Il permet aux experts en cryptographie d'analyser des mécanismes de sécurité complexes et d'identifier les vulnérabilités potentielles. Pour fonctionner, Tamarin nécessite une structure de protocole, les détails de l'adversaire, et les fonctionnalités prévues du protocole [77]. Le protocole décrit les rôles des agents, tels que le nouvel arrivant, le répondant et le serveur de clés. Avec des lemmes et des sessions illimitées, Tamarin vérifie automatiquement les principaux aspects de sécurité tels que le secret, l'authenticité et l'autorisation. Il prend également en charge des modèles d'adversaires pour analyser le trafic en temps réel et permet des spécifications détaillées du schéma pour des déductions efficaces.

Le schéma de sécurité proposé, développé en Haskell, définit trois objectifs de sécurité : le secret, l'accord injectif et l'accord non injectif. Les lemmes garantissent que le modèle implémente le schéma en toute sécurité. Tamarin accepte les fichiers d'entrée avec une extension .spthy. Il existe deux méthodes de déploiement : utiliser un navigateur web de manière interactive ou un terminal. L'outil

propose un serveur web interactif avec des pages JavaScript et HTML [78]. L'outil Tamarin vérifie

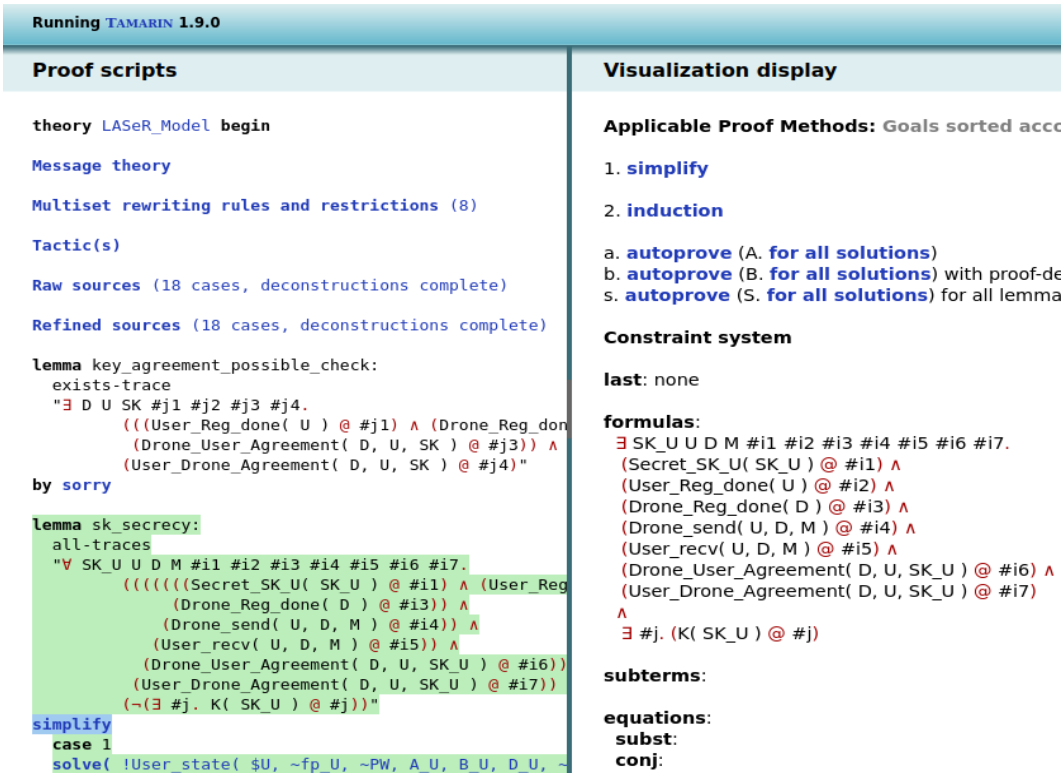


FIGURE 6.8 : Résultats de l'analyse de sécurité de LASeR avec Tamarin.

plusieurs attributs de sécurité. Pour la disponibilité, il établit la validité du schéma à travers des sessions illimitées. La confidentialité est assurée par le secret de la clé secrète de l'utilisateur. L'authentification est vérifiée par l'authentification de l'utilisateur. Nous avons effectué la vérification, et l'outil a démontré que les lemmes sont toujours vrais (voir la figure 6.8). Cela signifie qu'il n'existe aucune exécution possible du protocole dans laquelle la clé secrète est exposée à une partie non autorisée.

6.4.4 Analyse de sécurité informelle

Dans cette section, LASeR est analysé de manière informelle pour démontrer sa résistance face à diverses attaques.

6.4.4.1 Authentification mutuelle

Dans le protocole LASeR, le processus d'authentification consiste à vérifier la validité de D_i' pour authentifier U_i par MD_i. Lorsque M_1 est reçu, le GSS valide U_i en vérifiant UID_i et U_4. Une fois la validation réussie, le GSS procède à l'authentification de U_i. De même, lorsque M_2 est reçu, D_j valide le GSS en vérifiant G_5. Après une validation réussie, D_j authentifie le GSS. De plus, après réception de M_3, U_i valide D_j en vérifiant D_3. Une fois la validation réussie, U_i authentifie indirectement le GSS

et directement D_j. Une fois l'authentification mutuelle accomplie, U_i effectue une vérification de la clé de session à la fin pour s'assurer que D_j et U_i partagent une clé de session commune. Par conséquent, le protocole LASeR réalise une authentification mutuelle sécurisée.

6.4.4.2 Attaque par rejeu

Dans LASeR, les messages échangés M_i (où i = 1, 2, 3) incluent des horodatages générés T_u, T_G et T_d. Si un attaquant $\mathscr{A}$ intercepte ces messages et tente de rejouer un ancien message pour tromper un participant, LASeR empêche ce type d'attaque. Les horodatages jouent un rôle crucial pour garantir la fraîcheur des messages. Cela signifie que le protocole LASeR assure la fraîcheur des messages, permettant ainsi de détecter et de prévenir les attaques par rejeu.

6.4.4.3 Attaques d'usurpation d'identité (IA)

Dans cette attaque, l'objectif de l'adversaire est d'usurper l'identité de participants légitimes. Nous avons les trois cas distincts suivants :

- Usurpation d'identité de l'utilisateur : Pendant l'exécution de LASeR, supposons qu'un adversaire $\mathscr{A}$ tente de capturer M_1 (UID_i, U_2, U_3, U_4, T_u). Maintenant, $\mathscr{A}$ essaie de collecter certains identifiants pour générer un M_1 valide au nom de U_i. Cependant, en raison du manque de connaissances des paramètres ID_i, b_i et U_1, il est difficile pour $\mathscr{A}$ de construire le message M_1 au nom d'un U_i légitime. Ainsi, LASeR assure la sécurité contre les attaques d'usurpation d'identité de U_i.

- Usurpation d'identité du GSS : Supposons qu'un adversaire $\mathscr{A}$ tente de capturer M_2 (R_u, G_2, G_4, G_5 T_u, T_G). Maintenant, $\mathscr{A}$ essaie de collecter des identifiants pour générer un M_2 valide au nom du GSS. Cependant, en raison du manque de connaissances des paramètres SID_{D_j}, X_{GSS} et G_1, il est difficile pour $\mathscr{A}$ de construire le message M_2 au nom du GSS. Ainsi, le protocole LASeR est sécurisé contre les attaques d'usurpation d'identité du GSS.

- Usurpation d'identité du drone : Supposons qu'un adversaire $\mathscr{A}$ tente de capturer M_3 (R_D, D_2, D_3, T_d). Maintenant, $\mathscr{A}$ essaie de collecter certains identifiants pour générer un M_3 valide au nom du drone (D_j). Cependant, en raison du manque de connaissances des paramètres X_{D_j}, r_u et l, il est difficile pour $\mathscr{A}$ de construire le message M_3 au nom du drone (D_j). Ainsi, LASeR assure également la sécurité contre les attaques d'usurpation d'identité du drone.

6.4.4.4 Attaque de l'homme du milieu

Selon le modèle de menace décrit dans la section 3.4, supposons que $\mathscr{A}$ intercepte et modifie tous les messages transmis pendant la phase de connexion et d'authentification via un canal ouvert. Maintenant,

$\mathscr{A}$ peut essayer de modifier les messages échangés pour convaincre le GSS, U_i et D_j. Si $\mathscr{A}$ tente de modifier M_1, il lui faudrait altérer U_2, U_3 et U_4, ce qui nécessite la connaissance des paramètres secrets U_1, HID_i et r_u. Le problème reste le même pour les autres messages M_2 et M_3. Cela démontre clairement que LASeR résiste aux attaques de type "homme du milieu".

6.4.4.5 Intraçabilité et anonymat de l'utilisateur

Supposons qu'un adversaire $\mathscr{A}$ intercepte les messages M_i (où i = 1, 2, 3) pendant la phase d'authentification de LASeR. Il est important de noter que ces messages sont générés à l'aide de divers horodatages et nombres aléatoires. Ces éléments contribuent à la création de messages uniques et aléatoires pour chaque session. Par conséquent, $\mathscr{A}$ est incapable d'établir une corrélation entre deux messages provenant de sessions différentes, garantissant ainsi l'intraçabilité dans le protocole LASeR. De plus, LASeR-UAP ne transmet pas d'informations d'identité réelles de U_i sous forme de texte en clair. Ainsi, LASeR-UAP garantit l'anonymat de U_i.

6.4.4.6 Attaque de divulgation de secrets éphémères (ESL)

Dans LASeR, la clé de session calculée par D_j partagée avec U_i est SK_d=$kdf((R_D+lQ_{D_j})(r_u+fHID_i) mod\ q \| T_d \| T_u)$ (=SK_u). Il convient de mentionner que la clé de session SK_d (=SK_u) est basée à la fois sur des secrets à court terme et à long terme. Nous examinerons la fiabilité de la clé de session de LASeR dans les cas suivants : *Cas 1 :* Si $\mathscr{A}$ compromet uniquement les paramètres secrets éphémères (r_u, r_D), il devient impossible de générer une clé de session valide sans la connaissance des paramètres secrets à long terme restants X_{GSS}, SID_{D_j}, HID_i et X_{D_j}. *Cas 2 :* Si $\mathscr{A}$ révèle uniquement les paramètres secrets à long terme X_{GSS}, SID_{D_j}, HID_i et X_{D_j}, la génération d'une clé de session valide est impraticable sans la connaissance des paramètres secrets éphémères supplémentaires (r_u, r_D).

Il est évident d'après les discussions précédentes que la clé de session est calculée uniquement lorsque les paramètres secrets à long terme et éphémères sont disponibles, une condition atteignable uniquement par les participants légitimes du réseau. Ainsi, dans LASeR, l'adversaire est incapable de déduire la clé de session au nom des participants légitimes. En outre, même si la clé de session actuelle ou spécifique est compromise, les clés de session futures et passées restent uniques et entièrement distinctes pour chaque session. Par conséquent, LASeR montre une résilience contre l'attaque ESL.

6.4.4.7 Attaque par vol de dispositif mobile et attaque de l'initié privilégié

Supposons qu'un attaquant $\mathscr{A}$ parvienne à obtenir le dispositif mobile perdu ou volé MD_i d'un utilisateur légitime U_i. En utilisant des attaques d'analyse de puissance [55], $\mathscr{A}$ peut extraire les identifiants suivants de MD_i $\{UID_i, A_i^*, B_i^*, C_i, D_i, L_i, ID_{GSS}, Gen(.), Rep(.), h(.), \tau_i\}$. Maintenant, envisageons un scénario où $\mathscr{A}$ est un utilisateur privilégié du GSS, agissant comme un attaquant interne.

Dans cette situation, $\mathscr{A}$ connaît $\{MID_i\}$ appartenant à un utilisateur enregistré. Cependant, malgré cette connaissance, $\mathscr{A}$ est incapable de calculer la clé de session SK_u=$kdf((R_D$+$lQ_{D_j})(r_u$+$fHID_i)mod\ q\ \|T_d\ \|T_u)$, où HID_i=$h(ID_i\ \|b_i)$, car $\mathscr{A}$ ne dispose d'aucune information concernant U_i (comme son mot de passe PW_i). Même si $\mathscr{A}$ possède le MD_i perdu ou volé, il est incapable de compléter la vérification biométrique BO_i requise par notre protocole.

6.4.4.8 Attaque par capture de drone

Selon le modèle de menace présenté dans la section 6.2.3, un adversaire $\mathscr{A}$ peut capturer le dispositif D_j en raison de son emplacement dans un environnement non surveillé. Par conséquent, $\mathscr{A}$ peut obtenir des paramètres sensibles tels que ID_{D_j}, SID_{D_j} et X_{D_j}. Cependant, compromettre ces paramètres ne donne pas accès aux paramètres secrets des autres drones D_j non compromis, car les paramètres sensibles sont distincts pour chaque appareil. En outre, les paramètres sensibles compromis de D_j sont insuffisants pour dériver la clé de session (SK), qui est partagée entre les drones D_j non compromis et U_i. Ainsi, LASeR reste sécurisé contre l'attaque de capture de drone.

6.4.4.9 Attaque par déni de service (DoS)

Pour initier une demande d'authentification utilisateur (UA) légitime vers le GSS, U_i doit fournir des entrées telles que ID_i, PW_i' et son empreinte biométrique BIO_i'. De plus, MD_i calcule σ_i' en utilisant la fonction $Rep(BIO_i', \tau_i)$ et révèle les valeurs de b_i, A_i, B_i et MID_i en utilisant l'identité, le mot de passe et la clé biométrique. Il vérifie ensuite si la condition ($D_i' \stackrel{?}{=} D_i$) est valide. Si la condition est remplie, MD_i envoie la demande d'UA au GSS. Cependant, si la condition n'est pas remplie, MD_i termine immédiatement le processus de connexion et interrompt la procédure d'UA pour empêcher l'envoi d'un grand nombre de demandes d'authentification au GSS. Par conséquent, le protocole LASeR proposé est résistant aux attaques DoS.

6.4.4.10 Attaque par mise à jour de mot de passe et/ou de données biométriques

Supposons qu'un attaquant $\mathscr{A}$ parvienne à produire le dispositif mobile perdu ou volé MD_i d'un utilisateur légitime U_i et extrayent les identifiants, tels que $\{UID_i, A_i^*, B_i^*, C_i, D_i, L_i, ID_{GSS}, Gen(.), Rep(.), h(.), \tau_i\}$ en utilisant des attaques d'analyse de puissance [55]. Maintenant, $\mathscr{A}$ tente de mettre à jour les informations de mot de passe/biométrie de U_i afin d'utiliser MD_i pour communiquer avec D_j. Pour cela, $\mathscr{A}$ doit générer de faux PW_i^A, BIO_i^A et ID_i^A et calculer A_i^A, b_i^A, et vérifier la condition $D_i^A \stackrel{?}{=} D_i$. Il est presque impossible pour $\mathscr{A}$ de réussir cette attaque sans ces secrets. Ainsi, la mise à jour des identifiants PW_i, σ_i, A_i, b_i et ID_i par un adversaire est impossible dans LASeR, protégeant ainsi contre les attaques de mise à jour de mot de passe et/ou biométriques.

TABLE 6.2 : Comparaison de la sécurité et des fonctionnalités du protocole LASeR par rapport aux protocoles existants

Reference	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
[38]	✓	✗	✗	✓	✓	✓	✗	✗	✗	✓	✓
[47]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
[42]	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
[43]	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓
[79]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
[80]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
[67]	✓	✓	✗	✗	✓	✓	✓	✗	✗	✓	✓
LASeR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

P1 :"Mutual authentication" ; P2 :"Replay attack" ; P3 :"Impersonation attack" ;
P4 :“Man-in-the-middle attack” ; P5 : “Anonymity” ; P6 :“Traceability” ;
P7 : “Ephemeral Secret Leakage Attack” ; P8 :“Stolen Mobile device” ;
P9 : “Privileged-Insider Attacks” ; P10 :“DoS” ; P11 :"Scalability" ;

6.5 Analyse des performances

Cette section évalue les performances du protocole proposé en comparaison avec les protocoles existants. L'évaluation se concentre sur quatre paramètres essentiels : les fonctionnalités de sécurité, les coûts computationnels et de communication, ainsi que l'énergie consommée. Les sous-sections suivantes fournissent un compte rendu détaillé des résultats. Pour la phase de simulation, nous avons utilisé l'environnement de simulation introduit par [42]. Dans leur configuration, les auteurs ont simulé les drones et dispositifs mobiles en utilisant des Raspberry Pi-3 avec "Ubuntu 16.04 LTS", un processeur "Quad-Core 1,2 GHz", et "1 Go de RAM". Pour simuler le serveur, ils ont utilisé un processeur "Intel(R) Core(TM) (i7-6700) CPU (3,40 GHz)" avec "Ubuntu LTS-16.4" et "8 Go de RAM". Pour évaluer et comparer les coûts computationnels des différentes opérations cryptographiques sur ces deux plateformes, ils ont utilisé "la bibliothèque cryptographique Multiprecision Integer and Rational Arithmetic" (MIRACL).

6.5.1 Fonctionnalités et sécurité

Dans cette sous-section, nous examinons les fonctionnalités et la sécurité du protocole LASeR proposé en comparaison avec les protocoles présentés par [38], [42], [47], [43], [67], [79], et [80]. tableau6.2 illustre les comparaisons, avec une coche (✓) indiquant que le protocole mentionné résiste à l'attaque ou offre la fonctionnalité, tandis qu'une croix (✗) indique que le protocole mentionné n'offre pas la fonctionnalité requise. Le tableau 6.2 met en évidence que les autres protocoles sont vulnérables

TABLE 6.3 : Le temps d'exécution des opérations utilisées pour évaluer LASeR et les protocoles existants.

Cryptographic Operation	Notation	Device	Server
ECC point multiplication	T_{sm}	2.42 ms	0.745 ms
ECC point addition	T_{sa}	0.134 ms	0.003 ms
Symmetric de(en)cryption	$T_{enc/dec}$	0.425 ms	0.07 ms
Hash function	T_h	0.381 ms	0.063 ms
Fuzzy extractor reproduction	T_{fe}	2.42 ms	0.745 ms

à diverses attaques de sécurité, telles que les attaques de relecture, d'usurpation d'identité, de vol de dispositifs mobiles et les attaques ESL. De plus, LASeR excelle en résistant à diverses attaques de sécurité tout en offrant des fonctionnalités telles que l'anonymat, la traçabilité et la scalabilité. Par conséquent, LASeR excelle par rapport aux protocoles précédents en matière de sécurité et de fonctionnalités.

TABLE 6.4 : Comparaison des coûts computationnels et communicationnels du protocole LASeR par rapport aux protocoles existants.

Protocol	U_i	GSS	D_j	Comp Cost	Comm Cost
[38]	$9T_h$= 3.429	$7T_h$ + $2T_{enc/dec}$= 0.581	$7T_h$ = 2.667	6.677 ms	2560 bits
[42]	$6T_h$ + $3T_{enc/dec}$+ $3T_{sm}$+ T_{fe}= 13.241	$2T_h$ +$3T_{enc/dec}$ +$2T_{sm}$=1.826	$3T_h$+$2T_{enc/dec}$ +$2T_{sm}$=6.833	21.900 ms	1984 bits
[47]	$12T_h$ + T_{fe}= 7.00	$9T_h$ = 0.567	$8T_h$+ T_{fe} = 5.468	13.035 ms	3104 bits
[43]	$1T_h$+$2T_{sm}$=9.031	$6T_h$+T_{sm}=1.123	$6T_h$ = 2.286	12.44 ms	2464 bits
[67]	$9T_h$+T_{sm}== 5.849	$4T_h$= 0.252	$3T_h$+T_{sm} = 3.56	9.661 ms	3072 bits
[79]	$12T_h$+$3T_{sm}$=11.832	$11T_h$+$2T_{sm}$=1.931	$11T_h$ = 4.191	17.954 ms	3072 bits
[80]	$8T_h$ + $2T_{enc/dec}$+T_{fe} +$2T_{sm}$+T_{sa}=11.292	$4T_h$ +$2T_{sm}$ +$2T_{enc/dec}$=1.882	$8T_h$+$2T_{enc/dec}$ +T_{sa}=4.032	17.206 ms	2880 bits
LASeR	$13T_h$ + T_{fe}= 7.373	$8T_h$+ T_{sm} = 1.249	$6T_h$+T_{sm} = 4.706	13.328 ms	2464 bits

6.5.2 Analyse des coûts computationnels

Pour évaluer le coût computationnel associé à la phase de connexion et d'authentification du protocole LASeR, nous prenons en compte le nombre d'opérations cryptographiques distinctes effectuées par l'utilisateur, le GSS et le drone, ainsi que leur temps d'exécution. D'après [42], les temps d'exécution correspondant aux opérations cryptographiques sont présentés dans le tableau 6.3. Sur la base de cette tableau, nous présentons la comparaison des coûts computationnels dans le tableau 6.4. Ce graphique à barres 6.9 compare les coûts computationnels de la phase d'authentification dans plusieurs études. Akram et al.[38] présente le coût le plus bas, mais reste vulnérable à plusieurs types d'attaques. En revanche, [42], [79], et [80] montrent des coûts plus élevés. La plupart des autres études, telles que [43] et [67], ainsi que notre protocole LASeR, se situent entre 10 et 13 ms. Cependant, ces deux protocoles

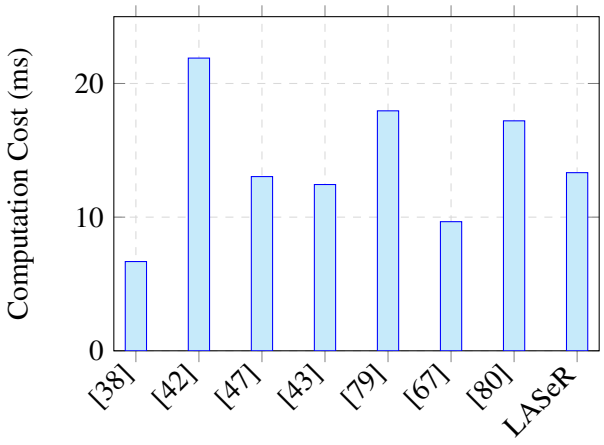


FIGURE 6.9 : Comparaison du coût computationnel pour compléter la phase d'authentification.

restent vulnérables, tandis que LASeR démontre un équilibre efficace par rapport aux autres protocoles. Ainsi, LASeR est bien adapté aux environnements IoD.

6.5.3 Analyse des coûts de communication

Le coût de communication représente le nombre de bits échangés dans les messages au cours de la phase d'authentification. Trouver un équilibre entre minimiser le coût de communication et préserver les fonctionnalités de sécurité est crucial. Pour comparer les coûts de communication, les éléments suivants sont pris en compte : identité (160 bits), nonce (128 bits), sortie de hachage SHA-2 (256 bits), et horodatage (32 bits). Nous supposons que le niveau de sécurité fourni par une ECC de 160 bits est équivalent à celui du système RSA. Au cours de la phase d'authentification du protocole LASeR proposé, trois messages sont échangés : $M_1 : \{UID_i, U_2, U_3, U_4, T_u\}$, $M_2 : \{R_u, G_2, G_4, G_5, T_u, T_G\}$, et $M_3 : \{R_D, D_2, D_3, T_d\}$ dont les tailles sont respectivement $\{256 + 160 + 128 + 256 + 32\}= 832$ bits, $\{160 + (128 + 256) + 160 + 256 + 32 + 32\}=1024$ bits, et $\{160 + 160 + 256 + 32\}= 608$ bits. Par conséquent, le coût total de communication du protocole LASeR est de 2464 bits. Les coûts de communication des protocoles mentionnés sont calculés en utilisant la même méthodologie. La figure 6.10 présente un graphique à barres comparant les coûts de communication (en bits) pour trois entités : l'utilisateur, le drone et le GSS, dans plusieurs études publiées. Ce graphique illustre la quantité de données envoyées par chacune de ces entités dans chaque étude. L'étude [79] se distingue par un coût de communication très élevé du côté de l'utilisateur, atteignant près de 1 500 bits, avec des coûts légèrement inférieurs pour le GSS et le drone. En revanche, dans [80], les coûts sont plus élevés pour le GSS, autour de 1 500 bits, tandis que ceux de l'utilisateur et du drone sont plus bas. Dans les études [43], [42], ainsi que dans notre protocole LASeR, les coûts de communication sont relativement équilibrés entre l'utilisateur et le drone. Cependant, ces deux protocoles sont vulnérables, et celui de [42] présente également un coût computationnel élevé, mettant en évidence l'efficacité de notre protocole LASeR en comparaison.

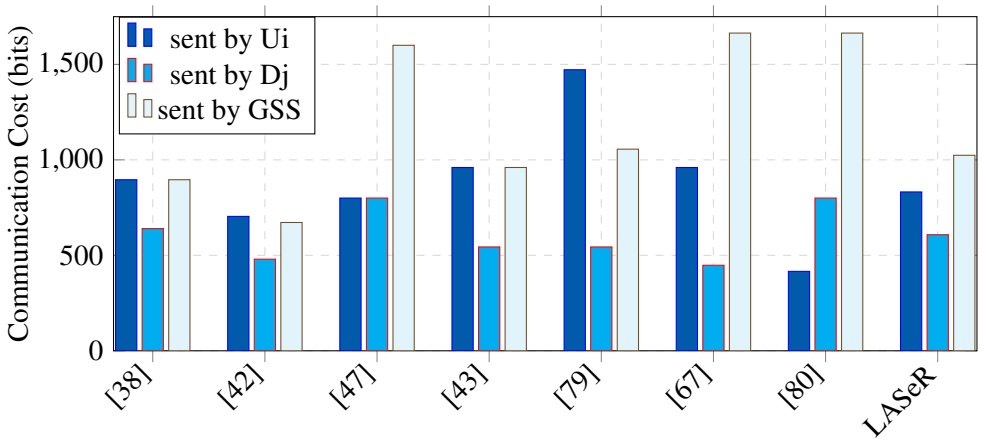


FIGURE 6.10 : Comparaison du coût de communication entre Ui, GSS et Dj

TABLE 6.5 : Comparaison de la consommation d'énergie computationnelle (μJ)

Protocols	Dj	Ui
[38]	22.077	28.293
[47]	56.389	109.495
[42]	45.627	57.750
[43]	18.891	74.007
[67]	29.370	47.526
[79]	34.576	97.386
[80]	33.264	92.913
LASeR	38.835	60.589

6.5.4 Coût énergétique computationnel

Dans cette section, la consommation d'énergie associée aux drones en termes de calcul est discutée. Dans la première partie, l'analyse implique l'évaluation des coûts énergétiques associés à l'exécution de différentes fonctions cryptographiques. Pour ce faire, l'énergie est calculée en utilisant la formule : énergie = courant × tension × T, où courant (cur) désigne le courant et tension (vol) représente la tension associée au Raspberry (PI-3) (vol= 3,3 V, cur=2500 mA) [42]. De plus, le paramètre T représente le temps nécessaire pour exécuter une primitive cryptographique spécifique. Par conséquent, la consommation énergétique globale liée au protocole LASeR côté drone peut être calculée comme cur × vol × $(6T_h + T_{sm})$ = 25,38 μJ. Le coût énergétique computationnel côté drone des protocoles liés est calculé en utilisant la même méthodologie. Nous présentons la comparaison de la consommation énergétique globale de l'utilisateur et du drone dans le tableau 6.5.

TABLE 6.6 : Comparaison du coût de communication au niveau du drone (bits)

Protocol	Received by Dj	Sent by Dj
[38]	896	640
[42]	672	480
[47]	1600	800
[43]	960	544
[67]	1216	448
[79]	1056	544
[80]	1664	800
LASeR	1024	608

6.5.5 Coût énergétique de la communication

La consommation d'énergie liée à la surcharge de communication des drones est calculée comme suit : Pour déterminer la consommation d'énergie pour la transmission (E_{tr}) et la réception (E_{rec}) des données, les formules suivantes sont appliquées [81] :

1. Pour la transmission de données :

$$E_{tr} = n \times E_{el} + n \times E_{fs} \times d^2$$

2. Pour la réception de données :

$$E_{rec} = n \times E_{el} + n \times E_{bf}$$

Ici, E_{el} représente l'énergie nécessaire pour initialiser les circuits de transmission et de réception (50 nJ/bit), E_{fs} est l'énergie pour le canal en espace libre (10 pJ/bit/m²), E_{bf} est l'énergie pour le beamforming (5 nJ/bit), et d^2 est la distance au carré entre les dispositifs. La variable n représente le nombre de sessions et leurs coûts de communication. Ainsi, nous comparons la consommation d'énergie du drone entre notre protocole LASeR, conçu pour un environnement de $100 \times 100\ m^2$, à celle des protocoles existants. La comparaison des bits envoyés et reçus par le drone est présentée dans le tableau 6.6. Concernant l'énergie, le protocole proposé utilise une énergie de 147,52 μJ, et la comparaison avec d'autres protocoles est illustrée dans la figure 6.11.

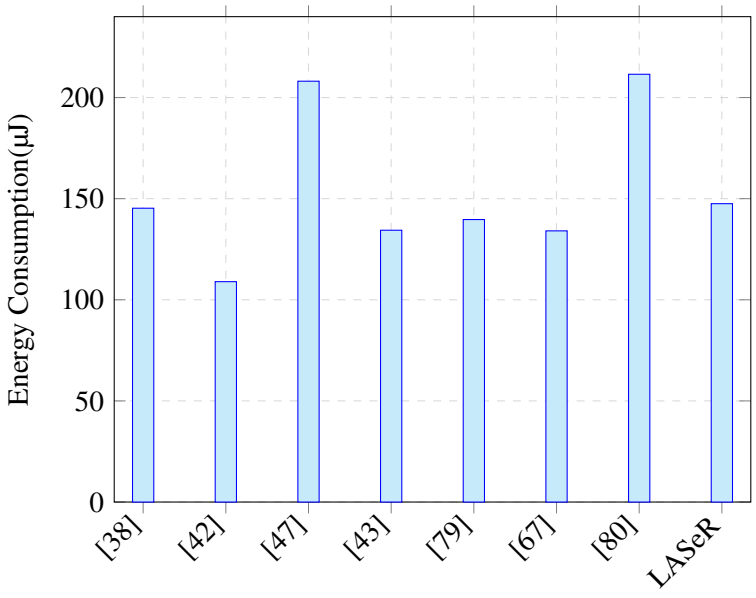


FIGURE 6.11 : Comparaison de la consommation d'énergie liée à la communication.

6.6 Conclusion

Dans ce chapitre nous avons proposé un protocole d'authentification léger et sécurisé pour l'IoD appelé LASeR, qui a la particularité d'offrir à l'utilisateur un accès basé sur la zone, permettant une surveillance continue d'une zone donnée. LASeR est basé sur des fonctions de hachage, l'opération XOR bit à bit, et bénéficie de la robustesse de la cryptographie sur courbes elliptiques. Pour garantir une double protection des communications, LASeR utilise une authentification à trois facteurs. En parallèle, il offre une authentification mutuelle, l'anonymat, forward/backward secrecy, ainsi que l'intraçabilité, le rendant plus robuste que les autres protocoles. L'analyse de sécurité montre que LASeR protège contre diverses attaques, telles que l'usurpation d'identité et le vol d'appareil mobile et de drone. De plus, les comparaisons de performance démontrent que LASeR requiert un coût computationnel et communicationnel inférieur à celui des autres protocoles connexes. Par conséquent, nous pouvons conclure que LASeR offre un meilleur compromis entre les coûts et les fonctionnalités de sécurité.

Bien que ce chapitre résolve le problème de la sécurité des communications entre l'utilisateur et le drone, un autre problème est apparu en septembre 2022, lorsque la Federal Aviation Administration (FAA) [81] et les autorités européennes [82] ont adopté une réglementation appelée RemoteID. Cette réglementation impose que tous les drones diffusent périodiquement leur identité pour répondre aux besoins des opérateurs d'infrastructures critiques. Cependant, les communautés de drones amateurs ont récemment déposé une plainte contre la FAA, invoquant des problèmes de confidentialité liés à la mise en œuvre obligatoire de RemoteID [83]. Pour clarifier cette situation, dans le chapitre suivant, nous proposons une solution d'authentification sans identification (authentification anonyme) entre les drones et les dispositifs de surveillance des zones.

Chapitre 7

2AS-DS : Schéma d'authentification anonyme basé sur fonction physique non clonable pour les essaims de drones

Dans ce chapitre, nous proposons un schéma d'authentification anonyme, visant à garantir la sécurité et l'efficacité des essaims de drones pour vérifier leur autorité d'accès, tout en respectant pleinement les réglementations de la FAA et en assurant l'anonymat et l'intraçabilité des messages de diffusion des drones.

7.1 Introduction

Dans ce chapitre, nous abordons les enjeux liés aux récentes réglementations de la FAA (Federal Aviation Administration) [81] et de l'Agence européenne de la sécurité aérienne (EASA) [82], qui ont introduit la réglementation appelée Identification à distance (RemoteID). Cette réglementation a été mise en place en raison des incidents signalés par les opérateurs d'infrastructures critiques, telles que les zones militaires et aéroportuaires, concernant l'accès non autorisé de drones amateurs, posant de sérieux problèmes de sécurité, de confidentialité et de sûreté. RemoteID exige que tous les drones diffusent périodiquement des messages indiquant leur identité. Depuis septembre 2022, la conformité à cette réglementation est devenue obligatoire. Bien que RemoteID soit conçu pour répondre aux besoins des opérateurs d'infrastructures critiques, il a provoqué des inquiétudes au sein de la communauté des amateurs de drones. Ces derniers ont récemment déposé une plainte contre la FAA [83], invoquant des violations de la vie privée liées à la mise en œuvre obligatoire de RemoteID. Comme cette réglementation est encore relativement nouvelle, les recherches sur les mécanismes d'authentification associés restent limitées. De plus, dans de nombreux domaines, plusieurs drones doivent souvent travailler ensemble pour accomplir des tâches complexes, dépassant les capacités

des drones individuels. L'idée des essaims de drones repose sur la coordination de plusieurs drones travaillant ensemble pour accomplir des tâches complexes, réduisant ainsi le temps total nécessaire à leur réalisation. De plus, les interactions collaboratives au sein d'un essaim de drones permettent un niveau élevé d'autonomie, minimisant le besoin d'intervention humaine dans le processus de contrôle. Toutefois, cela introduit une complexité dans l'application des nouvelles réglementations RemoteID, telles que le nombre de demandes d'authentification provenant des essaims de drones, qui doivent être transmises au serveur de contrôle, ainsi que les risques de sécurité importants [85].

- Premièrement, étant donné que la communication est réalisée via la technologie des balises Wi-Fi, qui est la technique de communication spécifiée par la norme RemoteID [**?**], l'adversaire peut intercepter des informations, lancer des attaques par rejeu ou effectuer des attaques d'usurpation d'identité. En conséquence, un drone malveillant pourrait être utilisé pour accéder à une zone, et il pourrait également tenter de pénétrer dans des endroits qui ne sont pas sur des itinéraires enregistrés.

- Deuxièmement, en raison de leur déploiement en plein air, les drones sont vulnérables aux attaques de clonage et à la capture de nœuds. Par conséquent, le processus d'authentification des drones doit être capable de protéger les infrastructures critiques contre les accès non autorisés.

- De plus, lorsqu'un essaim de drones communique simultanément avec la station de contrôle, une congestion significative des signaux peut survenir. En conséquence, la station de contrôle peut être confrontée à des dénis de service ou à des échecs de transmission, affectant négativement la qualité globale du service, y compris des échecs potentiels du processus d'authentification.

Plusieurs études ont examiné les questions de sécurité et de confidentialité liées à la mise en œuvre de la réglementation RemoteID pour les drones. Alkadi et Shoufan [69] ont proposé un protocole décentralisé pour la gestion du trafic, mais sans prendre en compte l'anonymat et l'authenticité des drones. D'autres travaux tel que Hashem et al. [86] ont proposé l'utilisation de la blockchain pour identifier les drones à distance, mais cette approche néglige l'anonymat et la confidentialité. La recherche sur l'authentification des essaims de drones est encore limitée. Ardin et al. [87] ont développé un système d'authentification pour les essaims, mais sans vérifier l'autorisation d'accès dans chaque zone aérienne. Abdel-Malek et al. [88] ont proposé une méthode d'authentification distribuée basée sur la délégation, mais celle-ci impose un lourd coût en termes de calcul lors de l'authentification de nombreux drones.

En dehors des progrès scientifiques, plusieurs systèmes commerciaux d'identification de drones émergent sur le marché. Ceux-ci incluent ScaleFyt de Thales [64], BLIP (Broadcast Location and Identification Platform) d'Unifly [65], et SIAM (Secure Airspace Integrated Management tool) de RelmaTech [66]. Ces solutions utilisent les fonctionnalités d'authentification et d'anonymat de la technologie cellulaire-LTE, qui ne constitue pas le mécanisme de communication actuellement proposé

par les réglementations RemoteID et le DRIP (Drone Remote Identification Protocol) du groupe de travail IETF.

Pour aborder les questions d'authentification et d'intégrité, le groupe de travail de l'IETF a proposé plusieurs RFC (Request for Comments), couvrant des sujets tels que les exigences pour les drones, la confidentialité et les enjeux de sécurité. Dans cette contribution, nous nous concentrons sur l'architecture de référence discutée dans la RFC 9434 [89]. Nous désignons par "drone" tout véhicule aérien sans pilote (UAV) et les dispositifs d'observation comme des récepteurs, tandis que le serveur de contrôle général est chargé de la vérification des drones. Ainsi, notre schéma est pleinement conforme à la terminologie, à l'architecture et aux exigences du DRIP. De plus, notre schéma, qui utilise des PUF (Fonctions Physiquement Inclaquables), vise à garantir la sécurité et l'efficacité des essaims de drones pour vérifier leur autorité d'accès (nommé 2AS-DS). 2AS-DS est conçu pour être entièrement conforme aux réglementations de la FAA tout en garantissant l'anonymat et l'intraçabilité des messages diffusés par les drones. Seule l'Autorité de Confiance (par exemple, le serveur de contrôle) peut révéler l'identité à long terme du drone.

7.2 Modèle de Système

L'architecture utilisée dans cette contribution est représentée dans la Figure 7.1. Nous supposons que le déploiement d'un essaim de drones dans une zone K nécessite de traverser plusieurs zones, et qu'à chaque passage, tous les drones de l'essaim doivent effectuer une authentification d'accès. En raison des ressources limitées des drones et du grand nombre de drones dans chaque essaim, il n'est pas optimal que chaque drone contacte directement le serveur de contrôle, car cela imposerait une charge excessive à ce dernier. Nous supposons que chaque drone envoie un message à l'observateur de zone, qui authentifie ensuite tous les drones simultanément. De plus, la récente réglementation RemoteID de la FAA permet de remplacer l'identité à long terme d'un drone par un pseudonyme. Afin de garantir une authentification anonyme, nous supposons que chaque paquet RemoteID envoyé par un drone contient à la fois une pseudo-identité individuelle et celle de l'essaim auquel il appartient.

Dans notre scénario, divers dispositifs d'observation (OD) reçoivent les paquets envoyés par les drones. Ces OD ont pour objectif de surveiller des infrastructures critiques telles que les aéroports et les installations militaires. Lorsqu'un drone entre dans une zone surveillée, le dispositif d'observation vérifie sa légitimité. Ce processus d'identification implique une communication entre le dispositif d'observation et un serveur de contrôle (Autorité de Confiance) afin de confirmer l'authenticité du drone.

Le serveur de contrôle est responsable de la régulation des activités des drones. Avant chaque vol, les opérateurs d'essaims de drones doivent enregistrer l'identité unique de l'essaim, celle de chaque drone, ainsi que des informations supplémentaires auprès du serveur de contrôle. En cas de détection d'un accès non autorisé, le dispositif d'observation peut signaler l'incident au serveur de contrôle,

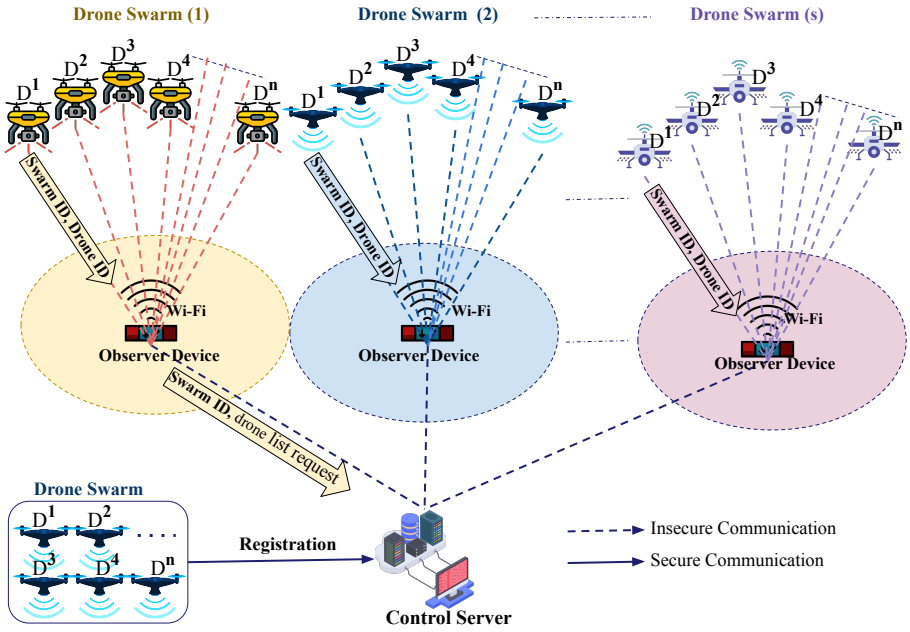


FIGURE 7.1 : Modèle de réseau (architecture) du protocole 2AS-DS proposé

qui est la seule entité capable de révéler l'identité à long terme des drones pour enquêter sur les incidents signalés. De plus, afin de garantir une sécurité renforcée à différents niveaux et de rendre notre proposition complète, notre mécanisme intègre un sous-schéma d'authentification et d'échange de clés entre le dispositif d'observation et le serveur de contrôle. Cela permet d'éviter les fausses alertes, tout en offrant à l'observateur la possibilité de récupérer en temps réel les paramètres de vérification de l'essaim de drones et d'informer le serveur de contrôle de tout accès non autorisé de manière confidentielle.

TABLE 7.1 : Notations pour le mécanisme d'authentification des essaims de drones.

Notation	Description
D_j, OD_k	Drone j^{th} et dispositif d'observateur k^{th}
CS	Serveur de contrôle de tous les drones et dispositifs d'observateur
$ID_{D_j}, ID_S, ID_{OD_k}$	Identités des drones j, du serveur S, et des observateurs k respectivement
ID_{CS}, (x_{CS}, Q_{CS})	Identité et clé publique/privée du serveur de contrôle
E	Courbe elliptique de référence
P	Point de base sur $E_p(a,b)$
a,b	Coefficients de la courbe elliptique
$k \cdot P$	Multiplication de point sur une courbe elliptique, où $k \in \mathbb{Z}_p^*$ et $P \in E_p(a,b)$
PUF	Fonction physique non clonable
C_S, Res_j	Défi de l'essaim et la réponse du drone D_j obtenue via la PUF
COD_k, Res_k	Défi et réponse du dispositif d'observateur OD_k obtenue via la PUF
LPID, LPRes	Liste de pseudo-identités et leurs pseudo-réponses correspondantes
N_{OD1}, N_{CS}, N_{OD2}	Nombres aléatoires
SK_{OD-CS}	Clé de session générée entre le serveur de contrôle et le dispositif d'observateur
$Gen(.), \tau_i$	Génération de clé avec extracteur flou, algorithme de reproduction, et paramètre
$h(.), \oplus, \|\|$	Fonction de hachage, opération XOR, et concaténation

7.3 Le protocole 2AS-DS proposé

Le protocole proposé, décrit dans cette section, se compose de trois phases : l'initialisation du système, la phase d'enregistrement de l'essaim de drones et l'authentification d'un essaim de drones.

7.3.1 Initialisation du système

Dans cette phase, le serveur de contrôle (CS) attribue une identité unique à chaque dispositif d'observation $ID_{OD_{k=1..m}}$ et sélectionne une valeur de défi spécifique à l'observateur C_{OD_k}. Par la suite, le dispositif d'observation calcule $R_k = PUF(C_{OD_k})$, $(Res_k, \tau_k) = Gen(R_k)$, et stocke $\{ID_{OD_k}, \tau_k\}$. En parallèle, le serveur de contrôle stocke de manière sécurisée $\{ID_{OD_k}, C_{OD_k}, Res_k\}$ dans sa base de données.

7.3.2 Phase d'enregistrement de l'essaim de drones

L'administrateur d'un essaim de drones est chargé d'enregistrer l'essaim auprès du serveur de contrôle (SC) géré par l'autorité aéronautique. Pour ce faire, l'administrateur génère aléatoirement un défi unique pour l'essaim (Cs). Ensuite, chaque drone génère une pseudo-identité et utilise le PUF pour

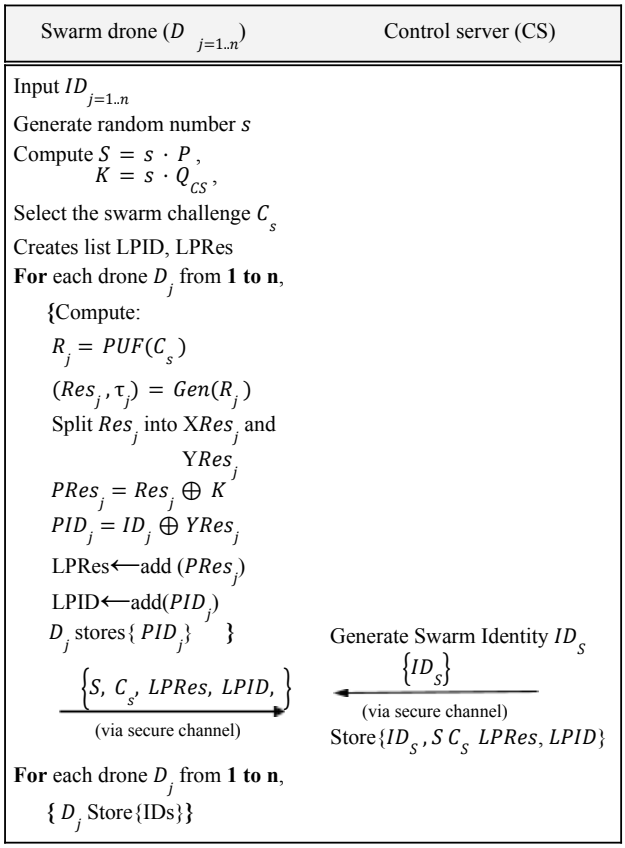


FIGURE 7.2 : Phase d'enregistrement de l'essaim de drones.

générer la réponse à Cs. Enfin, l'administrateur crée une liste de pseudo-identités accompagnées de leurs réponses respectives et du plan de vol de l'essaim de drones, qui est partagé avec le CS, permettant ainsi une identification unique de l'essaim. Cette phase est résumée dans la figure 7.2.

7.3.3 Authentification d'un essaim de drones

La récente réglementation RemoteID de la FAA permet de remplacer l'identité à long terme d'un drone par un pseudonyme. Afin de garantir une authentification anonyme, nous supposons que les essaims de drones diffusent en broadcast un paquet RemoteID. Chaque paquet envoyé par un drone contient à la fois une pseudo-identité individuelle et celle de l'essaim auquel il appartient. Lors de la réception du premier paquet émis par un drone, le processus d'authentification est déclenché par l'observateur afin de vérifier la légitimité des essaims de drones. Le diagramme de séquence utilisé pour la vérification de la légitimité de l'essaim de drones dans notre protocole 2AS-DS est illustré dans la Figure 7.3 et expliqué étape par étape dans la suite.

Étape $U1$ (*Demande d'authentification de l'essaim de drones*)
Chaque drone de l'essaim envoie sa pseudo-identité, l'identité de l'essaim et un nonce (PID_{D_j},ID_S,

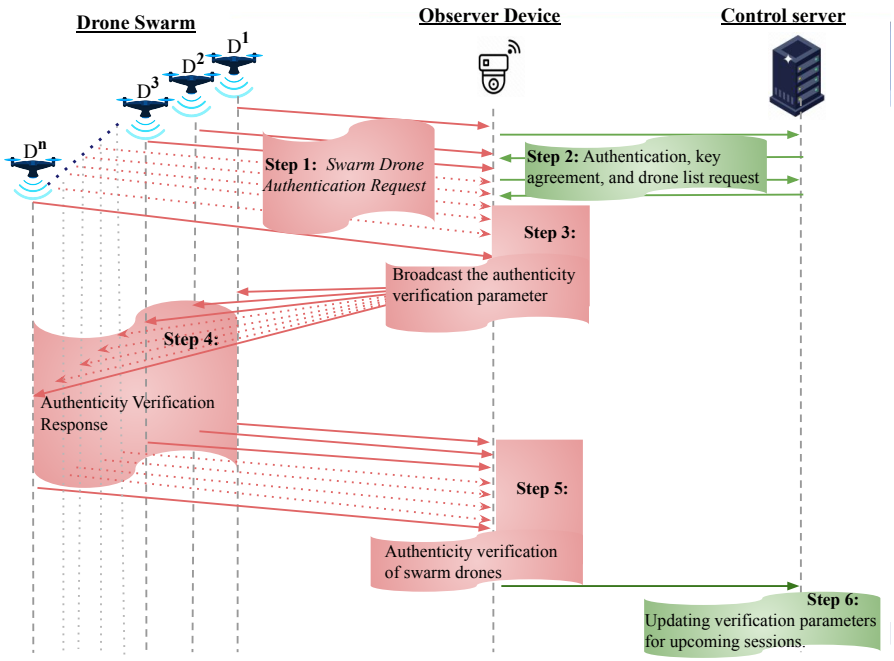


FIGURE 7.3 : Diagramme de séquence pour l'authentification d'un essaim de drones.

N_{D_j}).

Étape $U2$ (*Authentification, échange de clés et demande de liste des drones*)
L'observateur déclenche son opération dès réception du premier message de l'essaim de drones (identité de l'essaim). Cette Étape est résumée dans la figure 7.4.
Ce processus implique quatre messages :

- Message 1 : L'observateur envoie son identité et un nonce (ID_{OD_k}, N_{OD_1}).
- Message 2 : Le serveur de contrôle vérifie l'existence de ID_{OD_k} dans sa base de données ; si trouvé, il récupère Res_k, C_{OD_k} et génère un nonce (N_{CS}). Il calcule ensuite $V_{CS-OD} = h(C_{OD_k}||Res_k||N_{OD_1}||N_{CS})$, et envoie $\{ C_{OD_k}, N_{CS}, V_{CS-OD}\}$ à l'observateur.
- Message 3 : À la réception de $\{ C_{OD_k}, N_{CS}, V_{CS-OD}\}$, l'observateur calcule $R_k^{'} = PUF(C_{OD_k})$, $(Res_k^{'}, \tau_k) = Gen(R_k^{'})$, $V_{CS-OD}^{'} = h(C_{OD_k}||Res_k^{'}||N_{OD_1}||N_{CS})$, et vérifie si la condition $V_{CS-OD}^{'} \stackrel{?}{=} V_{CS-OD}$ est satisfaite. En cas de vérification réussie, l'observateur calcule $SK_{OD-CS} = h(Res_k^{'}||N_{OD_1}||N_{CS})$, génère une nouvelle valeur de défi pour l'observateur $C_{OD_k}^{new}$, puis calcule $R_k^{new} = PUF(C_{OD_k}^{new})$, et $(Res_k^{new}, \tau_k) = Gen(R_k^{new})$. Il envoie ensuite l'identité de l'essaim pour demander la liste des pseudo-identités des drones appartenant à cette identité d'essaim et leurs paramètres de vérification $\{ID_S, N_{CS}, C_{OD_k}^{new}, Res_k^{new}\}$, chiffrés avec SK_{OD-CS}.
- Message 4 : À la réception du message de l'observateur, le serveur de contrôle calcule $SK_{OD-CS} = h(Res_k||N_{OD_1}||N_{CS})$, puis déchiffre le Message 3. Il vérifie ensuite si le nonce envoyé dans ce

message correspond à celui envoyé dans le Message 2 ; si c'est le cas, il met à jour les valeurs de vérification pour l'observateur pour la prochaine session. Il recherche ensuite dans sa base de données la liste des pseudo-identités des drones et leurs paramètres de vérification, et envoie $\{C_s, LPID, LRes\}$ chiffrés avec SK_{OD-CS} au dispositif de l'observateur.

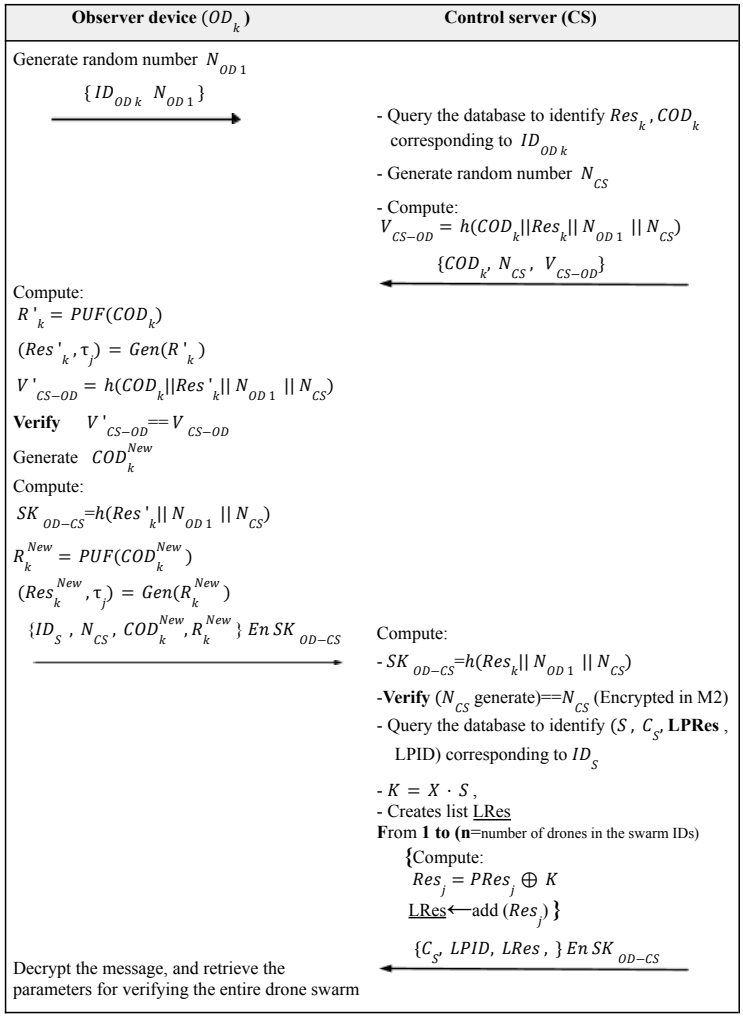


FIGURE 7.4 : Authentification, accord de clé et demande de liste des drones.

Étape $U3$ (*Diffusion du paramètre de vérification d'authenticité à tous les drones de l'essaim*)
Cette étape commence à la réception du Message 4 de l'Étape 2, qui inclut les défis des drones de l'essaim (C_s), la liste des pseudo-identités des drones $(LPID_{D_j})$, et les paramètres de vérification pour chaque drone de l'essaim $(LRes)$. Si ces conditions sont remplies, l'observateur génère un nonce N_{OD_2} et diffuse le défi de l'essaim et N_{OD_2} à l'ensemble de l'essaim de drones.

Étape $U4$ (*Réponse de vérification d'authenticité*)
À la réception du message (C_s), N_{OD_2}, chaque drone (D_j) calcule :
$R_j = PUF(C_s)$, $(Res_j, \tau_j) = Gen(R_j)$,

$V_{D_j} = h(PID_{D_j}||N_{D_j}||Res_{D_j}||N_{OD_2})$,
C_s^{new}=$h(C_s||N_{OD_2})$,
$R_j^{new} = PUF(C_s^{new})$, $(Res_j^{new}, \tau_j) = Gen(R_j^{new})$,
$PRes_j^{new}$=$Res_j^{new} \oplus XRes_j$,
PID_j^{new}= $PID_j \oplus YRes_j \oplus YRes_j^{new}$, et envoie { PID_j, $PRes_j^{new}$, V_{D_j}} à l'observateur.

Étape $U5$ (*Vérification d'authenticité de l'essaim de drones*)

À la réception du message de réponse de vérification d'authenticité $PID_j, PRes^{new}\, j, VD_j$, l'observateur lance l'Algorithme 1 (Vérification d'authenticité de l'essaim). Cet algorithme prend en entrée la liste des pseudo-identités, la liste des résultats de challenge de l'essaim reçus à l'étape 2, ainsi que la clé générée avec le serveur de contrôle. À la fin de l'algorithme, le résultat est l'authentification des différents drones de l'essaim et l'émission d'alertes en cas de présence d'un drone non-membre. Enfin, les différents paramètres de vérification sont mis à jour pour la prochaine session ou changement de zone.

Étape $U6$ (*Mise à jour des paramètres de vérification pour les prochaines sessions*)

Le serveur de contrôle, à la réception du message de l'appareil observateur, le déchiffrera en utilisant la clé de session générée à l'étape 2 et récupérera le message UpdateListe. Ensuite, il mettra à jour la liste des drones dans cet essaim et enregistrera le nouveau challenge de l'essaim pour la prochaine session ou changement de zone.

Algorithm 1: Vérification d'Authenticité de l'Essaim de Drones

Input: $LPID$, $LRes$, SK_{OD-CS}

Output: UpdateListes, unauthenticated drones (Alert)

for $j = 1$ *to* n **do**
  wait for `message received` (PID_j, $PRes_j$, V_{D_j})
  Query $LRes$ to find Res_j^* corresponding to PID_j
  $V_{D_j}^* = h(PID_{D_j}||N_{D_j}||Res_{D_j}^*||N_{OD_2})$
  IF (V_{D_j}==$V_{D_j}^*$)
  Authentication is successful of D_j
  Res_j^{new}=$PRes_j^{new} \oplus XRes_j$
  Update the parameter Res_j in the LRes list.
  PID_j^{new}=$PID_j \oplus YRes_j \oplus YRes_j^{new}$
  Update the parameter PID_j in the LPID list.
  Else
  Alert : D_j is not a member of this swarm.
end
C_s^{new}=$h(C_s||N_{OD_2})$
UpdateListes = {$LPID$, $LRes$, C_s^{new}} encrypted by SK_{OD-CS}

7.4 Analyse de sécurité

Nous évaluons la sécurité de 2AS-DS contre divers attaques connues, comme détaillé ci-dessous.

7.4.1 Anonymat des drones

2AS-DS garantit l'anonymat des drones et reste pleinement compatible avec les règles RemoteID. Dans la phase d'authentification,Chaque message envoyé par D_j à l'instant t contient une pseudo-identité, associée à l'identité à long terme, mais qui change a chaque session. Le renouvellement continu du pseudonyme garantit l'unlinkabilité des messages. Une attaque interceptant deux messages ne peut pas déterminer s'ils proviennent du même drone ou de drones différents.

7.4.2 Non-traçabilité

Dans 2AS-DS, après une authentification réussie, le drone met à jour ses paramètres de vérification. La pseudo-identité varie à chaque session, empêchant la réutilisation dans des sessions sécurisées successives. Cela garantit la non-traçabilité du drone et la sécurité des paramètres d'authentification.

7.4.3 Résilience aux attaques de masquerade et de type homme du milieu

Un adversaire pourrait accéder aux secrets stockés dans le drone et le dispositif d'observation. Par conséquent, les entités ne devraient pas stocker de paramètres secrets dans leur mémoire. Dans 2AS-DS, aucun paramètre secret n'est stocké dans la mémoire des entités. Puisque PUF est propre au matériel, il reste inaccessible aux adversaires. Dans notre schéma 2AS-DS, chaque drone et dispositif d'observation doit posséder un PUF, empêchant les attaques de masquerade. La transmission chiffrée de Res_j et Res_k garantit leur sécurité entre les drones et les dispositifs d'observation, ainsi qu'entre les dispositifs d'observation et le serveur de contrôle, aidant à prévenir les attaques de type homme du milieu.

7.4.4 Résilience aux attaques de rejeu et de clonage

À chaque session d'authentification, une nouvelle pseudo-identité, un nouveau nonce, ainsi qu'un nouveau défi de groupe (C_s) et une réponse sont générés. Par conséquent, les messages ne peuvent pas être décryptés en utilisant d'anciens paramètres, ce qui empêche les attaques de rejeu. Si un adversaire capture un dispositif, ces paramètres restent inaccessibles car ils ne sont pas stockés dans la mémoire du drone ou du dispositif d'observation. De plus, comme les PUFs ne peuvent pas être clonés, un adversaire ne peut pas cloner le drone ou le dispositif d'observation. Ainsi, 2AS-DS est sécurisé contre les attaques de rejeu et de clonage.

7.5 Analyse comparative des mesures de sécurité

Dans cette section, nous démontrons que notre schéma proposé est mieux sécurisé que les schémas récemment proposés. Le tableau 7.2 présente une comparaison des propriétés de sécurité de 2AS-DS avec celles d'autres schémas. Comme indiqué dans le tableau 7.2, Lounis et al. [91] et Khan et al. [94] ne peuvent pas garantir l'anonymat des drones et la non-traçabilité, car l'identité du drone reste toujours la même. De plus, Gope et al. [93], Alladi et al. [92], et Karmakar et al. [95] n'utilisent pas d'extracteurs flous pour réduire les erreurs dans les PUFs, et les véritables CRP (paires défi-réponse) sont enregistre dans la base de données CS, ce qui les rend vulnérables à l'exposition par un adversaire. En outre, Alladi et al. [92] est vulnérable aux attaques de déni de service (DoS). En revanche, 2AS-DS est capable de satisfaire chacune des exigences de sécurité.

TABLE 7.2 : Évaluation des performances en fonction des exigences de sécurité

Properties	**2AS-DS**	**Lounis et al.** [91]	**Alladi et al.** [92]	**Gope et al.** [93]	**Khan et al.** [94]	**Karmakar et al.** [95]
Anonymat	✓	✗	✓	✓	✗	✓
Intraçabilité	✓	✗	✓	✓	✗	✓
Scalabilité	✓	✗	✓	✓	✓	✓
Résistance aux DOS	✓	✓	✗	✓	✓	✓
CRP stocké de manière chiffrée	✓	✓	✗	✗	N/A	✗
FE pour réduire les erreurs de PUF	✓	✓	✗	✗	N/A	✓

7.6 Conclusion

Dans ce chapitre, nous avons abordé les enjeux cruciaux de sécurité et de confidentialité liés à la mise en œuvre obligatoire du RemoteID pour les drones, imposée par la FAA en septembre 2022. Nous avons également souligné la complexité supplémentaire qu'implique l'application de cette réglementation aux essaims de drones.

Pour répondre à ces défis, nous avons proposé un schéma d'authentification anonyme basé sur des PUF (Physical Unclonable Functions). Ce schéma assure l'anonymat et la non-traçabilité des messages émis par les drones, tout en réservant au serveur de contrôle (l'autorité de confiance) la capacité exclusive de révéler l'identité à long terme d'un drone. Dans le cadre du modèle 2AS-DS, pour vérifier efficacement l'autorisation d'accès à une zone spécifique par des essaims de drones, l'appareil d'observation collecte les paramètres de vérification, remplaçant ainsi le serveur de contrôle dans ce rôle. Grâce à ce processus

de vérification par essaim, l'appareil d'observation peut valider les informations d'authentification des drones membres de l'essaim, et envoyer un rapport à le serveur de contrôle en cas de tentative d'accès non autorisé. Afin de garantir une sécurité renforcée à différents niveaux et de rendre notre proposition plus complète, nous avons également intégré un sous-schéma d'authentification entre l'appareil d'observation et le serveur de contrôle, dans le but de prévenir les fausses alertes. Le modèle proposé, 2AS-DS, est résistant à diverses attaques connues, telles que les interceptions, les rejets et le clonage.

Toutefois, étant donné que notre évaluation se limite pour l'instant à une analyse de sécurité informelle, des analyses plus approfondies seront entreprises dans nos travaux futurs. Nous prévoyons notamment de réaliser des évaluations complémentaires, de comparer nos résultats avec d'autres recherches, et de mener une analyse de sécurité formelle plus exhaustive. De plus, nous inclurons des critères supplémentaires tels que le coût de calcul, la consommation d'énergie et les coûts de communication.

Chapitre 8

Conclusion

Ce chapitre présente tout d'abord une synthèse des travaux réalisés dans cette thèse. Ensuite, les principales limites des recherches effectuées sont exposées. Enfin, des orientations de recherches futures sont proposées, ouvrant des perspectives pour des travaux ultérieurs.

8.1 Synthèse des travaux

Cette thèse a pour objectif principal de développer des mécanismes robustes visant à améliorer la sécurité dans le contexte de l'Internet des objets, et plus particulièrement dans le domaine des villes intelligentes. Dans ce domaine, le transport intelligent représente l'un des principaux défis, pouvant entraîner divers problèmes tels que la congestion routière, la complexité de la coordination des systèmes interconnectés, ainsi que l'optimisation des coûts d'implémentation. Ces systèmes s'appuient principalement sur l'accès à des données en temps réel, où la collecte et la transmission des données soulèvent des défis de sécurité majeurs, nécessitant des solutions adaptées.

Cette thèse porte sur trois contributions principales pour améliorer la sécurité. La première contribution porte sur l'authentification de l'utilisateur dans l'Internet des véhicules, ce qui nous a conduits à proposer deux améliorations détaillées ci-dessous. Dans le cadre de l'automatisation du système global, la deuxième contribution a intégré des drones pour fournir des informations sur des zones spécifiques, automatisant ainsi divers services tels que l'inspection routière, la police de la circulation, ou les équipes de secours. Enfin, la troisième contribution porte sur l'automatisation du déploiement des drones de manière autonome. Ce processus a soulevé des défis en matière d'authentification pour l'accès à des zones spécifiques, ce qui nous a conduits à proposer un mécanisme d'authentification d'essaim de drones permettant aux autorités publiques d'identifier les drones en temps réel, tout en garantissant la confidentialité des utilisateurs.

Les principales contributions de cette thèse peuvent être résumées de la manière suivante :

1. Conception d'un protocole d'authentification et d'échange de clés pour l'Internet des véhicules, avec deux améliorations majeures :

    - La première amélioration se base sur le protocole proposé par Q. Jiang et al. [25], qui implique trois entités (utilisateur, centre de données et véhicule). Notre cryptanalyse a révélé une vulnérabilité majeure concernant la confidentialité des messages, permettant à un attaquant de déduire la clé de session partagée entre le véhicule et les autres entités. Bien que notre amilioration augmente légèrement le temps de calcul global, elle corrige cette vulnérabilité tout en maintenant un temps de traitement faible pour les véhicules, rendant ainsi le protocole applicable à l'IoV.
    - La deuxième amélioration concerne le protocole UAP-BCIoT [48], adapté à la collecte de big data et aux systèmes nécessitant des réponses rapides. Le protocole élimine la communication directe entre l'utilisateur et le centre de données, dans le but de réduire les coûts et de diminuer les risques liés à ces communications directes. Notre cryptanalyse a permis de corriger une faille affectant l'authentification entre les utilisateurs et les dispositifs IoT déployés, garantissant ainsi un accès sécurisé aux données.

2. Proposition d'un protocole d'authentification et d'échange de clés nommé LASeR, destiné aux interactions entre les utilisateurs (tels que pompiers ou policiers) et les drones déployés sur des zones spécifiques. LASeR se distingue par son accès basé sur des zones géographiques, assurant une surveillance continue tout en offrant un compromis optimal entre coût et sécurité. Une analyse de sécurité formelle, réalisée à l'aide d'outils tels que Scyther, AVISPA et Tamarin, ainsi qu'une analyse informelle, ont démontré sa robustesse face à douze attaques connues.

3. Conception d'un protocole d'authentification pour les essaims de drones, nommé 2AS-DS, conforme aux réglementations de la FAA tout en garantissant l'anonymat et l'intraçabilité des communications. Une analyse de sécurité préliminaire montre que 2AS-DS résiste aux attaques telles que l'interception, le rejeu et le clonage, tout en réduisant la charge sur la station de confiance, ce qui améliore les performances en termes d'efficacité et de rapidité.

8.2 Limitations des travaux réalisés

En dépit du fait que les contributions évoquées précédemment soient significatives, les travaux réalisés dans le cadre de cette thèse présentent les limitations suivantes :

1. **Points amiliorer en matière de sécurité et perfermance :**

    - Dans notre amélioration du protocole Jiang et al.[25] proposé dans le chapitre 4, nous avons corrigé la vulnérabilité existante tout en maintenant un temps de traitement relativement faible pour le véhicule, ce qui garantit son applicabilité à l'Internet des véhicules.

Cependant, il pourrait ne pas être idéal pour les environnements Big Data, car l'échange de 4 messages, avec des tailles de messages pouvant atteindre 3648 bits, peut augmenter la charge sur le réseau, ce qui pourrait entraîner des ralentissements dans les systèmes.

- Dans la deuxième amélioration du protocole proposée dans le chapitre 5, nous avons renforcé la sécurité tout en garantissant l'efficacité du protocole UAP-BCIoT [48]. Ainsi, celui-ci nécessite seulement 2496 bits (soit 3 messages), et le centre de données n'intervient qu'une seule fois (avec une seule réception et un seul envoi). Cependant, ce protocole génère uniquement 2 clés de session : une entre les objets IoT déployés dans le véhicule et l'utilisateur, et l'autre avec le centre de données. Il n'y a donc pas de clé directe entre l'utilisateur et le serveur.
- Dans notre protocole LASeR présenté au chapitre 6, le serveur de la station au sol (GSS) montre une résistance aux attaques par déni de service (DoS) grâce à ses opérations de vérification peu coûteuses. Toutefois, il doit les réaliser à chaque demande d'authentification et ne peut pas rejeter les tentatives d'authentification malveillantes par une simple recherche. Il est donc recommandé d'intégrer un système de détection des attaques DoS au niveau du GSS.
- Les solutions proposées utilisent la carte à puce comme élément central pour l'authentification sécurisée. En ce qui concerne les attaques par canaux auxiliaires, les données sensibles sont stockées de manière masquée, réduisant ainsi la probabilité d'exploitation directe. Cependant, le canal de communication entre l'application mobile et l'applet sur la carte à puce n'est pas suffisamment protégé. Bien que l'écoute de ce canal nécessite des moyens sophistiqués, la complexité de l'attaque ne doit pas être un critère suffisant pour négliger cette vulnérabilité.

  Il est donc essentiel que le système de la carte mette en place des mécanismes adaptés pour garantir un canal de communication sécurisé avec l'application mobile.

2. **Limitations en termes d'évaluation de performance :** L'objectif principal de l'analyse de performance des protocoles d'authentification est de prendre en compte la complexité des fonctions cryptographiques utilisées, ainsi que le nombre de fois qu'elles sont appliquées. Bien qu'il soit recommandé, dans le cadre d'une étude comparative, de se baser sur des paramètres de comparaison identiques (chaque fonction cryptographique ayant son propre temps d'exécution), cela ne doit pas justifier l'absence d'une évaluation du temps d'exécution de chaque fonction. Cette évaluation est essentielle pour garantir la faisabilité et l'efficacité des solutions dans des conditions d'utilisation réelles.

8.3 Orientations de recherches futures

Nous concluons cette thèse en identifiant des pistes de recherche susceptibles d'éclairer la voie pour de futurs travaux.

- L'intégration des drones pour automatiser certains services, tels que l'inspection routière, la surveillance du trafic et les interventions d'urgence, représente une innovation significative. Néanmoins, la sécurisation des communications entre essaims de drones reste un défi majeur. La conception de mécanismes garantissant la sécurité et la confidentialité de ces interactions constitue une piste de recherche intéressante.
- Dans cette thèse, nous avons proposé deux améliorations pour deux protocoles d'authentification destinés à l'Internet des véhicules (IoV). De plus, nous avons proposé deux nouveaux protocoles pour l'Internet des drones (IoD), visant à garantir un accès sécurisé aux drones et à contrôler l'accès aux zones de vol. Toutefois, il reste essentiel de développer un protocole d'authentification capable de garantir une communication sécurisée entre les drones et les véhicules, en assurant l'authenticité et l'intégrité des données transmises. Ces avancées contribueraient ainsi à renforcer la sécurité globale des réseaux IoV/IoD.

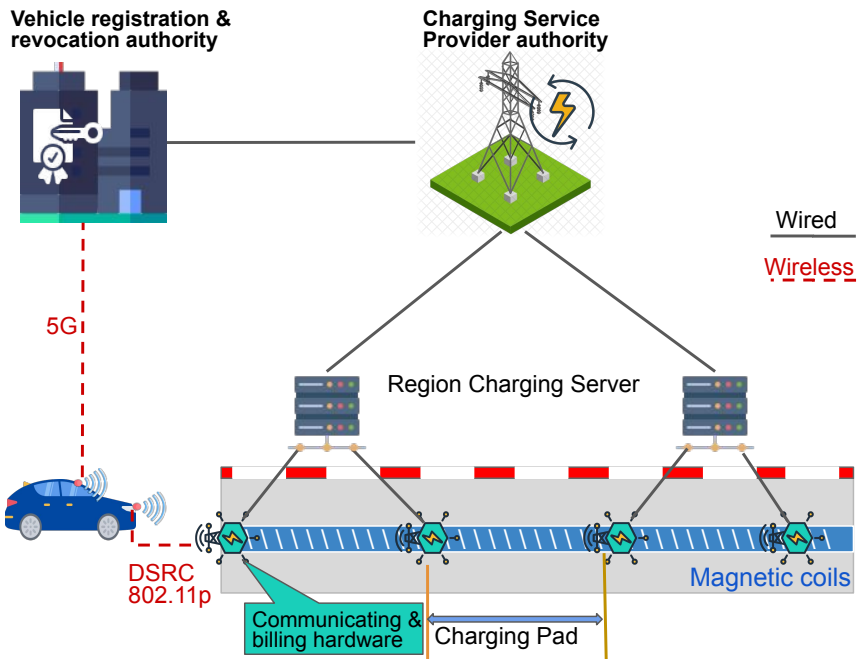


FIGURE 8.1 : Modèle système de recharge sans fil pour véhicules électriques dans un segment routier

- L'émergence des véhicules électriques (VE) a incité les innovateurs à rechercher une infrastructure de recharge flexible. DWPT (Dynamic Wireless Power Transfer) [96] [97] est une technologie novatrice qui permet de recharger les VE pendant qu'ils roulent, grâce à des plaques

de recharge (*charging pads*) installées sur la route, comme illustré à la figure 8.1. En général, les véhicules sont équipés d'un composant dédié à la recharge et à la communication. En ce qui concerne les plaques de recharge, elles se composent de deux éléments : une bobine inductive pour la transmission d'énergie et d'un dispositif de communication. Dans ce cadre, le réseau câblé est considéré comme sécurisé. Cependant, la communication entre le véhicule et les plaques de recharge soulève de nouveaux défis, tels que des attaques spécifiques au système DWPT, comme le *free-riding* et le *double-spending*. De plus, des défis subsistent concernant la mise en œuvre de mécanismes de sécurité robustes, notamment en ce qui concerne l'authentification des utilisateurs, sans révéler leur véritable identité au cours de plusieurs cycles de recharge dynamique. Ce sujet représente une piste de recherche prometteuse.

Bibliographie

[1] Fadele Ayotunde Alaba, Mazliza Othman, Ibrahim Abaker Targio Hashem, and Faiz Alotaibi. Internet of things security : A survey. Journal of Network and Computer Applications, 88 :10–28, 2017. doi : 10.1016/j.jnca.2017.04.002.

[2] KOBLITZ, Neal. Elliptic curve cryptosystems. Mathematics of computation, 1987, vol. 48, no 177, p. 203-209.

[3] MILLER, Victor S. Use of elliptic curves in cryptography. In : Conference on the theory and application of cryptographic techniques. Berlin, Heidelberg : Springer Berlin Heidelberg, 1985. p. 417-426.

[4] PAPPU, Ravikanth, RECHT, Ben, TAYLOR, Jason, et al. Physical one-way functions. Science, 2002, vol. 297, no 5589, p. 2026-2030.

[5] GASSEND, Blaise, CLARKE, Dwaine, VAN DIJK, Marten, et al. Silicon physical random functions. In : Proceedings of the 9th ACM Conference on Computer and Communications Security. 2002. p. 148-160.

[6] MAES, Roel et VERBAUWHEDE, Ingrid. Physically unclonable functions : A study on the state of the art and future research directions. Towards Hardware-Intrinsic Security : Foundations and Practice, 2010, p. 3-37.

[7] ROEL, M. A. E. S. Physically unclonable functions : Constructions, properties and applications. Katholieke Universiteit Leuven, Belgium, 2012, p. 148-160.

[8] WACHSMANN, Christian et SADEGHI, Ahmad-Reza. Physically unclonable functions (PUFs) : Applications, models, and future directions. Morgan & Claypool Publishers, 2014.

[9] MAES, Roel et VERBAUWHEDE, Ingrid. Physically unclonable functions : A study on the state of the art and future research directions. Towards Hardware-Intrinsic Security : Foundations and Practice, 2010, p. 3-37.

[10] HERDER, Charles, YU, Meng-Day, KOUSHANFAR, Farinaz, et al. Physical unclonable functions and applications : A tutorial. Proceedings of the IEEE, 2014, vol. 102, no 8, p. 1126-1141.

[11] SUH, G. Edward et DEVADAS, Srinivas. Physical unclonable functions for device authentication and secret key generation. In : Proceedings of the 44th annual design automation conference. 2007. p. 9-14.

[12] DANGER, J.-L., GUILLEY, Sylvain, NGUYEN, Philippe, et al. PUFs : Standardization and evaluation. In : 2016 Mobile System Technologies Workshop (MST). IEEE, 2016. p. 12-18.

[13] YANG, Wencheng, WANG, Song, YIN, Xuefei, et al. A review on security issues and solutions of the internet of drones. IEEE Open Journal of the Computer Society, 2022, vol. 3, p. 96-110.

[14] S. Jiang, X. Zhu, and L. Wang, "An efficient anonymous batch authentication scheme based on hmac for vanets," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 8, pp. 2193–2204, 2016

[15] M. Wazid, A. K. Das, N. Kumar, V. Odelu, A. G. Reddy, K. Park, and Y. Park, "Design of lightweight authentication and key agreement protocol for vehicular ad hoc networks," IEEE Access, vol. 5, pp. 14 966–14 980, 2017

[16] B. Ying and A. Nayak, "Anonymous and lightweight authentication for secure vehicular networks," IEEE Transactions on Vehicular Technology, vol. 66, no. 12, pp. 10 626–10 636, 2017.

[17] P. Mohit, R. Amin, and G. Biswas, "Design of authentication protocol for wireless sensor network-based smart vehicular system," Vehicular Communications, vol. 9, pp. 64–71, 2017.

[18] Y. Zhou, S. Liu, M. Xiao, S. Deng, and X. Wang, "An efficient v2i authentication scheme for vanets," Mobile Information Systems, vol. 2018.

[19] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for internet of vehicles," Ieee Access, vol. 7, pp. 12 047–12 057, 2019.

[20] C. Wang, L. Xiao, J. Shen, and R. Huang, "Neighborhood trustworthiness-based vehicle-to-vehicle authentication scheme for vehicular ad hoc networks," Concurrency and Computation : Practice and Experience, vol. 31, no. 21, p. e4643, 2019.

[21] M. Ma, D. He, H. Wang, N. Kumar, and K.-K. R. Choo, "An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8065–8075, 2019.

[22] H. Vasudev, V. Deshpande, D. Das, and S. K. Das, "A lightweight mutual authentication protocol for v2v communication in internet of vehicles," IEEE Transactions on Vehicular Technology, vol. 69, no. 6, pp. 6709– 6717, 2020.

[23] H. Vasudev, D. Das, and A. V. Vasilakos, "Secure message propagation protocols for iovs communication components," Computers & Electrical Engineering, vol. 82, p. 106555, 2020.

[24] W. Othman, M. Fuyou, K. Xue, and A. Hawbani, "Physically secure lightweight and privacy-preserving message authentication protocol for vanet in smart city," IEEE Transactions on Vehicular Technology, vol. 70, no. 12, pp. 12 902–12 917, 2021.

[25] Q. Jiang, X. Zhang, N. Zhang, Y. Tian, X. Ma, and J. Ma, "Threefactor authentication protocol using physical unclonable function for iov," Computer Communications, vol. 173, pp. 45–55, 2021.

[26] J. Wang, L. Wu, H. Wang, K.-K. R. Choo, L. Wang, and D. He, "A secure and efficient multi-server authentication and key agreement protocol for internet of vehicles," IEEE Internet of Things Journal, 2022.

[27] I. Ahmim, N. Ghoualmi-Zine, A. Ahmim, and M. Ahmim, "Security analysis on "three-factor authentication protocol using physical unclonable function for iov"," International Journal of Information Security, pp. 1–8, 2022.

[28] P. Gope and B. Sikdar, "Privacy-aware authenticated key agreement scheme for secure smart grid communication," IEEE Transactions on Smart Grid, vol. 10, no. 4, pp. 3953–3962, 2018.

[29] M. Ahmim, A. Ahmim, M. A. Ferrag, N. Ghoualmi-Zine, and L. Maglaras, "Esike : An efficient and secure internet key exchange protocol," Wireless Personal Communications, pp. 1–16, 2022.

[30] A. Derhab, O. Cheikhrouhou, A. Allouch, A. Koubaa, B. Qureshi, M. A. Ferrag, L. Maglaras, and F. A. Khan, "Internet of drones security : Taxonomies, open issues, and future directions," Vehicular Communications, p. 100552, 2022.

[31] M. Wazid, A. K. Das, and J.-H. Lee, "Authentication protocols for the internet of drones : taxonomy, analysis and future directions," Journal of Ambient Intelligence and Humanized Computing, pp. 1–10, 2018.

[32] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Tcalas : Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment," IEEE Transactions on Vehicular Technology, vol. 68, no. 7, pp. 6903–6916, 2019.

[33] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 3572–3584, 2019.

[34] Y. Zhang, D. He, L. Li, and B. Chen, "A lightweight authentication and key agreement scheme for internet of drones," Computer Communications, vol. 154, pp. 455–464, 2020.

[35] S. Yu, A. K. Das, Y. Park, and P. Lorenz, "Slap-iod : Secure and lightweight authentication protocol using physical unclonable functions for internet of drones in smart city environments," IEEE Transactions on Vehicular Technology, vol. 71, no. 10, pp. 10 374– 10 388, 2022.

[36] S. Hussain, K. Mahmood, M. K. Khan, C.-M. Chen, B. A. Alzahrani, and S. A. Chaudhry, "Designing secure and lightweight user access to drone for smart city surveillance," Computer Standards & Interfaces, vol. 80, p. 103566, 2022.

[37] S. Hussain, M. Farooq, B. A. Alzahrani, A. Albeshri, K. Alsubhi, and S. A. Chaudhry, "An efficient and reliable user access protocol for internet of drones," IEEE Access, pp. 1–1, 2023.

[38] M. W. Akram, A. K. Bashir, S. Shamshad, M. A. Saleem, A. A. AlZubi, S. A. Chaudhry, B. A. Alzahrani, and Y. B. Zikria, "A secure and lightweight drones-access protocol for smart city surveillance," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 10, pp. 19 634–19 643, 2022.

[39] N. Zhang, Q. Jiang, L. Li, X. Ma, and J. Ma, "An efficient threefactor remote user authentication protocol based on bpv-fourq for internet of drones," Peer-to-Peer Networking and Applications, vol. 14, pp. 3319–3332, 2021.

[40] M. Tanveer, A. H. Zahid, M. Ahmad, A. Baz, and H. Alhakami, "Lake-iod : Lightweight authenticated key exchange protocol for the internet of drone environment," IEEE Access, vol. 8, pp. 155 645–155 659, 2020.

[41] S. Liu and C.-M. Chen, "Comments on "a secure and lightweight drones-access protocol for smart city surveillance"," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 12, pp. 25 054–25 058, 2022.

[42] M. Tanveer, A. U. Khan, N. Kumar, and M. M. Hassan, "Ramp-iod : A robust authenticated key management protocol for the internet of drones," IEEE Internet of Things Journal, vol. 9, no. 2, pp. 1339– 1353, 2022.

[43] BERINI, Aymen Dia Eddine, FERRAG, Mohamed Amine, FAROU, Brahim, et al. HCALA : Hyperelliptic curve-based anonymous lightweight authentication scheme for Internet of Drones. Pervasive and Mobile Computing, 2023, vol. 92, p. 101798.

[44] M. Nikooghadam, H. Amintoosi, S. H. Islam, M. F. Moghadam, A provably secure and lightweight authentication scheme for internet of drones for smart city surveillance, Journal of Systems Architecture 115 (2021) 101955

[45] S. Hussain, S. A. Chaudhry, O. A. Alomari, M. H. Alsharif, M. K. Khan, N. Kumar, Amassing the security : An ecc-based authentication scheme for internet of drones, IEEE Systems Journal 15 (3) (2021) 4431–4438.

[46] M. Tanveer, A. Alkhayyat, A. Naushad, N. Kumar, A. G. Alharbi, et al., Ruam-iod : A robust user authentication mechanism for the internet of drones, IEEE Access 10 (2022) 19836–19851.

[47] S. Yu, A. K. Das, Y. Park, P. Lorenz, Slap-iod : Secure and lightweight authentication protocol using physical unclonable functions for internet of drones in smart city environments, IEEE Transactions on Vehicular Technology 71 (10) (2022) 10374–10388

[48] Srinivas, J., Das, A. K., Wazid, M., & Vasilakos, A. V. "Designing secure user authentication protocol for big data collection in IoT-based intelligent transportation system". IEEE Internet of Things Journal, (2021) 8(9), 7727-7744.

[49] CHEN, An. "Utilizing the variability of resistive random access memory to implement reconfigurable physical unclonable functions. IEEE Electron Device Letters", 2015, vol. 36, no 2, p. 138-140.

[50] XIONG, Jun, MA, Dongtang, WONG, Kai-Kit, et al. Robust masked beamforming for MISO cognitive radio networks with unknown eavesdroppers. IEEE Transactions on Vehicular Technology, 2015, vol. 65, no 2, p. 744-755.

[51] ZERROUKI, Fahem, OUCHANI, Samir, et BOUARFA, Hafida. A survey on silicon PUFs. Journal of Systems Architecture, 2022, vol. 127, p. 102514.

[52] DODIS, Yevgeniy, REYZIN, Leonid, et SMITH, Adam. Fuzzy extractors : How to generate strong keys from biometrics and other noisy data. In : Advances In Cryptology-EUROCRYPT 2004 : International Conference On The Theory And Applications Of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23. Springer Berlin Heidelberg, 2004. p. 523-540.

[53] DOLEV, Danny et YAO, Andrew. On the security of public key protocols. IEEE Transactions on information theory, 1983, vol. 29, no 2, p. 198-208.

[54] CANETTI, Ran et KRAWCZYK, Hugo. Universally composable notions of key exchange and secure channels. In : Advances in Cryptology—EUROCRYPT 2002 : International Conference on the Theory and Applications of Cryptographic Techniques Amsterdam, The Netherlands, April 28–May 2, 2002 Proceedings 21. Springer Berlin Heidelberg, 2002. p. 337-351.

[55] T. Messerges, E. Dabbish, and R. Sloan, “Examining smartcard security under the threat of power analysis attacks,” IEEE Transactions on Computers, vol. 51, no. 5, pp. 541–552, 2002.

[56] D. Wang and P. Wang, “Two birds with one stone : Two-factor authentication with security beyond conventional bound,” IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 4, pp. 708–722, 2018.

[57] S. Garg, K. Kaur, G. Kaddoum, J. J. P. C. Rodrigues, and M. Guizani, “Secure and lightweight authentication scheme for smart metering infrastructure in smart grid,” IEEE Transactions on Industrial Informatics, vol. 16, no. 5, pp. 3548–3557, 2020

[58] Cheng, J., Yuan, G., Zhou, M., Gao, S., Liu, C., Duan, H., Zeng, Q. : Accessibility analysis and modeling for IoV in an urban scene. IEEE Trans. Veh. Technol. 69(4), 4246–4256 (2020)

[59] Saleem, M.A., Mahmood, K., Kumari, S. : Comments on “AKMIoV : authenticated key management protocol in fog computingbased internet of vehicles deployment”. IEEE Internet Things J. 7(5), 4671–4675 (2020).

[60] Butun, I., Österberg, P., Song, H. : Security of the internet of things : vulnerabilities, attacks, and countermeasures. IEEE Commun. Surv. Tutor. 22(1), 616–644 (2020).

[61] Masood, A., Lakew, D.S., Cho, S. : Security and privacy challenges in connected vehicular cloud computing. IEEE Commun. Surv. Tutor. 22(4), 2725–2764 (2020).

[62] FAA. 2021. Remote Identification of Unmanned Aircraft. Available Online : https ://www.faa.gov/newsroom/remoteid-final-rule

[63] European Commission. 2021. Detailed rules on unmanned aircrafts. Technical Report. European Commission.

[64] “ScaleFyt : Remote ID identification and tracking.” Thales Group. 2022. Accessed : mars. 14, 2024. [Online]. Available : https ://www.scaleflyt.com/ remoteid

[65] “Broadcast location and identification platform (BLIP).” Unifly. 2022. Accessed : mars. 14, 2024. [Online]. Available : https ://unifly.aero/ products/blip

[66] [13] “Relmatech.” 2022. Accessed : mars. 14, 2024. [Online]. Available : https :// www.relmatech.com/

[67] Ullah Jan, S., Bilal, M., Ghani, A., Khan, S., Ahmad, R., Kim, D., 2024. Robust and lightweight authentication for securing communication in the internet-of-drones (iod) environment, in : IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 01–06. doi :10.1109/INFOCOMWKSHPS61880.2024.10620898.

[68] Aydin, Y., Kurt, G. K., Ozdemir, E., & Yanikomeroglu, H. (2022). Authentication and handover challenges and methods for drone swarms. IEEE Journal of Radio Frequency Identification, 6, 220-228.

[69] R. Alkadi and A. Shoufan, ”Unmanned Aerial Vehicles Traffic Management Solution Using Crowd-Sensing and Blockchain,” in IEEE Transactions on Network and Service Management, vol. 20, no. 1, pp. 201-215, March 2023.

[70] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, "A robust ECC-based provable secure authentication protocol with privacy preserving for industrial Internet of Things," IEEE Trans. Ind. Informat., vol. 14, no. 8, pp. 3599–3609, Aug. 2018

[71] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase authentication protocol for wireless sensor networks in distributed IoT applications," in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Istanbul, Turkey, 2014, pp. 2728–2733.

[72] S. Banerjee et al., "A provably-secure and lightweight anonymous user authenticated session key exchange scheme for Internet of Things deployment," IEEE Internet Things J., vol. 6, no. 5, pp. 8739–8752, Oct. 2019.

[73] AVISPA, . Automated validation of internet security protocols and applications. URL : http : //www.avispa-project.org/. 2006.

[74] Von Oheimb, D., 2005. The high-level protocol specification language hlpsl developed in the eu project avispa, in : Proceedings of APPSEM 2005 workshop, pp. 1–17.

[75] Cremers, C.J., 2008. The scyther tool : Verification, falsification, and analysis of security protocols, in : International conference on computer aided verification, Springer. pp. 414– 418.

[76] Cao, J., Ma, M., Fu, Y., Li, H., Zhang, Y., 2021. Cppha : Capability-based privacy-protection handover authentication mechanism for sdn-based 5g hetnets. IEEE Transactions on De- pendable and Secure Computing 18, 1182–1195. doi :10.1109/TDSC.2019.2916593

[77] Meier, S., Schmidt, B., Cremers, C., Basin, D., 2013. The tamarin prover for the symbolic analysis of security protocols, in : Computer Aided Verification : 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings 25, Springer. pp. 696–701.

[78] Surapaneni, P., Bojjagani, S., Khan, M.K., 2024. Vesecure : Verifiable authentication and efficient key exchange for secure intelligent transport systems deployment. Vehicular Com- munications 49, 100822. doi :https ://doi.org/10.1016/j.vehcom.2024.100822.

[79] Dwivedi, S.K., Abdussami, M., Amin, R., Khan, M.K., 2024. D³apts : Design of ecc- based authentication protocol and data storage for tactile internet enabled iod system with blockchain. IEEE Transactions on Consumer Electronics 70, 4239–4248. doi :10.1109/TCE. 2023.3345893.

[80] Sadhukhan, D., Ray, S., Dasgupta, M., Khan, M.K., 2024. Development of a prova- bly secure and privacy-preserving lightweight authentication scheme for roaming services in global mobility network. Journal of Network and Computer Applications 224, 103831. doi :https ://doi.org/10.1016/j.jnca.2024.103831.

[81] umar, V., Ali, R., Sharma, P.K., 2024. Iov-6g+ : A secure blockchain-based data collection and sharing framework for internet of vehicles in 6g-assisted environment. Vehicular Com-640 munications 47, 100783. doi :https ://doi.org/10.1016/j.vehcom.2024.100783.

[82] FAA. 2021. Remote Identification of Unmanned Aircraft. Available Online : https ://www.faa.gov/newsroom/remoteid-final-rule

[83] European Commission. 2021. Detailed rules on unmanned aircrafts. Technical Report. European Commission.

[84] Drone DJ. 2021. Crowdfunded lawsuit filed against the FAA over Remote ID. https ://tinyurl.com/euy342jr. (Accessed : 2024-01-24).

[85] N. Wang, J. Duan, B. Chen, S. Guo, T. Xiang and K. Zeng, "Efficient Group Key Generation Based on Satellite Cluster State Information for Drone Swarm," in IEEE Transactions on Information Forensics and Security, vol. 19, pp. 4464-4479, 2024

[86] Yousef Hashem and Elmedin Zildzic. Secure drone identification with hyperledger iroha. In Proceedings of the 11th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, pages 11–18, 2021.

[87] Aydin, Y., Kurt, G. K., Ozdemir, E., & Yanikomeroglu, H. (2022). Authentication and handover challenges and methods for drone swarms. IEEE Journal of Radio Frequency Identification, 6, 220-228.

[88] M. A. Abdel-Malek, K. Akkaya, A. Bhuyan, and A. S. Ibrahim, “A proxy signature-based swarm drone authentication with leader selection in 5G networks,” IEEE Access, vol. 10, pp. 57485–57498, 2022

[89] Stuart W. Card and Adam Wiethuechter and Robert Moskowitz and Shuai Zhao and Andrei Gurtov Rfc 9434 : Drone Remote Identification Protocol (DRIP) Architecture,jul 2023.

[90] Drone Laws, “Drone laws in France,” Jan 2023. [Online]. Available : https ://drone-laws.com/drone-laws-in-france/

[91] K. Lounis, S. H. H. Ding and M. Zulkernine, "D2D-MAP : A Drone to Drone Authentication Protocol Using Physical Unclonable Functions," in IEEE Transactions on Vehicular Technology, vol. 72, no. 4, pp. 5079-5093, April 2023

[92] T. Alladi, V. Venkatesh, V. Chamola and N. Chaturvedi, "Drone-MAP : A Novel Authentication Scheme for Drone-Assisted 5G Networks," IEEE INFOCOM 2021

[93] P. Gope and B. Sikdar, "An Efficient Privacy-Preserving Authenticated Key Agreement Scheme for Edge-Assisted Internet of Drones," in IEEE Transactions on Vehicular Technology, vol. 69, no. 11, pp. 13621-13630, Nov. 2020

[94] M. A. Khan et al., "A Provable and Privacy-Preserving Authentication Scheme for UAV-Enabled Intelligent Transportation Systems," in IEEE Transactions on Industrial Informatics, vol. 18, no. 5, pp. 3416-3425, May 2022

[95] R. Karmakar, G. Kaddoum and O. Akhrif, "A PUF and Fuzzy Extractor-Based UAV-Ground Station and UAV-UAV Authentication Mechanism With Intelligent Adaptation of Secure Sessions," in IEEE Transactions on Mobile Computing, vol. 23, no. 5, pp. 3858-3875, May 2024.

[96] Y. Wang, H. T. Luan, Z. Su, N. Zhang and A. Benslimane, "A Secure and Efficient Wireless Charging Scheme for Electric Vehicles in Vehicular Energy Networks," in IEEE Transactions on Vehicular Technology, vol. 71, no. 2, pp. 1491-1508, Feb. 2022, doi : 10.1109/TVT.2021.3131776.

[97] Bianchi, Tommaso, Alessandro Brighente, and Mauro Conti. "DynamiQS : Quantum Secure Authentication for Dynamic Charging of Electric Vehicles." Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks. 2024.

[98] I. Ahmim, N. Ghoualmi-Zine, A. Ahmim, and M. Ahmim, "Lightweight Authentication Protocols for Internet of Vehicles : Network Model, Taxonomy and Challenges", PAIS (IEEE International Conference on Pattern Analysis and Intelligent Systems), 2022.

[99] Ahmim, I., Bouakkaz, F., Rachedi, A and Ghoualmi-Zine, N. "LASeR : Lightweight and Secure Remote User Authentication Protocol for Internet of Drones" Journal of Network and Computer Applications 2024.

[100] Ahmim, I., Ghoualmi-Zine, N., Bouakkaz, F., and Rachedi "2AS-DS : Anonymous Authentication Scheme Based on Physical Unclonable Function for Drone Swarms", EDiS (IEEE International Conference on Embedded and Distributed Systems) 2024

[101] Ahmim, I., Ghoualmi-Zine, N., Bouakkaz, F., and Rachedi, A. "Enhancement of a User Authentication Scheme for Big Data Collection in IoT-Based Intelligent Transportation System." IEEE WINCOM (International Conference on Wireless Networks and Mobile Communications), 2023.

[102] YANG, Qing, ZHU, Xiaoqian, WANG, Xiaoliang, et al. A novel authentication and key agreement scheme for Internet of Vehicles. Future Generation Computer Systems, 2023, vol. 145, p. 415-428.

[103] HE, Zhimin, TAN, Weijie, LONG, Yangyang, et al. VC-MAKA : Mutual Authentication and Key Agreement Protocol Based on Verifiable Commitment for Internet of Vehicles. IEEE Internet of Things Journal, 2024.

[104] SUTRALA A K, BAGGA P, DAS A K, et al. " On the designofconditional privacy preserving batch verification-basedauthenticationscheme for Internet of vehicles deployment". IEEETransactions on Vehicular Technology, ,2020.

Annexe A

Liste des publications

International journals

- Ahmim et al. "LASeR : Lightweight and Secure Remote User Authentication Protocol for Internet of Drones", 2025, Journal of Network and Computer Applications (Q1, IF :8, A+) [published] ; https ://doi.org/10.1016/j.jnca.2025.104275
- Ahmim et al. "Security analysis on Three-factor authentication protocol using physical unclonable function for IoV", 2022, International Journal of Information Security (Q1, IF :3.2, A) [published] ; https ://doi.org/10.1007/s10207-022-00595-6

International conferences

- Ahmim et al. 2AS-DS : Anonymous Authentication Scheme Based on Physical Unclonable Function for Drone Swarms, EDiS (IEEE International Conference on Embedded and Distributed Systems) 2024, [published] ; https ://ieeexplore.ieee.org/abstract/document/10783195
- Ahmim et al. "Enhancement of a User Authentication Scheme for Big Data Collection in IoT-Based Intelligent Transportation System", 2023 WINCOM (IEEE International Conference on Wireless Networks and Mobile Communications) [published] ;

  https ://doi.org/10.1109/WINCOM59760.2023.10323030
- Ahmim et al. "Lightweight Authentication Protocols for Internet of Vehicles : Network Model, Taxonomy and Challenges", 2022, PAIS (IEEE International Conference on Pattern Analysis and Intelligent Systems) [published] ; https ://doi.org/10.1109/PAIS56586.2022.9946662