UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

A Systematic Analysis of Security and Performance in the Internet of Drones: Re-evaluation and Enhancement of the PMAP Protocol

Leonardo Pereira de Castro

Itajubá, 21 de maio de 2026

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Leonardo Pereira de Castro

A Systematic Analysis of Security and Performance in the Internet of Drones: Re-evaluation and Enhancement of the PMAP Protocol

Dissertação submetida ao Programa de Pós-Graduação em Ciência e Tecnologia da Computação como parte dos requisitos para obtenção do Título de Mestre em Ciência e Tecnologia da Computação.

Área de Concentração: Matemática da Computação

Orientador: Prof. Dr. Alexandre Carlos Brandão Ramos
Coorientador: Prof. Dr. Marcelo Santiago de Sousa

Maio de 2026
Itajubá – MG

Leonardo Pereira de Castro

A Systematic Analysis of Security and Performance in the Internet of Drones: Re-evaluation and Enhancement of the PMAP Protocol

Dissertação submetida ao Programa de Pós-Graduação em Ciência e Tecnologia da Computação como parte dos requisitos para obtenção do Título de Mestre em Ciência e Tecnologia da Computação.

Prof. Dr. Alexandre Carlos Brandão Ramos
Orientador
Universidade Federal de Itajubá (UNIFEI)

Prof. Dr. Marcelo Santiago de Sousa
Coorientador
Universidade Federal de Itajubá (UNIFEI)

Prof. Dr. Lourenço Alves Pereira Jr.
Membro da banca
Instituto Tecnológico de Aeronáutica (ITA)

Prof. Dr. Laércio Pioli Jr.
Membro da banca
Universidade Federal de Itajubá (UNIFEI)

Prof. Dr. Adler Diniz de Souza
Membro da banca
Universidade Federal de Itajubá (UNIFEI)

Itajubá – MG
2026

Agradecimentos

Esta seção é um espaço para agradecer a todos que, de alguma forma, fizeram parte dessa caminhada até a conclusão desta dissertação.

Quero começar agradecendo à minha esposa e à minha filha, que ainda está por vir. Obrigado por todo o apoio, carinho e compreensão durante esse período. Em muitos momentos precisei me dedicar intensamente a este trabalho e, mesmo assim, sempre encontrei em vocês incentivo, paciência e força para continuar.

Aos meus orientadores, professor Marcelo Santiago e professor Alexandre Ramos, deixo meu sincero agradecimento pela orientação, apoio e confiança ao longo de toda a pesquisa. Também sou muito grato por terem me apresentado à área de cibersegurança na Internet dos Drones, um tema que no início não era o meu foco, mas que acabou se tornando uma grande motivação ao longo do mestrado.

Agradeço também ao professor Adler Diniz, coordenador do curso, pelo apoio no direcionamento da revisão bibliográfica e pelo incentivo durante essa jornada.

Deixo também meu agradecimento ao meu país Brasil, por proporcionar um sistema de educação que tornou possível chegar até aqui.

E, claro, aos meus amigos e familiares, que estiveram ao meu lado em todos os momentos, oferecendo apoio e compreensão, especialmente nas fases mais desafiadoras.

A todos vocês, muito obrigado.

A vida é uma maravilha!

Antônio Peticov

Resumo

Esta dissertação investiga protocolos de autenticação aplicados ao contexto da Internet dos Drones (IoD), um ambiente caracterizado por alta mobilidade, restrições computacionais e rigorosos requisitos de segurança. O trabalho é dividido em duas frentes complementares. Na primeira parte, é conduzida uma revisão sistemática da literatura, seguindo diretrizes metodológicas consolidadas, com o objetivo de identificar, classificar e analisar protocolos recentes de autenticação e acordo de chaves para IoD. São examinados aspectos como topologias de comunicação, modelos de ameaça, primitivas criptográficas empregadas, propriedades de segurança analisadas formal e informalmente, além de métricas de desempenho. Na segunda parte, é realizada uma reavaliação técnica do protocolo PMAP, contemplando os cenários Drone-to-ZSP (D2Z) e Drone-to-Drone (D2D). O protocolo é analisado sob diferentes perspectivas, incluindo o modelo formal Real-or-Random (RoR), a verificação automatizada por ferramentas como AVISPA, Scyther e ProVerif, a análise informal de resistência a ataques e a avaliação dos custos comunicacionais e computacionais. Os resultados indicam que, sob os modelos adotados, não foram identificadas violações das propriedades de segurança especificadas nas verificações formais. Entretanto, a análise também evidencia limitações relacionadas à ausência de Perfect Forward Secrecy, à exposição a ataques de insider privilegiado, à ausência de mecanismos específicos de mitigação contra ataques de negação de serviço e ao custo computacional das operações baseadas no mapa de Hénon em plataformas restritas. Do ponto de vista de desempenho, o protocolo mostrou-se executável em ambiente embarcado, mas com latência significativa, especialmente no cenário D2D. A partir dos resultados obtidos, observa-se que o avanço da área não depende apenas da proposição de novos mecanismos criptográficos, mas também da adoção de metodologias de avaliação mais padronizadas e robustas, combinando modelos formais, análises informais e métricas compatíveis que permitam a comparação entre diferentes estudos. Dessa forma, esta dissertação contribui para uma compreensão mais estruturada das propriedades, limitações e desafios de avaliação de protocolos de autenticação aplicados à Internet dos Drones.

Palavras-chave: Internet dos Drones; Autenticação; Segurança; Protocolos; IoD.

Abstract

This dissertation investigates authentication protocols applied to the context of the Internet of Drones (IoD), an environment characterized by high mobility, computational constraints, and strict security requirements. The work is divided into two complementary parts. In the first part, a systematic literature review is conducted, following established methodological guidelines, with the objective of identifying, classifying, and analyzing recent authentication and key agreement protocols for IoD. Aspects such as communication topologies, threat models, employed cryptographic primitives, security properties analyzed both formally and informally, and performance metrics are examined. In the second part, a technical reassessment of the PMAP protocol is carried out, encompassing the Drone-to-ZSP (D2Z) and Drone-to-Drone (D2D) scenarios. The protocol is analyzed from different perspectives, including the formal Real-or-Random (RoR) model, automated verification using tools such as AVISPA, Scyther, and ProVerif, an informal analysis of resistance to attacks, and an evaluation of communication and computational costs. The results indicate that, under the adopted models, no violations of the specified security properties were identified in the formal verification procedures. However, the analysis also reveals limitations related to the absence of Perfect Forward Secrecy, exposure to privileged insider attacks, the lack of specific mitigation mechanisms against denial-of-service attacks, and the computational cost of Hénon map based operations on constrained platforms. From a performance perspective, the protocol proved to be executable in an embedded environment, but with significant latency, especially in the D2D scenario. Based on the obtained results, it is observed that the advancement of the field depends not only on the proposal of new cryptographic mechanisms, but also on the adoption of more standardized and robust evaluation methodologies, combining formal models, informal analyses, and compatible metrics that enable comparison across different studies. In this way, this dissertation contributes to a more structured understanding of the properties, limitations, and evaluation challenges of authentication protocols applied to the Internet of Drones.

Keywords: Internet of Drones; Authentication; Security; Protocols; IoD.

Contents

List of Figures

List of Tables

Glossary

IoD	*Internet of Drones*
UAV	*Unmanned Aerial Vehicle*
PUF	*Physical Unclonable Function*
CRP	*Challenge–Response Pair*
RSL	*Random Session Label*
DY	*Dolev–Yao*
CK	*Canetti–Krawczyk*
eCK	*Extended Canetti–Krawczyk*
ECC	*Elliptic Curve Cryptography*
HCC	*Hyperelliptic Curve Cryptography*
SSS	*Shamir Secret Sharing*
BCH	*Bose–Chaudhuri–Hocquenghem*
SBTP	*Symmetric Bivariate t-degree Polynomial*
BAN	*Burrows–Abadi–Needham Logic*
ROR	*Real-or-Random Model*
GNY	*Gong–Needham–Yahalom Logic*
ROM	*Random Oracle Model*
GSS	*Ground Station Server*
TA	*Trusted Authority*
CR	*Cluster Router*
RSU	*Roadside Unit*
ZSP	*Zone Service Provider*
AP	*Access Point*
MITM	*Man-in-the-Middle*
PFS	*Perfect Forward Secrecy*
ESL	*Ephemeral Session Leakage*
DoS	*Denial of Service*
KSSTI	*Known Session Specific Temporary Information*
D_x	*Drone x*
ID_x	*Identity of entity x*
PID_x	*Pseudonymous Identity of entity x*
C_x	*Challenge value of entity x*
R_x	*Response value of entity x*

N_x	*Nonce generated by entity x*
SK_xy	*Session Key between entities x and y*
D2Z	*Drone-to-Zone*
D2D	*Drone-to-Drone*

1 Introduction

The rapid evolution of wireless communication technologies, combined with advances in embedded systems, has driven the emergence of the IoD as an extension of the Internet of Things (IoT). Conceptually, the IoD can be understood as a network of interconnected Unmanned Aerial Vehicles (UAVs) capable of exchanging information among themselves and with supporting infrastructures, thus forming a distributed, cooperative, and highly dynamic ecosystem. Within this paradigm, UAVs operate in a coordinated and connected manner, enabling applications in environmental monitoring, smart cities, logistics, agriculture, and military operations [1].

This research is especially relevant to organizations that employ UAVs in sensitive and security-critical scenarios, including military institutions and security forces. In these environments, authentication mechanisms play an important role in preserving the reliability of communications, protecting operational information, and supporting mission continuity. Thus, the study is directed toward application domains in which UAV systems require stronger security guarantees due to their operational relevance and exposure to cyber threats.

However, the distributed nature, dynamic topology, and high mobility of these systems introduce significant challenges related to communication security and reliability. Since communication links are predominantly wireless and operate in open and potentially hostile environments, UAVs become susceptible to different classes of cyberattacks, including eavesdropping, replay, and impersonation.

Furthermore, UAVs operate under severe constraints in terms of energy, processing capacity, and memory, which limits the adoption of conventional cryptographic mechanisms, typically designed for environments with higher computational capacity. Excessively costly protocols may compromise system autonomy and the execution of real-time missions [1].

In this context, the literature has focused on the development of lightweight authentication protocols capable of balancing security guarantees with computational and communication efficiency [2]. To this end, low-cost cryptographic primitives are commonly employed, such as hash functions, XOR operations, elliptic curve cryptography, and Physical Unclonable Functions (PUFs).

The validation of these solutions requires the consideration of realistic adversary models and the use of both formal and informal analyses to evaluate properties such as confidentiality, authenticity, and key secrecy [3]. Therefore, security and performance must

be jointly analyzed, taking into account the operational particularities of the IoD.

Several lightweight authentication protocols have been proposed for IoD environments, addressing different security requirements, system architectures, and operational scenarios [37, 58, 54, 23, 69, 72, 53]. Although these proposals provide relevant contributions to authentication and secure communication in IoD, they differ in terms of architectural complexity, computational cost, communication overhead, and authentication scope.

In this dissertation, PMAP [4] is selected as the main object of analysis because it combines lightweight mutual authentication, session key establishment, and privacy-preserving mechanisms based on PUFs, hash operations, XOR operations, and chaotic maps. These characteristics make it particularly relevant for sensitive IoD applications, such as military and security operations, where secure communication, device legitimacy, operational confidentiality, and mission continuity are essential requirements.

In its original proposal, PMAP was evaluated through formal methods, including BAN logic and the AVISPA tool, as well as through informal security analysis and computational and communication cost assessment. The reported results indicated a suitable balance between security and efficiency for IoD environments. Based on this relevance, this dissertation provides a broader reassessment of PMAP, considering its design, security properties, threat assumptions, verification results, and performance behavior.

The reassessment examines PMAP in terms of its protocol structure, resistance to attacks described in the literature, formal and informal security properties, computational cost, and communication overhead. Based on the limitations identified throughout the analysis, possible mitigation directions are discussed with the objective of strengthening its security properties and outlining potential paths for its evolution in more demanding IoD scenarios.

Figure 1 summarizes the overall workflow adopted in this dissertation, showing how the systematic review supports the classification of authentication protocols and guides the reassessment of PMAP.

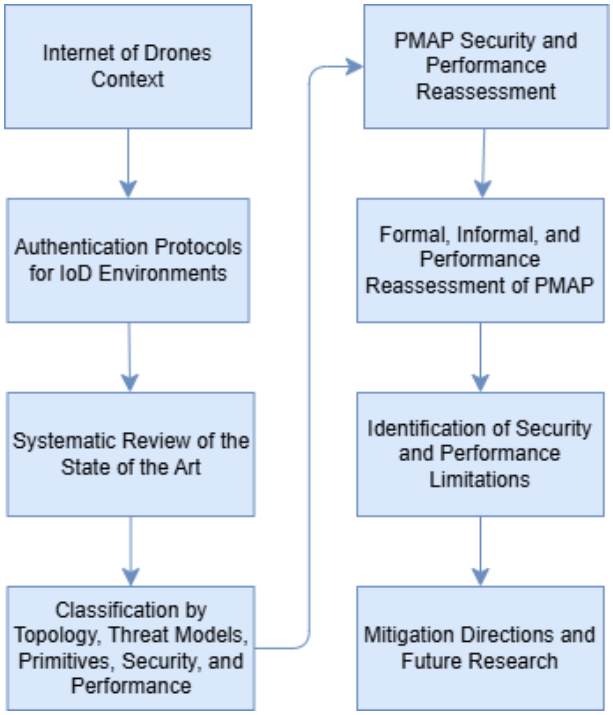

Figure 1 – Workflow adopted for the analysis and reassessment of the PMAP protocol.

1.1 Objectives

1.1.1 General Objective

The general objective of this study is to analyze the PMAP protocol [4] in the context of the IoD, considering its protocol structure, security properties, and computational and communication performance.

1.1.2 Specific Objectives

To achieve the general objective, the following specific objectives are defined:

- To conduct a systematic review of authentication protocols proposed for IoD environments, identifying the main cryptographic mechanisms, topological models, security evaluation approaches, and performance metrics reported in the literature;
- To describe the structure and operation of the PMAP protocol, including its authentication phases, cryptographic primitives, exchanged messages, and session key establishment process;

- To reassess the security properties of PMAP through formal and informal analyses, considering adversary models, verification tools, and attacks described in the literature;
- To evaluate the computational and communication costs of PMAP, considering the operations and exchanged messages involved in its authentication procedures;
- To identify limitations of PMAP and discuss possible improvements aimed at strengthening its security properties and supporting its evolution for more demanding IoD scenarios.

1.2 Methodology

This section presents the methodological structure adopted in this study. The research is characterized as exploratory and analytical. It is exploratory because it investigates authentication protocols proposed for IoD environments, seeking to identify how the literature addresses security evaluation, cryptographic mechanisms, communication topologies, threat models, and performance metrics. This stage provides the theoretical and technical foundation required to contextualize the analysis of the PMAP protocol.

The research is also analytical because it examines PMAP in terms of its structure, assumptions, security properties, limitations, and performance behavior. This analysis involves the interpretation of the protocol specification, the reassessment of its resistance to attacks described in the literature, and the evaluation of its computational and communication costs. Therefore, the study does not aim to propose a new authentication protocol, but rather to analyze an existing one and discuss possible improvements based on the identified limitations.

Regarding the methodological procedures, the study is organized into two main stages. The first stage consists of a literature review on authentication protocols proposed for IoD environments. This stage aims to identify the main approaches adopted in the literature for security analysis and performance evaluation, including the use of formal and informal methods, adversary models, verification tools, cryptographic mechanisms, and computational and communication cost metrics.

The second stage consists of the reassessment of the PMAP protocol in terms of security and performance. In this stage, PMAP is analyzed according to its protocol structure, authentication process, cryptographic primitives, security properties, resistance to attacks described in the literature, and computational and communication costs. Based on the limitations identified during this reassessment, possible improvements are discussed with the aim of enhancing the protocol security.

1.3 Contributions of this Work

This work contributes to the study of authentication protocols in the context of the IoD by combining a systematic organization of the literature with an in-depth reassessment of the PMAP protocol.

The first contribution consists of the systematization and classification of authentication protocols proposed for IoD environments. Based on the analyzed studies, this work organizes the literature according to relevant technical dimensions, including communication topology, cryptographic primitives, threat models, formal and informal security evaluation methods, and computational and communication performance metrics. This organization provides a taxonomic view of the area and supports a more structured understanding of how authentication protocols for IoD have been designed and evaluated.

The second contribution consists of the reassessment of the PMAP protocol [4], considering its protocol structure, security properties, and computational and communication performance. This reassessment expands the original evaluation by analyzing PMAP under additional security perspectives, verification tools, threat assumptions, and attack scenarios described in the literature.

As part of this contribution, the protocol is examined through formal and informal security analyses, including the use of tools and models such as AVISPA, Scyther, ProVerif, and the Real-Or-Random (RoR) model. In addition, its computational and communication costs are evaluated in order to discuss the balance between security and efficiency in resource-constrained IoD environments.

Finally, based on the limitations identified during the reassessment, possible improvements to PMAP are discussed with the aim of strengthening the security properties of the protocol while preserving its lightweight characteristics.

1.4 Outline

This dissertation is organized into five chapters, which progressively present the problem context, theoretical foundations, literature review, conducted evaluation, and final conclusions.

Chapter 1 presents the introduction to the work, contextualizing the IoD, the security and performance challenges, the research objectives, and the methodology adopted for the development of the study.

Chapter 2 addresses the necessary theoretical foundations for understanding the topic, including concepts related to the IoD, the communication models employed, the security

requirements, threat models, and the formal and informal security analysis approaches used in the literature.

Chapter 3 presents the systematic review of the literature, in which different authentication protocols proposed for the context of the IoD are analyzed. The analysis aims to identify the main trends present in the literature, highlighting aspects such as the most used cryptographic primitives, the considered adversary models, the security verification methods employed, and the performance metrics adopted. To facilitate this analysis, graphs are presented that synthesize the frequency and distribution of these characteristics among the analyzed works.

Chapter **??** is dedicated to the evaluation of the PMAP protocol as a case study. In this chapter, its security properties, resistance to attacks described in the literature, and the computational and communication costs associated with its execution are analyzed, along with a discussion of possible improvements and extensions.

Finally, Chapter 5 presents the conclusions of the dissertation, summarizing the main results, contributions, and limitations, as well as outlining directions for future work.

2 Fundamentals

This chapter presents the theoretical foundations that support this dissertation, introducing the main concepts, models, and methods employed in the design and analysis of authentication protocols applied to the IoD. Based on the descriptive analysis of the literature from the selected studies, whose results are presented in Chapter 3, it is observed that the literature recurrently follows a structure composed of the definition of the communication topology, specification of the threat model, description of the cryptographic primitives, protocol formalization, security analysis, and performance evaluation.

Following this same organization, this chapter consolidates the conceptual elements necessary for the subsequent analysis, highlighting that security in IoD depends on the interaction between three central components: the communication model, the threat model, and the cryptographic mechanisms employed [2]. These factors determine simultaneously the achievable security properties and the computational and operational costs of the protocols, critical aspects in environments characterized by wireless communication, high mobility, and severe resource constraints [1].

In this context, the chapter begins with the presentation of typical communication topologies in the IoD, discussing the different interaction models among UAVs, ground stations, and supporting entities. Understanding these topologies is essential, as they directly influence the attack surface and the assumptions adopted in the security analysis of protocols.

Subsequently, commonly considered threat models in the IoD literature are discussed, with a focus on the capabilities attributed to the adversary and the assumptions regarding control over the communication channel and the potential physical compromise of devices [3].

Next, the main cryptographic primitives used in IoD protocols are introduced, highlighting widely employed mechanisms such as hash functions, message authentication codes, symmetric cryptography, and elliptic curve based asymmetric cryptography. The discussion emphasizes the adoption of lightweight primitives compatible with the energy, processing, and memory constraints of UAVs [1].

The chapter also addresses the security evaluation methods employed in the literature, encompassing both informal analyses and formal verification techniques. These methods allow for a complementary assessment of whether protocols satisfy properties such as authenticity, confidentiality, and resistance to attacks under well defined threat models.

Finally, concepts related to the performance evaluation of authentication protocols are

presented, considering metrics such as computational cost and communication overhead. These aspects are essential for analyzing the practical feasibility of the proposed solutions in the context of the IoD.

2.1 Topology

The communication topology describes how the different entities of a system based on the IoD, such as unmanned Aerial vehicles (UAVs), ground stations, and supporting servers, are organized and interact during the execution of a protocol [2]. This structure defines which entities participate in the system, which communication channels are established among them, and which connectivity assumptions are considered.

Figure 2 illustrates a generic IoD topology involving drones, a user, and a ground station. This representation is used to highlight how different entities may participate in the authentication process and how the communication links among them define the operational scenario considered by the protocol.

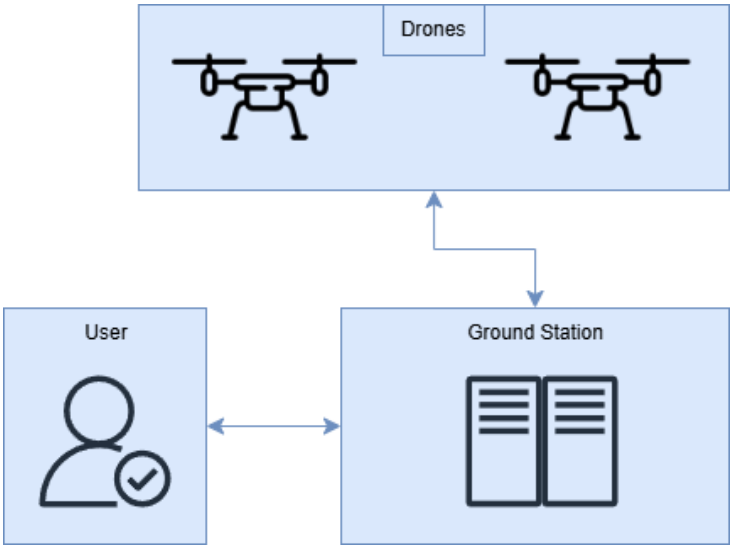

Figure 2 – Example of a generic topology in an IoD environment.

In the context of authentication protocols, the topology plays a fundamental role, as it determines how messages are exchanged between entities and which points of the communication may be subject to interception or manipulation. Thus, the way communication links are established directly influences the security requirements, the need for mutual authentication, and the cryptographic mechanisms adopted [1].

Furthermore, the definition of the topology provides the foundation for characterizing the threat model. Based on the identification of the entities and the used communication

channels, it becomes possible to analyze which components may be targets of attacks and which types of adversaries should be considered. In this way, the topology establishes the scenario in which the security properties of the protocol will be evaluated.

In the PMAP [4], for instance, UAVs communicate with a server responsible for network coordination, while a trusted authority participates in the initial phase of registration and credential distribution. During system operation, the UAV and server execute the authentication protocol through a potentially insecure communication channel, making it necessary to use cryptographic mechanisms to ensure mutual authentication, message integrity, and secure session key establishment.

2.2 Threat Models

The threat model is a concept widely used in security research to describe the adversary's behavior and define the capabilities assumed for an attacker during protocol analysis[2]. In the context of the IoD, this model establishes the actions that an adversary can perform, the resources available for executing attacks, and the type of information that may be compromised throughout the communication process.

The definition of the threat model plays a particularly important role in IoD due to the intrinsic characteristics of these systems. The predominant use of wireless communications, the high mobility of UAVs, and the possibility of physical exposure of devices during operation increase the attack surface [1]. These characteristics require authentication protocols to be evaluated under stronger and more realistic adversarial assumptions.

In general, the literature assumes that communications in IoD occur over wireless links, making them susceptible to both passive and active attacks [3]. In this scenario, a large portion of the works adopts variations of the Dolev-Yao adversary model [5], in which the attacker has full control over the communication channel. According to this model, the adversary is capable of intercepting, modifying, delaying, retransmitting, and injecting arbitrary messages into the network, without, however, breaking cryptographic primitives assumed to be secure.

This adversarial setting is illustrated in Figure 3, which represents the attacker positioned over the wireless communication channel between drones and the ground infrastructure. The figure summarizes the main capabilities usually attributed to the adversary in this context, such as message interception, modification, replay, and injection.

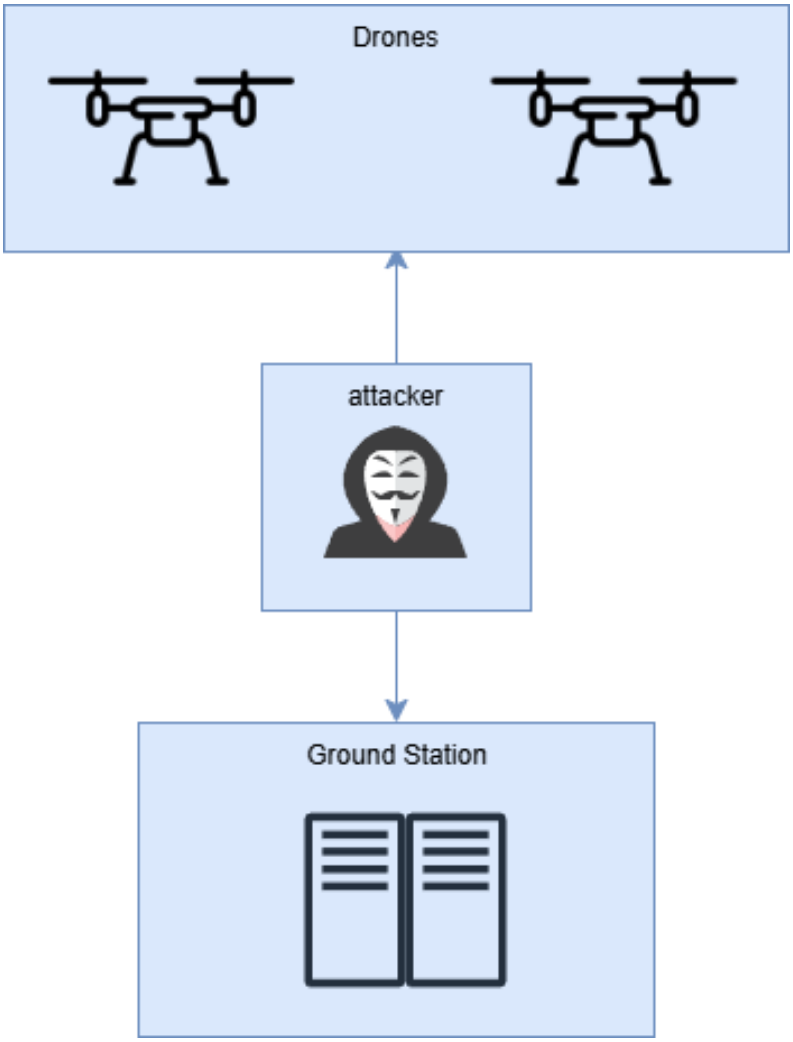

Figure 3 – High-level representation of the Dolev–Yao adversarial model in IoD communication.

In addition to control over the communication channel, several studies consider the possibility of physical compromise of UAVs. This assumption is particularly relevant in IoD, as drones may be captured or accessed during operation. In security models such as the Canetti–Krawczyk (CK) model [6], it is assumed that the adversary may obtain access to information stored on the compromised device, such as identifiers or temporary parameters. This possibility reinforces the need for mechanisms capable of limiting the impact of such exposure.

Although different types of attacks may be explored in these scenarios, the detailed characterization of the mitigated threats depends directly on the analyzed protocol and the considered communication topology. Thus, in this work, the threat model is defined in terms of the general capabilities attributed to the adversary, while the specific analysis of attacks and the achieved security properties is presented in the chapters dedicated to

security evaluation.

Thus, the explicit definition of the threat model establishes the scope within which security properties are analyzed, serving as a basis for both formal and informal evaluations of the protocols investigated throughout this dissertation.

2.3 Cryptographic Primitives

Cryptographic primitives are fundamental mechanisms used as building blocks in the implementation of security systems and protocols. Each primitive is designed to fulfill a specific objective, such as ensuring confidentiality, integrity, authenticity, or anonymity. Classic examples include hash functions, symmetric and asymmetric encryption algorithms, message authentication codes, and key derivation functions [7]. In IoD systems, these primitives are employed to secure communications among the entities defined in the system topology, such as UAVs, servers, and trusted authorities.

Given the previously defined threat model, where communication channels may be intercepted, modified, or exploited through replay attacks, cryptographic primitives are essential to ensuring the required security properties. In this context, in authentication and key establishment protocols, the achieved security properties depend directly on the adopted primitives and how they are combined. An inadequate choice of these primitives may compromise system security, while overly complex solutions may hinder their practical application in resource constrained environments, such as those involving UAVs.

In the context of the IoD, the selection of cryptographic primitives becomes even more important due to the inherent constraints of UAVs, such as limited energy, processing capacity, memory, and bandwidth [1]. Additionally, characteristics such as high mobility, unstable wireless links, and physical exposure of devices increase the attack surface and require cryptographic mechanisms that are both efficient and robust [2]. Therefore, protocols designed for this environment must balance security requirements with operational constraints.

Given these conditions, the literature has prioritized the use of lightweight cryptographic primitives, which are capable of providing adequate security guarantees with low computational and communication costs. This approach aims to enable the deployment of cryptographic protocols on resource constrained devices, such as UAVs used in IoD systems [8].

2.4 Security Evaluation

The security evaluation of cryptographic protocols aims to assess whether the desired security properties hold under a previously defined threat model [7]. In the context of this work, this evaluation is carried out as a continuation of the discussions presented in the previous sections. Initially, the system topology defines the involved entities and the considered communication channels. Next, the threat model establishes the capabilities attributed to the adversary in this scenario. Finally, the cryptographic primitives introduced earlier represent the mechanisms used to protect these communications. In this way, security evaluation consists of analyzing whether, given the adopted topology and the assumptions defined for the adversary, the set of primitives used is capable of ensuring the security properties required by the protocol.

In the context of the IoD, this step assumes a central role, as security guarantees do not depend only on the cryptographic primitives used in isolation, but also on how they are applied within the considered communication architecture and under the defined assumptions for the adversary [1]. In this way, security evaluation seeks to verify if properties such as mutual authentication, key confidentiality, message integrity, anonymity, and resistance to known attacks are effectively preserved under the assumed conditions throughout this chapter.

In general, the literature adopts two complementary approaches for the security evaluation of cryptographic protocols: informal analysis and formal analysis. The former is based on logical and descriptive reasoning to discuss whether the protocol is capable of resisting the attacks considered in the threat model, being widely used in IoD proposals [2]. In this sense, in addition to evaluating the desired properties, informal analysis also allows the identification of suitable guidelines for system operation. The latter relies on mathematical models and, in many cases, automated verification tools, such as model checking techniques, with the objective of validating security properties in a more rigorous manner [9].

In this work, both approaches are considered. Initially, an informal security analysis is presented, in which the desired properties are related to the adversary's capabilities, which must be continuously updated according to the requirements of the analyzed topology, as well as to the mechanisms employed by the protocol. Subsequently, formal evaluation techniques are discussed, with the objective of strengthening security guarantees under defined models.

2.4.1 Informal Security Analysis

Informal security analysis consists of a qualitative evaluation of the protocol, based on the detailed description of its phases and on reasoning about how its mechanisms prevent or mitigate known attacks [1]. This approach is widely used in IoD related works due to its flexibility and ease of adaptation to different threat models and communication topologies [2].

In this type of analysis, the protocol is examined from the perspective of the capabilities attributed to the adversary, as defined in the threat model. Based on this, it is discussed whether an attacker would be able to obtain sensitive information, forge valid messages, compromise session keys, or impersonate legitimate entities during the authentication process.

The IoD literature commonly employs informal analysis to demonstrate resistance to classical attacks, such as message interception, replay attacks, impersonation, Man-in-the-Middle attacks, and denial-of-service attacks [3]. In general, this evaluation is carried out by describing the data transmitted in each phase of the protocol and justifying that critical information is protected by appropriate cryptographic primitives, such as hash functions, MACs, or encryption [2].

Although informal analysis does not provide rigorous mathematical proofs, it plays an important role in identifying evident design flaws, logical inconsistencies, or implicit assumptions that may compromise the security of the protocol [1]. Furthermore, this approach contributes to a better intuitive understanding of the protocol's operation, serving as a basis for more detailed formal evaluations.

In this way, informal security analysis represents a first level of validation for authentication protocols in IoD, complementing formal approaches and providing a clear view of how security properties are achieved under the considered threat models.

2.4.2 Formal Security Analysis

Formal security analysis aims to rigorously verify whether a cryptographic protocol satisfies well defined properties under an explicit threat model. Unlike informal analysis, which is based on descriptive reasoning, formal analysis employs mathematical, logical, or computational models to represent the behavior of both the protocol and the adversary, enabling the systematic validation of properties such as secrecy, authenticity, and key agreement [1].

In the context of the IoD, formal analysis plays a fundamental role, as subtle modeling flaws may not be easily identified through visual inspection, especially in protocols involving multiple entities, states, and message exchanges [2]. Thus, formal methods are

widely employed in the literature to reinforce the security guarantees presented in informal analysis.

In general, formal analysis approaches can be grouped into two main categories: logical models and automated verification tools. Logical models are based on manual mathematical analyses to verify the security properties of a given protocol [10, 11, 12]. In contrast, automated tools use formal specifications and paradigms, such as threat models, to verify the security properties of protocols [13, 14, 15].

Among the most widely used logical models is the Burrows–Abadi–Needham (BAN) logic [11], employed to analyze properties such as mutual authentication and key agreement through the formalization of the beliefs of the protocol participants. BAN logic allows verifying whether, at the end of the execution, the beliefs of the involved parties are consistent with the established security goals. Subsequent extensions and variations, such as the models proposed by Mao and Boyd [16] and the GNY model [17], were developed with the aim of overcoming limitations of the original approach, particularly with regard to the more precise representation of cryptographic operations and intermediate states.

Within the scope of game based security models, the Real-or-Random (RoR) model [10] and approaches based on the Random Oracle Model (ROM) [12] stand out. These frameworks allow the formal analysis of properties such as session key secrecy and resistance to key exposure under explicit assumptions about the adversary's capabilities, modeling interactions through well defined cryptographic experiments.

In addition to logical models, the IoD literature has prioritized the use of automated formal verification tools, such as AVISPA [13], ProVerif [14], Scyther [15], and Tamarin [18]. These tools enable the modeling of protocols and the automatic verification of security properties, such as confidentiality, authentication, and event correspondence, by exploring possible executions under a formal adversary model.

Although formal analysis provides stronger guarantees compared to informal analysis, its results depend on the assumptions adopted in the cryptographic model and on the abstractions used in the modeling. In this way, the results should be interpreted in conjunction with informal analysis and performance evaluation, providing a comprehensive view of the protocol's security and practical feasibility.

In this work, formal analysis is employed as a complementary mechanism to reinforce the security guarantees of the investigated protocols, and is applied in detail in the case study presented in the subsequent chapters.

2.5 Performance Evaluation

In addition to security guarantees, the practical viability of authentication protocols in the IoD depends on their performance [2]. Performance evaluation aims to analyze the impact of the protocol on the computational and communication resources of UAVs, ensuring that security properties are achieved without compromising mission execution or device autonomy.

In the context of IoD, performance evaluation assumes a particularly relevant role due to the severe constraints imposed on drones, such as limited processing capacity, reduced memory, energy constraints, and unstable communication links [1]. Overly costly protocols can result in high energy consumption, increased authentication latency, and network overload, making them impractical in real-world scenarios.

The design of authentication protocols involves a trade-off between security guarantees and resource consumption. As illustrated in Figure 4, the adoption of stronger security mechanisms may affect computational cost, communication overhead, and energy consumption, which are critical aspects in UAV-based environments.

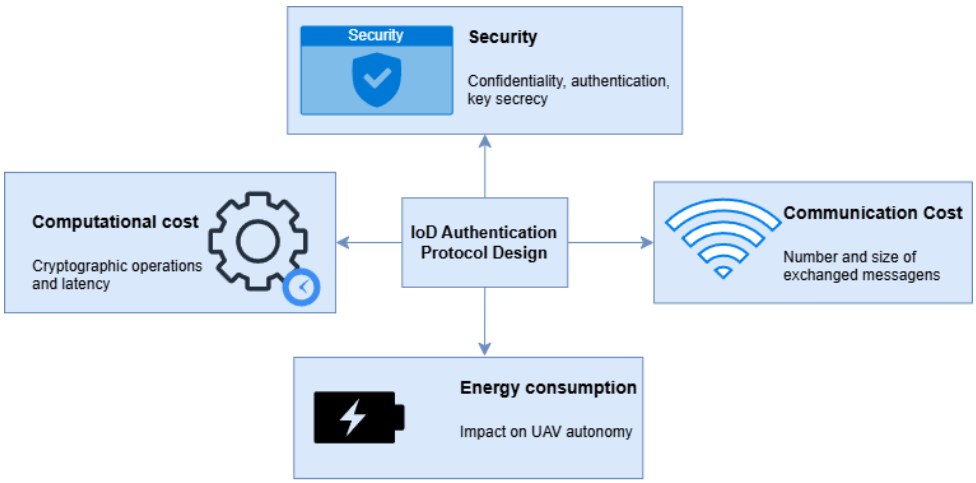

Figure 4 – Security-performance trade-off in IoD authentication protocol design.

In general, the literature evaluates the performance of authentication protocols for IoD based on two main metrics: computational cost and communication overhead [2]. These metrics allow quantifying the impact of the employed cryptographic operations, as well as the volume of data transmitted during protocol execution.

The computational cost refers to the time required for the execution of cryptographic operations in each entity involved in the topology. This analysis generally considers the quantity and type of cryptographic primitives employed by the protocol. As discussed in the cryptographic primitives section, different mechanisms present distinct computational

costs, which directly influence the system's performance. In IoD environments, protocols that prioritize lightweight primitives tend to reduce computational and energy costs [2].

The communication overhead is related to the number of messages exchanged during protocol execution and the total size of these messages. In wireless networks, such as those used in IoD, increasing the volume of transmitted data can directly impact latency, communication reliability, and energy consumption. Therefore, efficient protocols aim to minimize both the number of exchanged messages and the size of the transmitted data [2].

In the literature, performance evaluation is generally conducted through analytical approaches, where the cost of each cryptographic operation is estimated based on experimental measurements or reference values, and through comparisons with related protocols [19, 20, 8]. This approach allows the evaluation of the balance between security and efficiency, highlighting the advantages or limitations of each proposal.

In this work, performance evaluation is carried out analytically and experimentally, considering the computational cost of the cryptographic operations employed and the communication overhead introduced by the analyzed protocols. The results obtained are used to discuss the efficiency of the proposals and their suitability to meet the operational requirements of the IoD.

3 Systematic Literature Review

This chapter presents the systematic literature review conducted in this dissertation, which was carried out to analyze the state of the art regarding the evaluation of authentication protocols in the IoD. A total of 61 articles were selected and used as the basis for the analysis, with the complete list of analyzed studies provided in Appendix A. The review focuses on communication topologies, threat models, cryptographic primitives, security evaluation methods, and performance metrics, thereby connecting the theoretical concepts discussed in Chapter 2 with the evidence extracted from the selected studies.

3.1 Methodology

The research methodology emphasizes transparency, traceability, and methodological rigor, allowing the results to be evaluated and replicated. The process was supported by the Parsifal tool [21] and guided by the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) protocol [22], ensuring a well-defined workflow from study identification to data extraction.

The overall research process adopted in this study is illustrated in Figure 5.

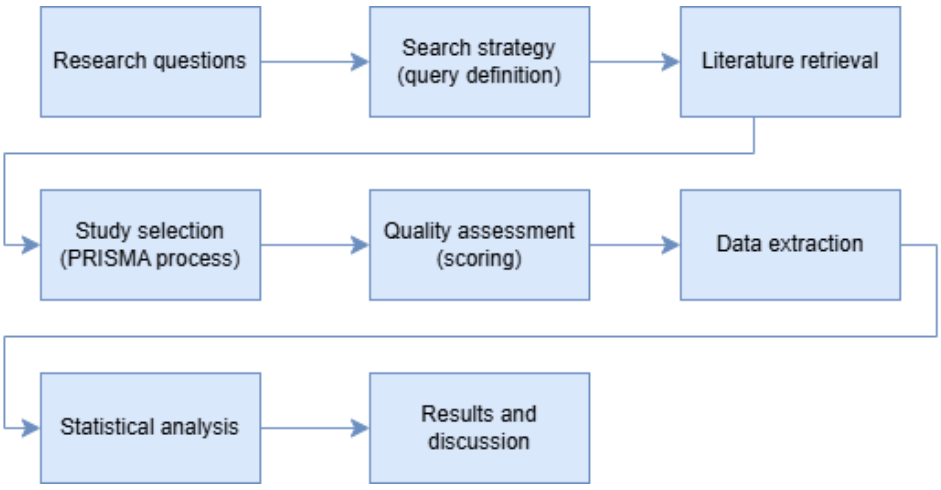

Figure 5 – Overview of the research methodology

This systematic review contributes to the IoD field by providing a statistical view of how authentication protocols are evaluated in the literature. Instead of proposing a new evaluation framework, it examines existing studies to identify patterns in cryptographic

primitives, threat models, formal and informal verification analyses, and performance evaluation.

3.1.1 Databases

The selected studies were retrieved from three well-established databases in computer science and engineering: IEEE Xplore, ACM Digital Library, and Scopus. These databases were chosen because they cover a wide range of publications in areas such as network security, cryptographic protocols, and embedded systems.

3.1.2 Search Strategy

To identify relevant studies, a structured search string was defined based on the PICOC (Population, Intervention, Comparison, Outcome, and Context) framework.

The Population consists of communication protocols applied to UAVs (Unmanned Aerial Vehicles), while the Intervention focuses on the evaluation of these protocols with respect to security properties and verification approaches (both formal and informal). Both motivate the inclusion of terms such as "UAV", "Drones", "IoD", "Mutual Authentication", "Authentication and Key Agreement", and "Security Protocol".

The Comparison was defined as the analysis of different protocols, verification tools, evaluated properties, and simulated or analyzed attack types, while the Outcome aims to identify the most commonly used tools, evaluated security properties, performance metrics, and types of attacks investigated. These components motivated the inclusion of terms related to evaluation approaches, security properties, and verification techniques, such as "Formal Analysis", "Security Evaluation", "Model Checking", and specific tools like "AVISPA", "Scyther", "ProVerif", and "Tamarin".

Finally, the Context refers to scenarios involving UAVs, especially in applications requiring security and efficiency (e.g., critical, multi-domain, military, or civilian environments), ensuring the inclusion of terms related to UAV environments.

Based on these definitions, the following search string was constructed:

```
("UAV" OR "Drone" OR "Internet of Drones" OR "IoD")
AND ("Mutual Authentication" OR "Cross-domain Authentication"
OR "Authentication and Key Agreement")
AND ("Security Protocol" OR "Authentication Protocol" OR "Security")
AND ("Formal Analysis" OR "Informal Analysis" OR "Security Evaluation"
OR "Protocol Verification" OR "Model Checking" OR "AVISPA" OR "Scyther"
OR "ProVerif" OR "Tamarin" OR "Burrows-Abadi-Needham"
OR "BAN Logic" OR "Mao and Boyd logic" OR "Random oracle model"
```

```
OR "ROM" OR "Real-Or-Random" OR "ROR")
```

This expression was applied to the metadata fields in the Scopus, IEEE Xplore, and ACM Digital Library databases, aiming to retrieve the largest possible number of publications relevant to the defined scope.

As a result, the following were identified:

- 67 articles from the Scopus database;
- 65 articles from the IEEE Xplore database;
- 10 articles from the ACM Digital Library.

Among the results obtained, 42 articles were identified as duplicates, resulting in 100 unique publications. In addition, 30 new articles were included through backward snowballing by analyzing the reference lists of selected survey papers, expanding the final corpus to 130 articles considered eligible for the screening and analysis stage.

3.1.3 Screening and Analysis

The screening stage aimed to remove articles that were outside the scope of the research or presented insufficient information for technical analysis. Thus, 69 articles were excluded based on the following criteria:

- Studies that did not use formal analysis tools;
- Studies that did not perform formal or informal verification;
- Studies in which UAVs were not the main focus;
- Studies that did not describe a communication protocol;
- Studies that were inaccessible or had restricted access to the full text.

Table 1 summarizes the exclusion criteria applied during the screening stage.

Table 1 – Exclusion criteria applied during the screening stage

Exclusion Criteria	**Removed Studies**
Studies that did not employ formal security analysis tools, such as AVISPA, Scyther, ProVerif, Tamarin, or equivalent verification approaches	8
Studies that did not provide either formal or informal security verification of the proposed protocol	20
Studies in which UAVs, drones, or IoD environments were not the main application focus	5
Studies that addressed security, communication, or networking aspects but did not describe an authentication or communication protocol	24
Studies whose full text was inaccessible or available only under restricted access, preventing complete technical analysis	12
Total	**69**

Next, a methodological quality assessment was conducted in order to select only articles that met minimum technical and scientific rigor. For this purpose, each study was evaluated based on the following questions:

- Does the study use any threat model?
- Does the study apply any method of formal verification?
- Does the study clearly describe the computational and communication costs?
- Are the cryptographic primitives identifiable and well described?
- Does the study report the formal computational cost?

Each item was scored as 0 (does not meet), 0.5 (partially meets), or 1 (fully meets). Studies that obtained a total score equal to or greater than 3 were considered eligible for the data extraction stage. As a result, 61 articles were selected for subsequent analyses.

The study selection process is illustrated in Figure 6.

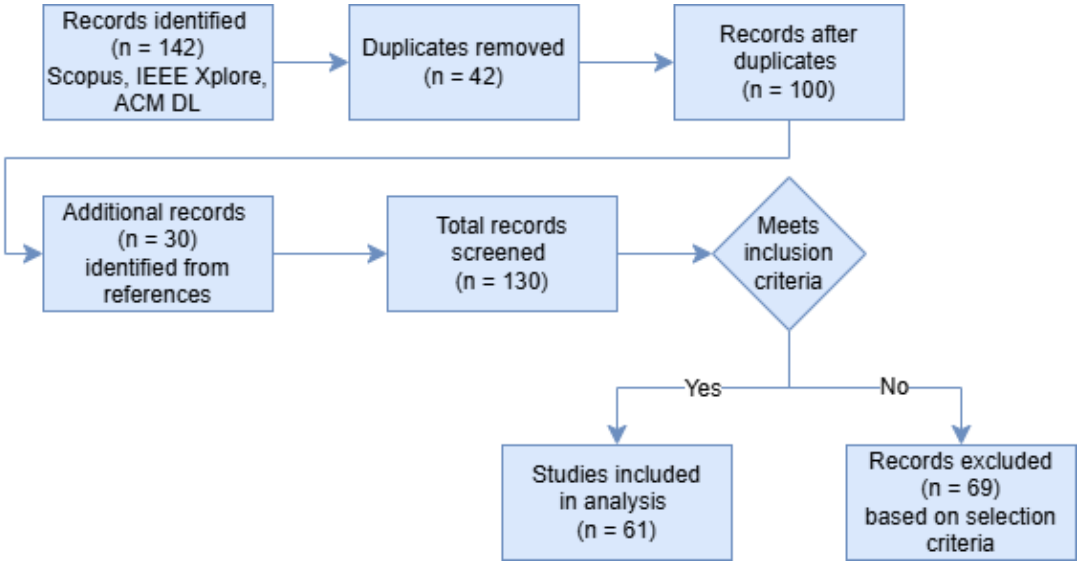

Figure 6 – Study selection process based on PRISMA guidelines

3.1.4 Data Extraction

For the data extraction stage, a standardized form was created with the goal of collecting, in a structured and consistent manner, the main information from each selected article. The extracted data covered key technical aspects necessary for the comparative analysis of authentication protocols applied to the IoD. The following fields were defined:

- **Cryptographic primitives used:** cryptographic techniques applied in the protocol;
- **Adopted topology:** description of the actors involved in the communication (e.g., Drone, User, Ground Station, Trusted Authority);
- **Considered threat model:** formally described adversarial scenarios in which the protocol is evaluated;
- **Formal security tool used:** formal verification tools such as AVISPA [13], ProVerif [14], Scyther [15], Tamarin [18], among others;
- **Formal security model:** theoretical methods adopted for formal validation, such as BAN Logic [11], ROM (Random Oracle Model) [12], ROR (Real-Or-Random) [10], etc.;
- **Informal security analysis:** qualitative assessments of security properties and attacks, without the use of formal tools;

- **Practical computational cost:** experimental data or informal estimates of execution time or resource consumption;
- **Communication cost:** volume of data exchanged between parties during protocol execution;

The extracted data are presented in Appendix A.

3.2 Results

The analysis of authentication protocols in UAV networks requires a structured and rigorous process, in which the use of verification tools plays a central role. The articles selected in this review mostly follow a well-defined pattern, which includes: the definition of the threat model, the protocol topology, the description of the cryptographic primitives used, the detailing of the protocol's operation, the security analyses (formal and informal), and finally, the performance evaluation.

This recurring structure enabled the extraction of valuable information, which will be presented and discussed throughout this section. To facilitate understanding, the results are organized into the following subsections:

- Topology
- Threat Model
- Cryptographic Primitives
- Informal Security Analysis
- Formal Security Analysis
- Performance

This division aims to systematically examine each stage of the verification process of protocols designed for IoD applications. The analysis of the main aspects identified in the studies will be accompanied by charts and statistics, providing a comprehensive and well-founded view of the current state of research in the field.

3.3 Topology

The topology analysis focuses on the communication and architectural organization adopted by authentication protocols in the context of the IoD. As discussed in the fundamentals chapter, IoD systems may involve different entities, such as drones, servers, or users, which interact to support secure communication and authentication.

Based on this perspective, the topologies identified in the selected studies were analyzed by mapping the entities involved in each proposed authentication protocol. The analysis considered the frequency with which each type of entity appeared in the literature, as well as its role in the establishment of authentication and in the structural organization of IoD systems, making it possible to identify recurring architectural patterns used in authentication protocols for IoD environments.

The identified entities were subsequently grouped into categories, described as follows:

- **Drone:** central element of IoD, responsible for task execution, data collection and transmission, as well as interaction with other nodes in the network.
- **Ground Station Server (GSS):** responsible for managing, monitoring, and establishing communication with drones.
- **User:** human or end system that interacts with the drone, typically initiating commands or receiving data.
- **Trusted Authority (TA):** responsible for issuing and validating identities, distributing cryptographic keys, and supporting trust management.
- **Blockchain:** decentralized infrastructure used for immutable record storage and distributed authentication.
- **Control Room (CR):** responsible for coordinating and monitoring system operations.
- **Vehicle:** ground vehicles that interact with UAVs in intelligent transportation systems.
- **Road Side Unit (RSU):** units deployed on urban roads or highways that act as intermediaries between vehicles and UAVs.
- **Zone Service Provider (ZSP):** regional entity responsible for authentication within geographically defined areas, common in zone based distributed architectures.

- **Access Point (AP):** access point (e.g., Wi-Fi) used to connect UAVs or users to a local network or the Internet.
- **Sensor:** embedded or external devices that provide environmental or operational data to the drone, and may participate in the authentication process to ensure the integrity of sensory data.

Figure 7 presents a representative example of a topology employed in IoD systems, similar to the architectures described in [8, 23, 24]. The presence of multiple drones (D1, D2, and D3) can be observed, operating autonomously in the execution of field tasks. In addition, a GSS is included, responsible for mediating and coordinating communications, as well as a CR, in charge of operational supervision, ensuring that the system remains stable and functional. Finally, the presence of an end user connected to the infrastructure is considered, responsible for receiving the data collected by the drones.

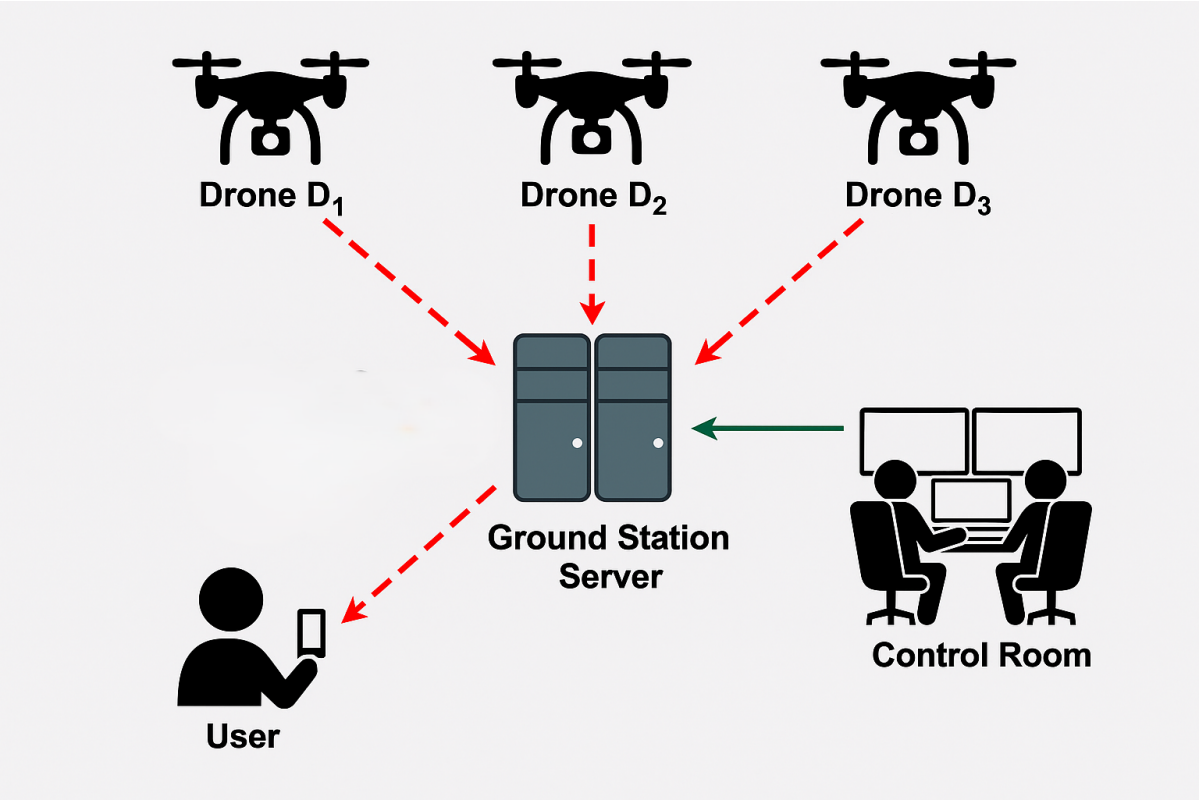

Figure 7 – Example of topology in drone based networks

Figure 8 presents the frequency of the entities identified in the analyzed protocols. It can be observed that, in addition to the drone, the most recurrent elements are the GSS and the user, highlighting the centrality of the interaction among the aerial vehicle, ground infrastructure, and the user in authentication processes.

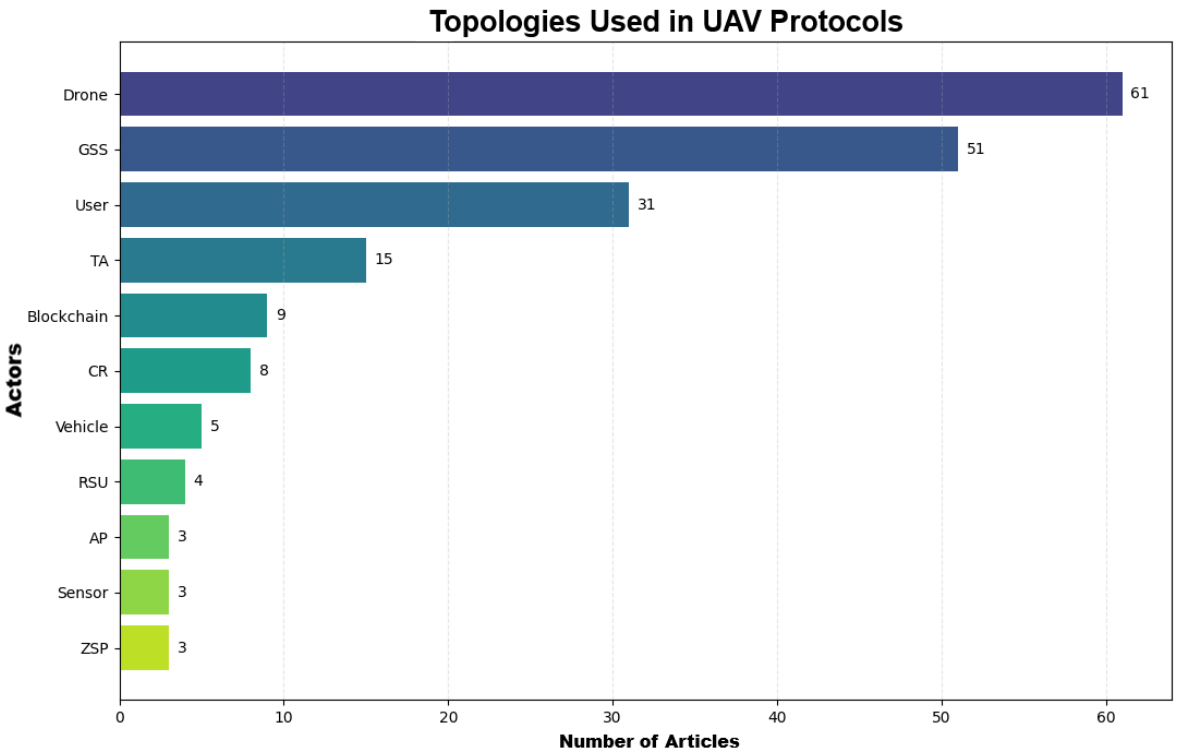

Figure 8 – Frequency of actors in authentication protocols for UAVs

This pattern indicates that most proposals are oriented towards supervised control scenarios, where identity validation and key establishment depend on a reliable intermediary entity. Although this approach simplifies authentication management, it introduces a structural dependency on central components, which can become single points of failure or critical targets in attack scenarios.

It is worth noting the presence of decentralized technologies, such as Blockchain, identified in nine studies [41, 37, 38, 46, 54, 51, 55, 27, 45], indicating a trend towards incorporating decentralization mechanisms. However, this adoption is still limited and, in many cases, does not completely eliminate the dependency on central entities, suggesting that the proposed decentralization is partial or hybrid.

Other actors, such as vehicle, RSU, and sensor, appear in more complex architectures, reflecting the expansion of protocols to heterogeneous environments. This diversity, although increasing the applicability of the models, also increases the attack surface and imposes additional challenges to ensuring consistent security across different domains.

In addition, the entities composition adopted in each protocol was also investigated. Figure 9 presents the frequency of the main combinations observed in the selected studies.

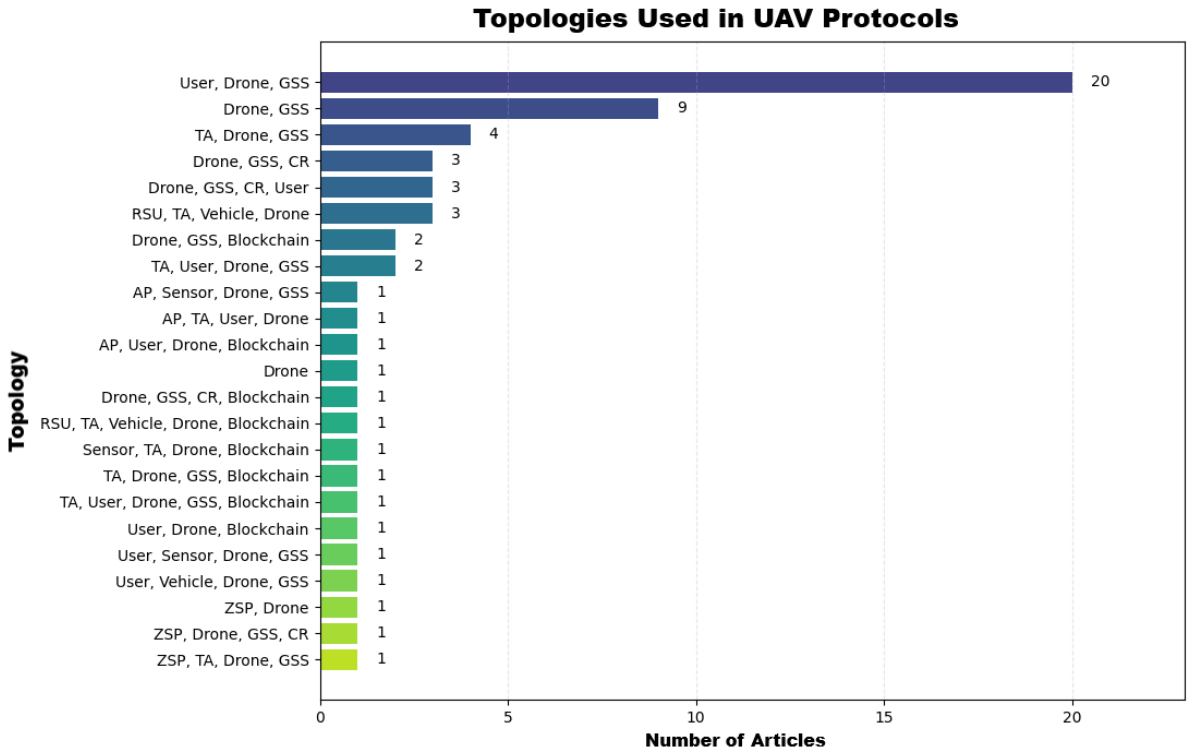

Figure 9 – Frequency of the main topological compositions

It can be observed that the topology composed of drone, GSS and user configures as the predominant architecture in the analyzed literature. Subsequently, simplified configurations, such as drone and GSS, as well as more elaborate compositions that incorporate entities like TA, Blockchain, Vehicle and Sensor are highlighted. These results indicate both the consolidation of a centralized architectural model and the coexistence of proposals oriented towards decentralization and integration with intelligent infrastructures.

3.4 Threat Models

As discussed in Chapter 2, the threat model plays a central role in the design and analysis of protocols for the IoD, as it defines the capabilities attributed to the adversary and delineates the scope of security guarantees.

For the realization of the analyses presented in this section, the threat models explicitly adopted in each of the selected articles were extracted and systematized. From this consolidation, it was possible to identify three predominant models employed in the analyzed protocols, which are described below:

- **Dolev–Yao (DY):** symbolic model in which the adversary has full control over the communication channel, being able to intercept, modify, or retransmit messages

arbitrarily. It assumes ideal cryptographic primitives and is widely used to analyze resistance to active and passive attacks in insecure networks [5].

- **Canetti–Krawczyk (CK):** computational model based on security experiments that considers multiple concurrent sessions and allows adversarial queries that expose certain internal states of the entities [6].
- **Extended Canetti–Krawczyk (eCK):** extension of the CK model that expands the adversary's capabilities by considering additional scenarios of secret exposure, including the revelation of ephemeral keys and combinations with long term keys. It is used to analyze properties such as forward secrecy and resistance to key compromise attacks [80].

Figure 10 presents the distribution of the threat models identified in the analyzed protocols. A significant predominance of the DY model can be observed, adopted in 54 out of the 61 selected studies. This result indicates that most of the literature still bases its analyses on a symbolic model in which the adversary has full control over the communication channel, even though it assumes idealized and unbreakable cryptographic primitives.

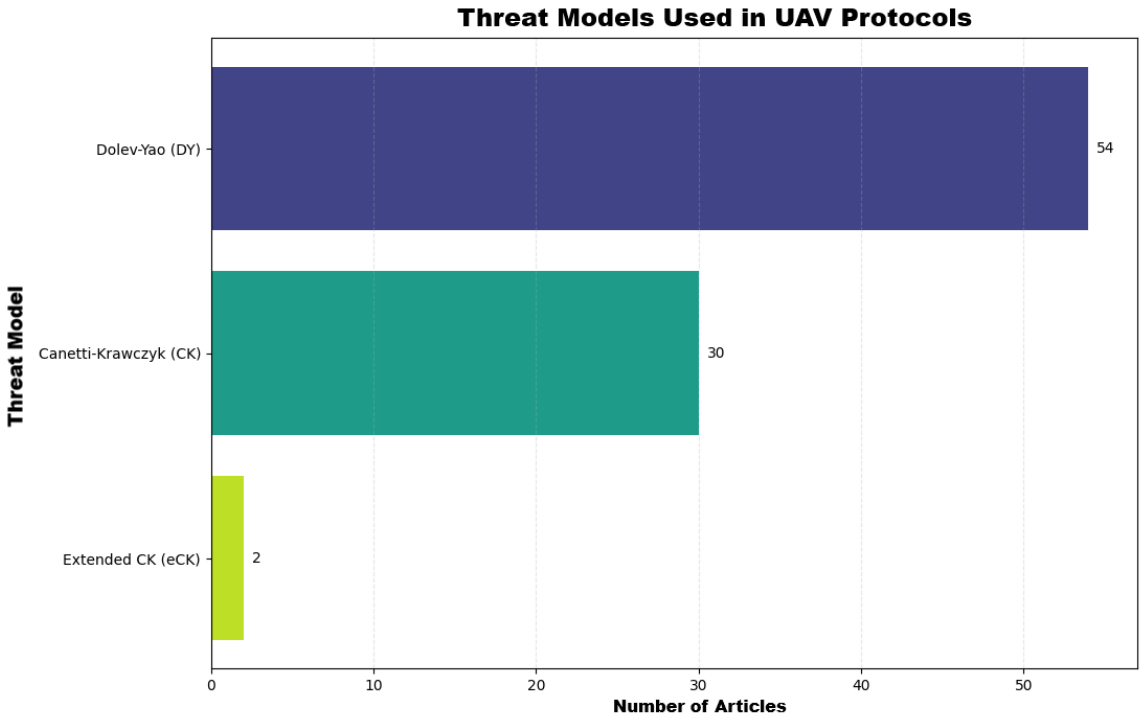

Figure 10 – Distribution of threat models

The CK model was identified in 30 studies, frequently employed in conjunction with the DY model. Unlike the purely symbolic approach, the CK model adopts a computational model based on security experiments, allowing for the consideration of multiple concurrent sessions and partial exposure of internal states. In many works, this model is also used under the assumption of physical capture of UAVs, expanding the scope of evaluated threats and bringing the analysis closer to more realistic scenarios.

In turn, the Extended eCK model was identified in only two studies. Although it provides stronger security guarantees, particularly with regard to the exposure of ephemeral keys and resistance to advanced attacks, its limited adoption may be associated with the complexity required to demonstrate security under enhanced adversarial capabilities.

In addition to the isolated analysis of each model, the combined use of different threat frameworks was also investigated. Figure 11 shows that a significant portion of the protocols employs more than one model to support their security guarantees.

The combination of the DY and CK models stands out, being identified in 26 studies. This strategy allows covering two complementary levels of abstraction: while DY validates resistance against classical interception and message manipulation attacks, CK enables the evaluation of properties under multiple sessions and partial exposure of secrets. The combined use of these models strengthens the security argumentation and enhances the robustness of the presented conclusions.

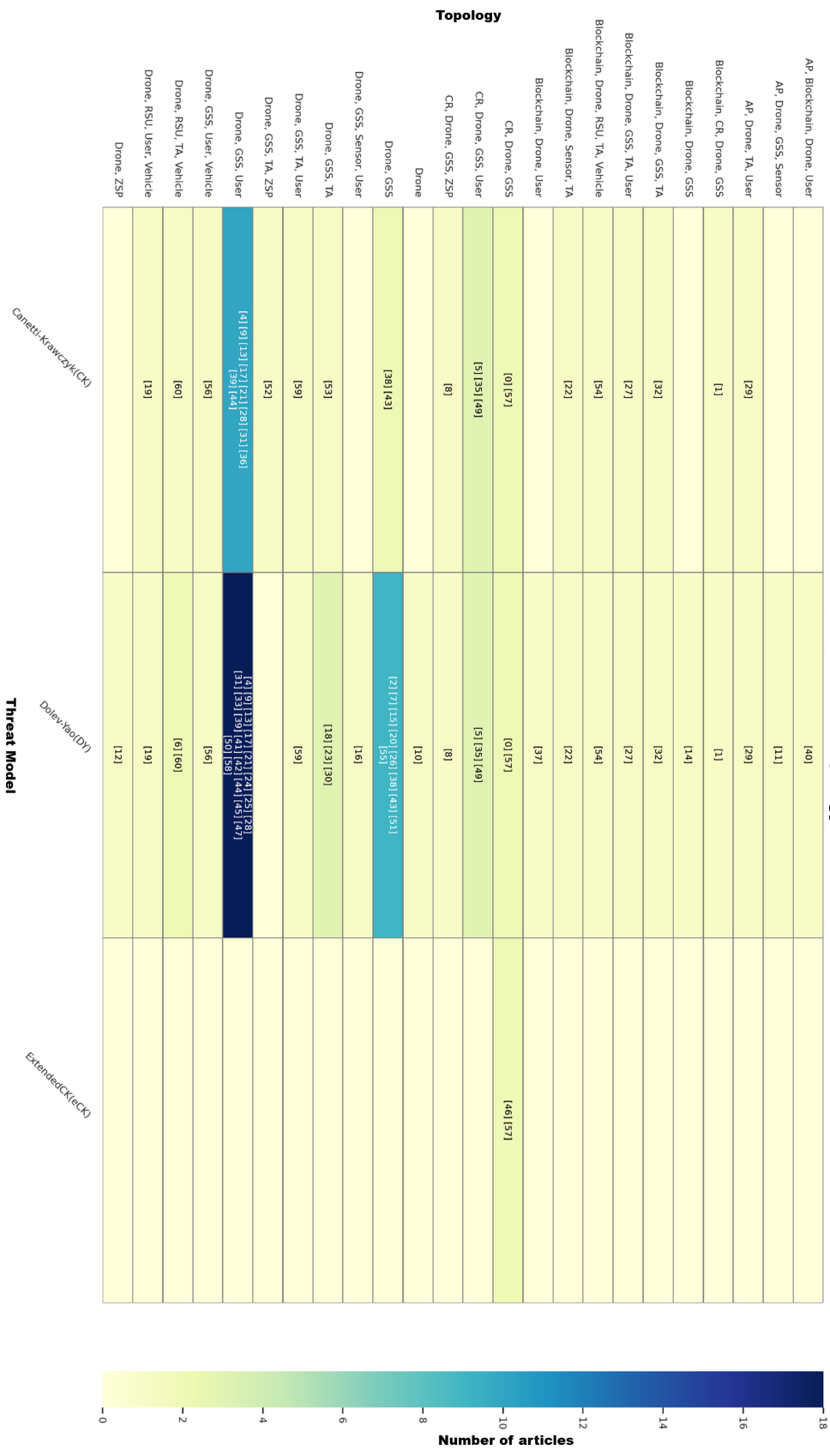

Figure 11 – Frequency of threat models across topologies

The analysis shows that, although the DY model remains dominant, there is a growing concern with more realistic modeling approaches, such as CK and eCK. However, the low adoption of the eCK model suggests that a significant portion of the literature still does not evaluate protocols under enhanced adversarial capabilities, which may represent a gap in security validation for highly dynamic IoD environments susceptible to physical compromise.

3.5 Cryptographic Primitives

In the context of this research, 16 distinct cryptographic primitives were identified across the 61 analyzed studies. These primitives play specific roles in the construction of protocols, being selected according to the topology, threat model, and performance constraints of each proposal.

Figure 12 presents the distribution of the main cryptographic primitives identified. An expressive predominance of hash functions and XOR operations is observed, widely used in lightweight protocols due to their low computational complexity and reduced energy consumption. This pattern indicates a strong orientation of the literature towards efficient solutions compatible with embedded environments. However, this choice also suggests a possible limitation in terms of cryptographic robustness, especially in scenarios that require stronger guarantees, such as PFS and resistance to long term secret exposure.

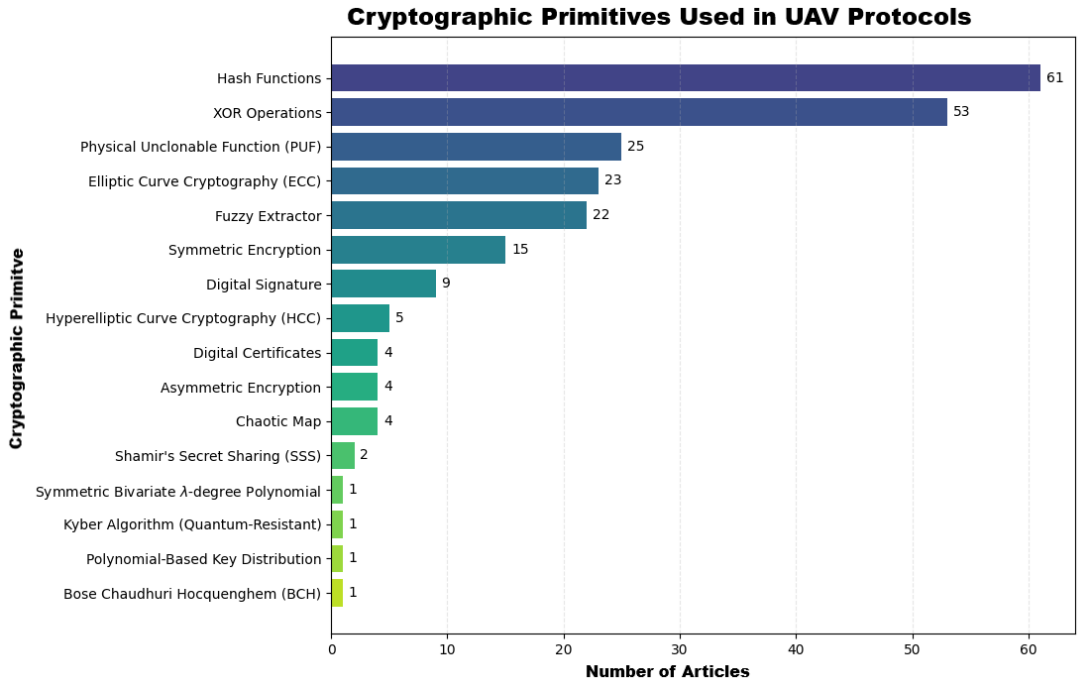

Figure 12 – Frequency of cryptographic primitives

In addition to these primitives, the recurrence of mechanisms such as Physical Unclonable Functions (PUFs), Elliptic Curve Cryptography (ECC) and Fuzzy Extractors is notable. The adoption of these techniques suggests an attempt to mitigate limitations of lighter approaches, incorporating greater cryptographic robustness and, in some cases, hardware based protection. However, these solutions often introduce additional dependencies, such as storage of CRPs (challenge-response pairs) or increased implementation complexity, which may impact scalability and introduce new attack vectors.

Although less frequent, other primitives such as Hyperelliptic Curve Cryptography (HCC), Shamir's Secret Sharing (SSS), and the post quantum scheme Kyber were also identified. The presence of these approaches indicates a trend toward cryptographic diversification in the literature, reflecting concerns with resilience against advanced attacks and post quantum scenarios. However, their low adoption suggests that such solutions are not yet widely viable in resource constrained environments, highlighting a gap between theoretical proposals and practical applicability.

Additionally, it was observed that not all studies explicitly specify the cryptographic algorithms used. In many cases, authors mention only the use of symmetric or asymmetric cryptography, without detailing the employed algorithms [39, 73, 24]. This lack of specification compromises the reproducibility of the proposals and hinders a precise security assessment, representing a recurring limitation in the analyzed literature.

3.6 Informal Security Analysis

The informal analysis was conducted based on the systematic extraction of all security properties explicitly evaluated in the 61 selected studies. In total, 45 distinct properties were identified, covering both specific threats and formal guarantees declared by the authors.

Figure 13 presents the frequency with which these properties and threats are discussed in the analyzed literature. It can be observed that attacks such as replay, impersonation, Man-in-the-Middle (MitM), and drone capture are among the most recurrent. This pattern suggests that authentication protocols for IoD primarily focus on mitigating classical attacks associated with communication channel control and the physical compromise of devices.

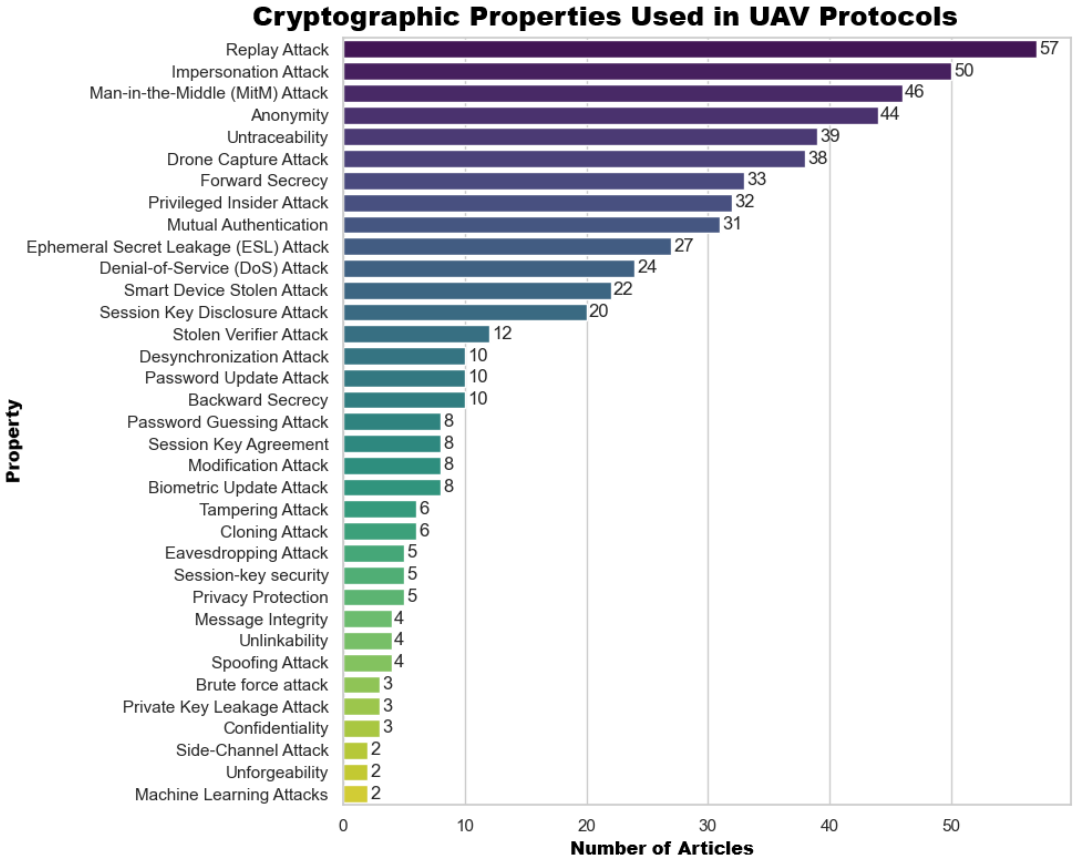

Figure 13 – Frequency of security properties and threats

In this context, the use of cryptographic primitives plays a fundamental role in mitigating these threats. In particular, hash functions, XOR operations, and nonce based mechanisms are widely used to ensure properties such as freshness and resistance to replay attacks. However, the predominance of these lightweight primitives suggests that the provided protection is often limited to basic attack scenarios and may be insufficient against more sophisticated adversaries.

Additionally, properties related to anonymity and untraceability also show significant incidence, indicating a growing concern within the scientific community regarding the protection of drone identities and the preservation of their movement patterns in the IoD ecosystem.

This concern is particularly relevant in scenarios where the drone may be physically captured by an adversary. Under such conditions, the attacker may exploit information stored on the device to carry out more sophisticated attacks, compromising the node's identity and potentially enabling tracking or system infiltration.

Figure 14 illustrates the correlation between the adopted threat models and the evaluated security properties. The DY model stands out as the most widely used, especially in

analyses involving Replay, Impersonation, and MitM. Its predominance can be attributed to its modeling simplicity and its ability to represent an adversary with full control over the communication channel.

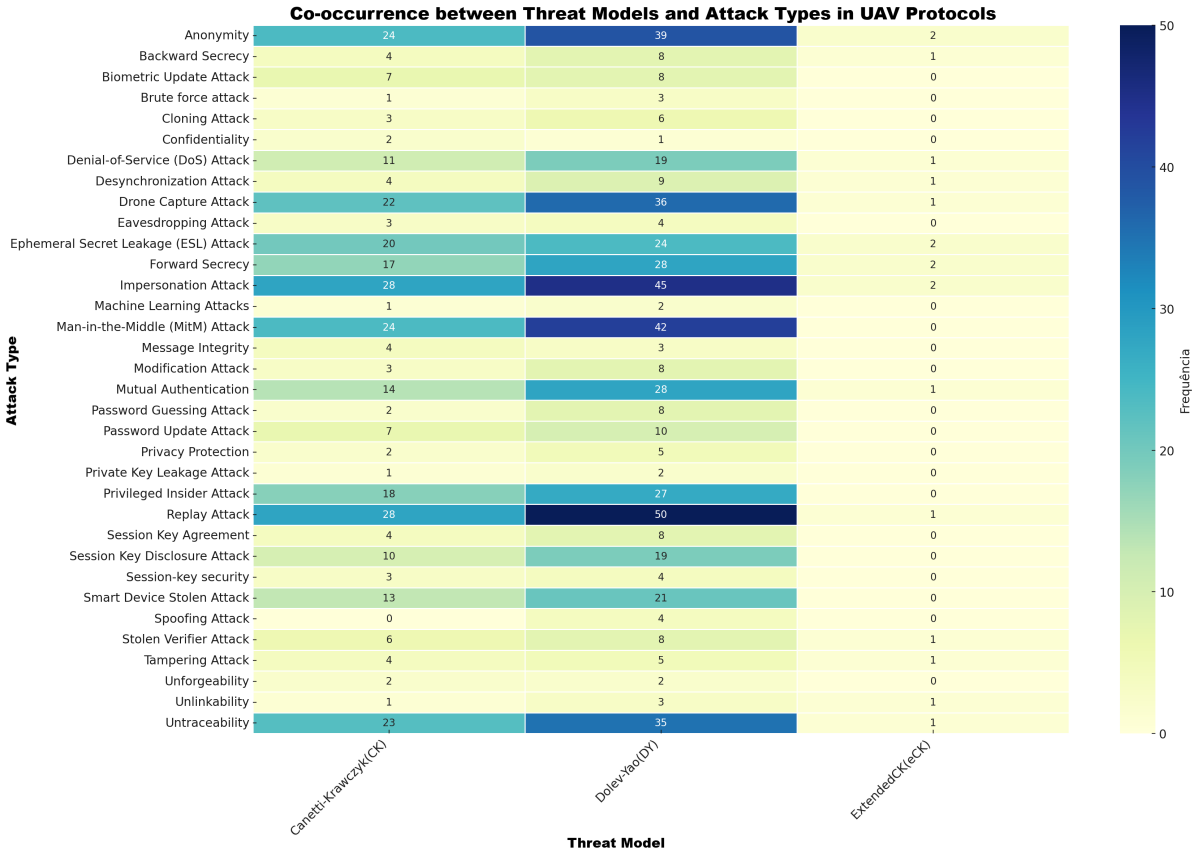

Figure 14 – Correlation between threat models and security properties

The CK model also shows significant presence, being mainly applied in the evaluation of scenarios involving partial exposure of internal states, Drone Capture, and properties such as anonymity and session key secrecy. In 26 studies, the combined use of DY and CK models was observed, highlighting a complementary strategy: while DY covers classical message manipulation attacks, CK extends the analysis by considering multiple sessions and controlled disclosure of secrets.

In turn, the eCK model was identified in a limited number of studies, appearing primarily in analyses involving Forward Secrecy and robust mutual authentication. The limited adoption of this model may be associated with the formal complexity required for its application.

Although the DY model allows covering a wide range of network attacks, its symbolic nature may restrict the analysis of more sophisticated scenarios involving partial compro-

mise of secrets. In this sense, expanding the use of more robust computational models, such as CK and eCK, could strengthen the rigor of the evaluations presented in the literature. A relevant gap is also observed in the analysis of emerging threats, such as machine learning assisted attacks, in which the adversary collects system data to train predictive models capable of inferring the behavior of the adopted cryptographic primitives. In [28], for example, the collection of information from the drone's PUF is investigated with the aim of predicting its responses.

3.7 Formal Security Analysis

In the formal analysis, six distinct formal models were identified among the 61 analyzed studies, employed to rigorously describe and verify the proposed authentication protocols. These models encompass approaches based on computational experiments, formal logics, and specific frameworks for the verification of security properties.

The main identified models are presented as follows:

- **Real-or-Random (ROR):** Computational model that evaluates whether an adversary can distinguish between a real session key and a random key, being widely used to analyze the security of key establishment protocols [10].
- **Burrows–Abadi–Needham (BAN):** Logical model based on the beliefs of participating entities, used to verify authentication properties at the end of the protocol execution [11].
- **Random Oracle Model (ROM):** Computational model that treats hash functions as ideal oracles, employed in the formal analysis of hash based protocols [12].
- **Mao and Boyd Logic:** Extension of BAN logic that incorporates additional elements, such as identity and sessions, allowing a more detailed analysis of protocol interactions [16].
- **GNY Logic (Gong–Needham–Yahalom):** Variant of BAN logic that models more refined beliefs regarding control over keys and messages [17].
- **Ouafi–Phan:** Model applied to protocols in wireless and resource constrained environments, focusing on properties such as anonymity and untraceability [81].

The frequency of occurrence of these models is presented in Figure 15, which summarizes the main formal methods employed in the reviewed studies.

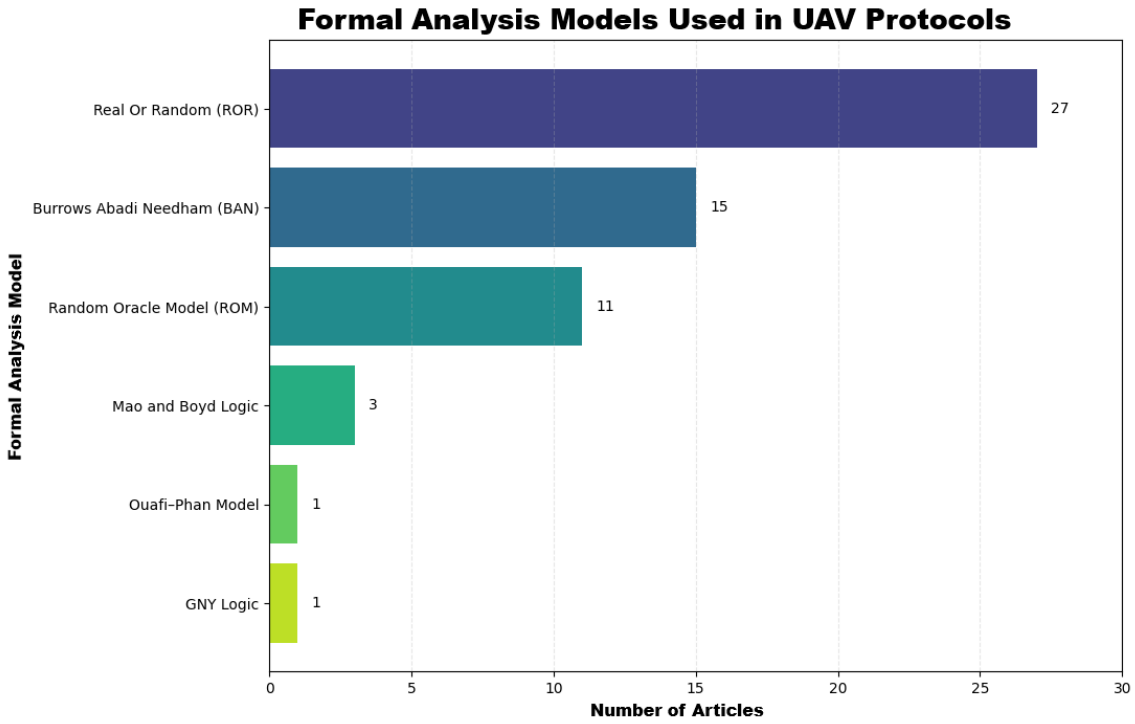

Figure 15 – Frequency of formal analysis models

The recurrence of ROR, BAN, and ROM may be related to the type of analysis usually conducted in authentication protocols. ROR and ROM are commonly adopted in proof-based evaluations, often structured through security games or sequences of experiments, which support the analysis of adversarial advantage and session key security. BAN logic, in turn, is frequently used to reason about authentication properties, such as freshness, message origin, and key agreement.

The results also show that several studies employ more than one formal analysis model. In particular, combinations between models focused on key security proofs and models based on authentication logic are observed. This suggests an attempt to evaluate protocol security properties from complementary perspectives.

On the other hand, models such as Mao and Boyd, GNY, and Ouafi-Phan appear with lower frequency, generally associated with more specific scenarios. Although less frequent, their presence demonstrates the methodological diversity employed in the formal analysis of protocols.

In addition to the models presented in Figure 15, formal analysis is also frequently conducted using automated verification tools. These tools enable the modeling of protocols, the simulation of executions under different adversarial capabilities, and the systematic verification of properties such as secrecy and authentication.

Among the identified tools, the following stand out:

- **AVISPA:** Widely used tool for verifying protocols under the Dolev–Yao model, allowing the identification of vulnerabilities such as replay and Man-in-the-Middle attacks [13].
- **ProVerif:** Tool based on the applied pi-calculus, which provides a set of constructs used to describe protocols. It is employed in the analysis of properties such as secrecy, authentication, and anonymity, including in multiple concurrent sessions [14].
- **Scyther:** Automated tool aimed at verifying confidentiality and authentication properties in protocols modeled under the Dolev–Yao model [15].
- **Tamarin:** Tool used for the formal analysis of complex cryptographic protocols, enabling the validation of security properties under different adversarial assumptions [18].
- **SPIN:** Model checking tool applied to the verification of logical and behavioral properties in concurrent systems, including security protocols [9].

Figure 16 presents the distribution of the main tools employed in the analyzed studies.

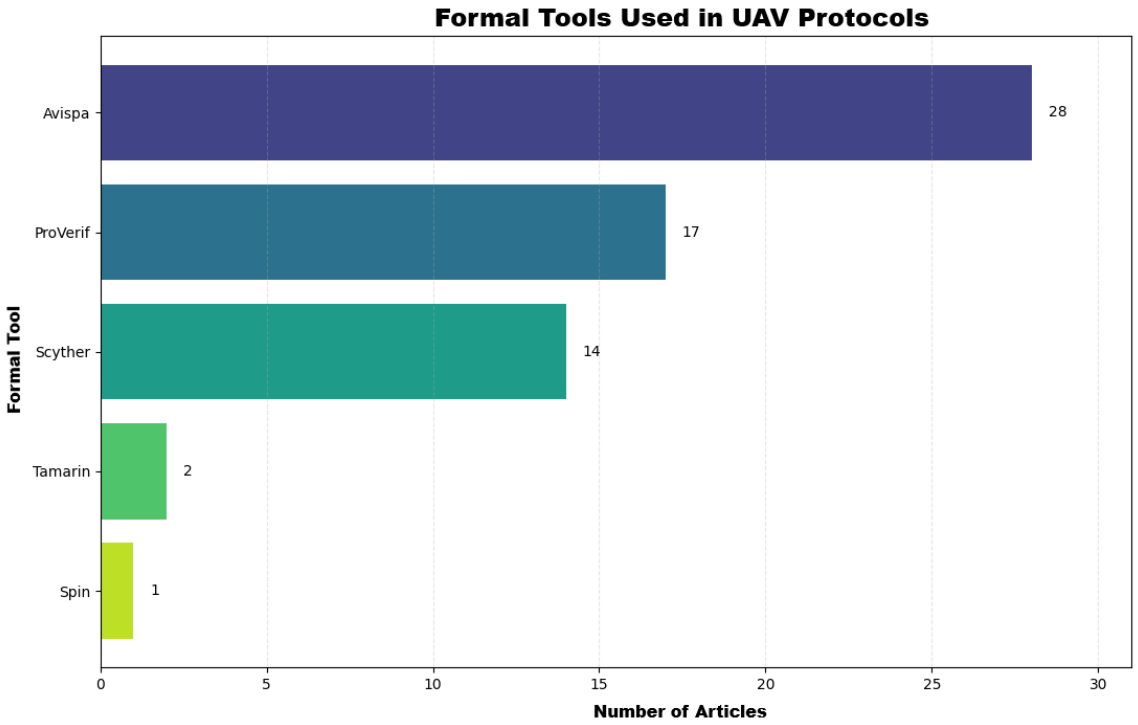

Figure 16 – Frequency of formal verification tools

The use of automated tools plays a fundamental role in modeling and verifying the security of protocols in IoD environments. Among the available options, AVISPA, ProVerif, and Scyther stand out as the most widely used tools in the literature, reflecting their relevance and reliability in supporting formal security analysis.

3.8 Performance Evaluation

Performance evaluation is widely used in the analysis of authentication protocols in IoD, especially given the energy and computational constraints of UAVs. Among the 61 studies, the most recurrent metrics are computational cost (authentication and key agreement execution time) and communication cost (volume of transmitted data).

However, these metrics present limitations regarding their comparability. The lack of standardization in measurement methods, which vary according to environment, hardware, and assumptions, combined with the lack of detail in the accounting of messages and costs, compromises the reproducibility and reliability of the results.

Nevertheless, the reported values allow the identification of strategies aimed at cost reduction, highlighting the strong emphasis of the literature on efficiency. However, this prioritization may occur at the expense of more robust security guarantees.

Table 2 – Descriptive statistics of computational and communication costs

Metric	**Computational (ms)**	**Communication (bits)**
Total number of studies with data	58	60
Mean	36.99	2238.47
Median	7.21	2048
Standard deviation	125.39	1009.91
Minimum	0.07	336
Maximum	902.00	6176

Table 2 presents the descriptive statistics of these metrics. A high variability in the results can be observed, reflecting differences in topology, experimental environment, and cryptographic choices adopted in each proposal. Although the mean computational cost is 36.99 ms, the median of 7.21 ms indicates that most protocols present significantly lower execution times, with extreme values responsible for increasing the mean.

Figure 17 highlights the concentration of communication cost in the range between 1500 and 2500 bits, with a predominance around 2000 bits. This behavior suggests a certain standardization in message sizing, possibly associated with the recurrent use of cryptographic structures based on hash functions and fixed size keys.

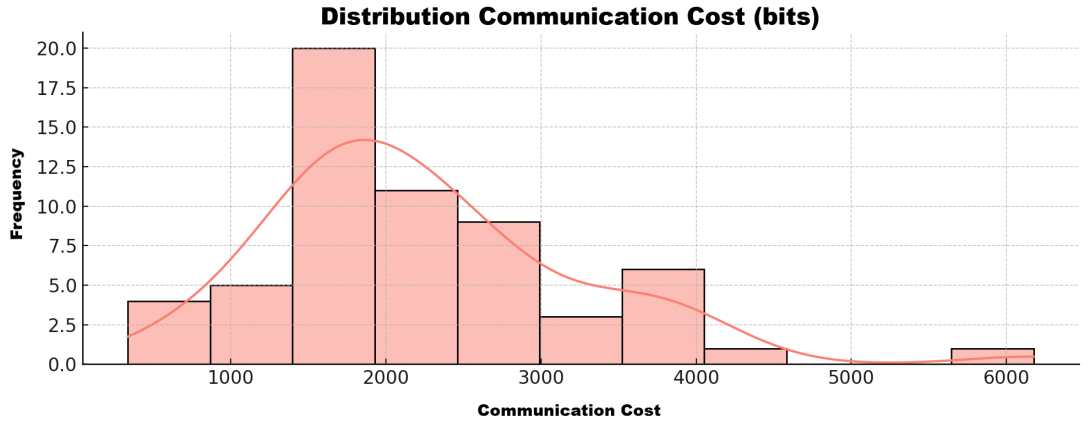

Figure 17 – Distribution of communication cost (in bits) in the analyzed protocols.

Regarding computational cost, Figures 18 and 19 indicate that most protocols are concentrated in the lower-cost range, especially below 25 ms. Nevertheless, the complete distribution shown in Figure 18 includes high-cost outliers, which extend the horizontal scale and make the lower-cost region less visually detailed. To complement this analysis, Figure 19 restricts the visualization to the interval from 0 to 100 ms, allowing a clearer observation of the range where most protocols are located. The difference between the frequency peaks observed in the two figures is a consequence of the different value ranges represented in each plot. While the complete plot highlights the overall distribution and the presence of outliers, the restricted plot emphasizes the concentration of protocols with lower computational costs.

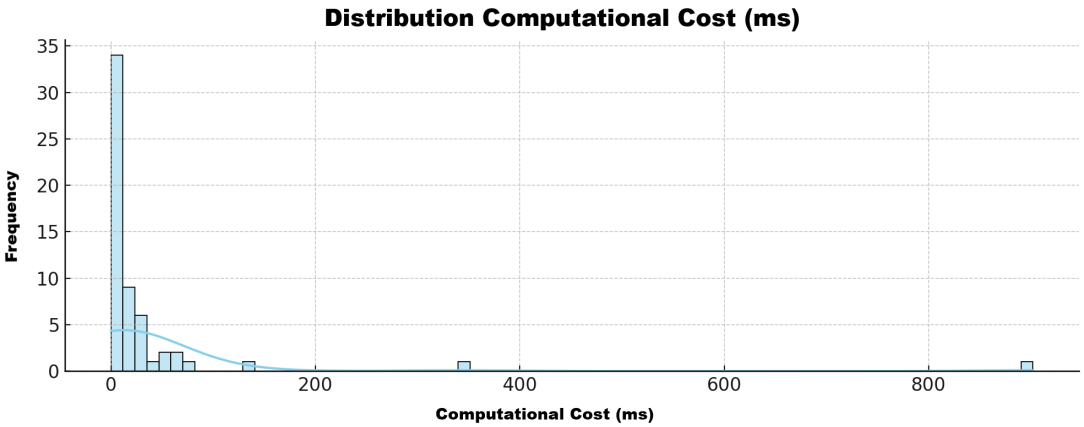

Figure 18 – Distribution of computational cost (in ms) in the analyzed protocols.

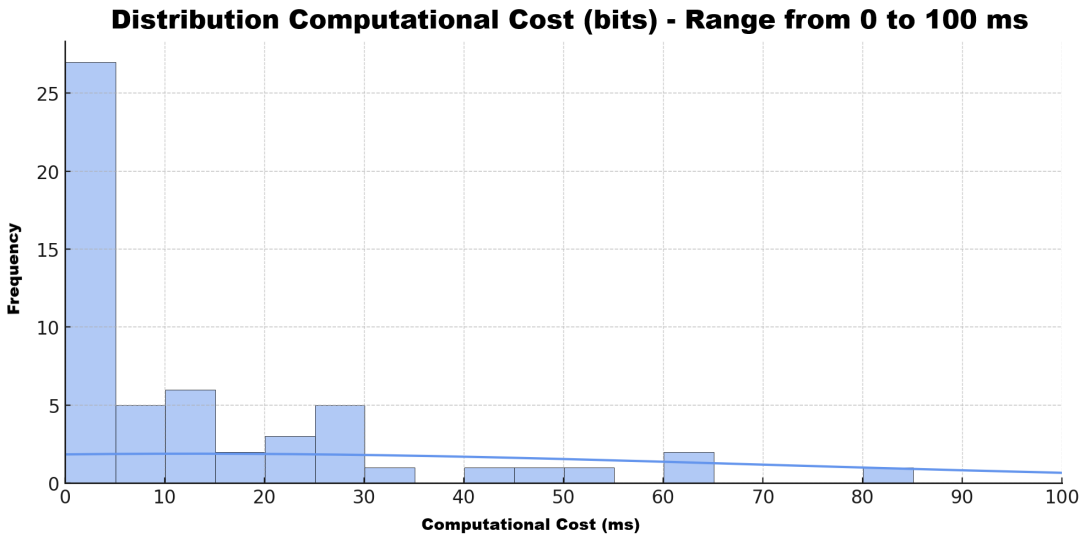

Figure 19 – Distribution of computational cost – Range from 0 to 100 ms.

Overall, the results indicate that the literature seeks to balance security and efficiency, with a predominance of protocols designed to operate within limits compatible with embedded environments. However, the high dispersion of the data highlights a lack of experimental standardization, making direct comparisons between proposals more difficult.

4 Analysis of the PMAP Protocol

This chapter presents an evaluation of the protocol selected for security and performance analysis. The chosen protocol is PMAP [4], proposed with the aim of providing mutual authentication and secure session key establishment among the entities that compose the IoD environment, such as drones and ZSPs.

Several authentication protocols have been proposed for IoD environments, addressing different security requirements, architectures, and operational scenarios [37, 58, 54, 23, 69, 72, 53]. However, these proposals differ in terms of authentication scope, computational cost, communication overhead, and suitability for sensitive applications. PMAP was selected because it is particularly aligned with security-critical IoD scenarios, including military and public security applications, where device legitimacy, authenticated communication, identity privacy, and mission continuity are essential requirements.

The selection of PMAP as the object of analysis is also justified by its alignment with the main characteristics observed in lightweight authentication protocols for IoD environments. The protocol was specifically designed for connected drone networks and combines low-cost cryptographic mechanisms, including hash functions, XOR operations, PUFs, and chaotic systems. In addition, it addresses central security requirements in this context, such as mutual authentication, session key establishment, and identity privacy through the use of dynamic pseudonyms.

Therefore, PMAP provides a representative case for reassessing security and performance aspects identified in the systematic literature review. Its analysis allows this work to examine, in greater detail, how commonly adopted mechanisms and claimed security properties behave when evaluated through complementary formal and informal approaches.

The reassessment presented in this chapter is organized according to the analytical dimensions identified in the systematic literature review. Thus, PMAP is examined in terms of its topology, cryptographic primitives, communication flow, threat model, formal and informal security analysis, and computational and communication costs. This organization provides continuity between the review of authentication protocols for IoD environments and the subsequent case study.

Based on this structure, the protocol's security properties are evaluated using different verification approaches, both formal and informal. This complementary analysis aims to identify possible limitations, assumptions, or gaps that may not have been fully explored in the original work, contributing to a more detailed assessment of the protocol.

Finally, the practical implications of the obtained results are discussed, and possible directions for improvement are presented, focusing on strengthening security properties and adapting the protocol to more demanding scenarios in the context of the IoD.

4.1 Topology

The PMAP protocol is structured around two main actors:

- **Drone**: acts as a field agent, being responsible for mission execution and data collection in external environments.
- **ZSP**: operates as a trusted entity, responsible for drone management, identity validation, and mediating secure communication among them.

Figure 20 – PMAP system model

The topology adopted by PMAP follows an infrastructure-assisted structure, in which drones rely on a trusted supporting entity during the authentication process. This organization simplifies the management of authentication information, since the ZSP maintains data required to validate drones, such as identifiers, pseudonyms, and CRP-related values.

However, if the ZSP becomes unavailable, the establishment of new authenticated sessions may be affected, which characterizes this entity as a potential single point of failure in the protocol architecture.

4.2 Cryptographic Primitives

The PMAP protocol employs five main cryptographic primitives to ensure mutual authentication between the system entities:

- **PUF:** plays a central role in the protocol's security, functioning as a generator of unique secrets [82].
- **Hénon Map:** used to shuffle messages, introducing randomness into the communication process [83]. It is applied in conjunction with the CRP pair provided by the PUF, acting as a lightweight confusion mechanism, although it does not constitute a traditional cryptographic algorithm.
- **Hash Function:** mainly employed in the generation of the final session key and in the construction of pseudonyms. Its function is to ensure irreversibility and uniformity in the derivation of temporary secrets [7].
- **XOR Operation:** used together with the hash function to compose the final session key. Despite its simplicity, this operation ensures the efficient combination of independent values, reinforcing the randomness of the established secret [7].
- **MAC (Message Authentication Code):** used in each step of the message exchange to verify the integrity and authenticity of transmitted data. The use of MACs prevents tampered messages from being accepted by the protocol, acting as a defense against modification and Man-in-the-Middle attacks [7].

4.3 Communication

The PMAP protocol operates in two communication scenarios. In the first, a drone establishes direct communication with the ZSP to achieve mutual authentication and obtain a secure session key. In the second, two drones authenticate each other with the mediation of the ZSP, which acts as a trusted authority to validate identities, preserve anonymity through pseudonyms, and coordinate the update of Challenge-Response Pairs (CRPs). In both scenarios, at the end of the authentication process, a session key is established between the participating entities and is used to protect subsequent communication through a secure channel based on symmetric cryptography.

To enable this process, a prior provisioning phase is required, in which the drone and the ZSP share and store initial information essential to the protocol, such as identifiers, pseudonyms, and CRPs. In this context, the ZSP maintains in its database the real identifiers, pseudonyms, and CRPs associated with each drone, while the drone stores its identifier, the identifier of the ZSP, and the CRP challenge, embedded in its PUF.

Once this foundation is established, the protocol uses random values to ensure session freshness, mitigate replay attacks, and enable key derivation. However, the process involves the repeated transmission of the same set of messages, which may increase communication cost and latency. Additionally, the original article does not specify explicit session management or renewal mechanisms, such as expiration policies or periodic revalidation of the established key, implicitly assuming that updates occur only through the complete re-execution of the protocol. This pattern may be inefficient in IoD environments, where communication is constrained, potentially impacting the scalability of the protocol.

4.3.1 Drone-ZSP Communication (PMAP D2Z)

The communication between a drone D_a with identifier ID_a and the control zone (ZSP) with identifier ID_z is initiated based on the drone's pseudonym PID_a^t, computed from the real identity ID_a, the current PUF response R_a, and the use of a nonce N_a^t to ensure session freshness. Messages are encrypted and authenticated (via MAC), and the ZSP validates the legitimacy of the drone by retrieving the corresponding CRP pair (C_a and R_a). Subsequently, the ZSP generates its own nonce N_z^{t+1}, which, combined with the drone's nonce, enables both the update of credentials (new pseudonym and new CRP) and the derivation of the session key SK_{az}.

At the end of the execution, both parties share the same session key and update their local records. The protocol also adopts redundancy in specific messages (e.g., transmission of M_3 and M_4) as an additional mechanism for verifying tampering, at the cost of increased communication overhead.

Figure 21 presents a simplified overview of the PMAP D2Z message flow. The figure summarizes the main operations performed by the drone and the ZSP.

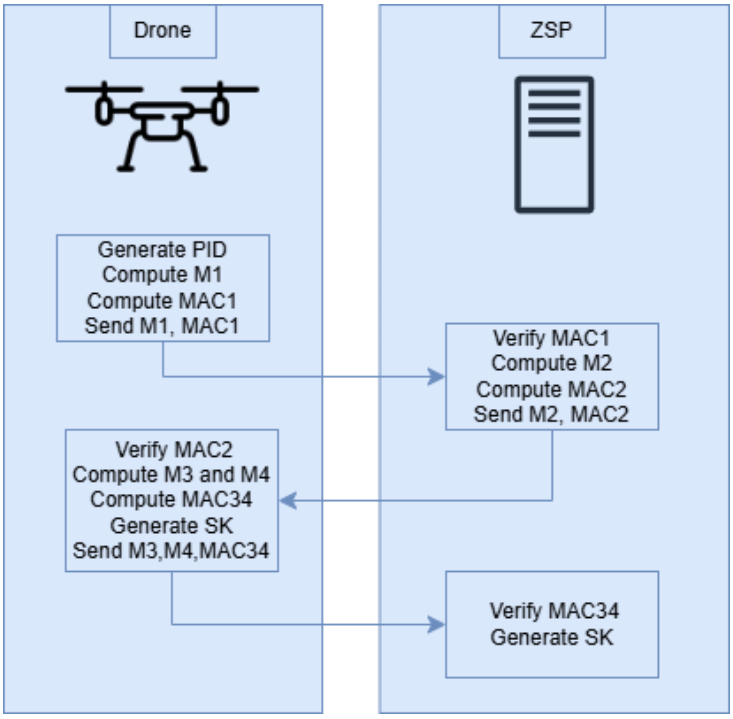

Figure 21 – Simplified overview of the PMAP D2Z message flow.

Consider a drone D_a and a ZSP. The following cryptographic functions are used:

- $H(\cdot)$: one-way cryptographic hash function
- $S(\cdot)$: symmetric encryption function
- $C(\cdot)$: MAC
- $P(\cdot)$: PUF
- $\|$: concatenation operator

The authentication phase of the PMAP protocol in the Drone-ZSP scenario is described in detail below. The process is presented sequentially, highlighting the operations performed by each entity, as well as the messages exchanged throughout the protocol. Each step explicitly specifies the generation of nonces, the verification of message authentication codes, and the update of security parameters, culminating in the establishment of a shared session key between the drone and the ZSP.

Step 1 (Drone → ZSP):

$$PID_a^t = H(ID_a \parallel R_a^t) \tag{4.1}$$

$$N_a^t \leftarrow \text{Random Nonce} \tag{4.2}$$

$$M_1 = S\big(PID_a^t \parallel ID_z \parallel N_a^t\big)_{(C_a^t, R_a^t)} \tag{4.3}$$

$$MAC_1 = C\big(M_1 \parallel N_a^t\big) \tag{4.4}$$

The drone sends:

$$\langle M_1, MAC_1 \rangle$$

Step 2 (ZSP):

- Verifies PID_a^t and retrieves (C_a^t, R_a^t);
- Extracts $N_a^{t'}$ from M_1;
- Computes and verifies:

$$MAC_1' = C\big(M_1 \parallel N_a^{t'}\big) \tag{4.5}$$

Step 3 (ZSP $\rightarrow$ Drone):

$$N_z^{t+1} \leftarrow \text{Random Nonce} \tag{4.6}$$

$$M_2 = S\left(\begin{matrix} PID_a^t \| ID_z \| N_a^{t'} \\ \| N_z^{t+1} \end{matrix}\right)_{(C_a^t, R_a^t)} \tag{4.7}$$

$$MAC_2 = C\big(M_2 \parallel N_a^{t'} \parallel N_z^{t+1}\big) \tag{4.8}$$

ZSP Sends:

$$\langle M_2, MAC_2 \rangle$$

Step 4 (Drone):

- Extracts N_z^{t+1} from M_2;
- Computes and verifies:

$$MAC_2' = C\big(M_2 \parallel N_a^t \parallel N_z^{t+1}\big) \tag{4.9}$$

Step 5 (Drone → ZSP):

$$N_a^{t+1} \leftarrow \text{New nonce} \tag{4.10}$$

$$C_a^{t+1} = S\big(N_z^{t+1} \parallel N_a^{t+1}\big)_{(C_a^t, R_a^t)} \tag{4.11}$$

$$R_a^{t+1} = P\big(C_a^{t+1}\big) \tag{4.12}$$

$$M_3 = S\big(PID_a^t \parallel ID_z \parallel N_z^{t+1} \parallel N_a^{t+1}\big)_{(C_a^t, R_a^t)} \tag{4.13}$$

$$M_4 = S\begin{pmatrix} PID_a^t \| ID_z \| N_z^{t+1} \\ \| N_a^{t+1} \| R_a^{t+1} \end{pmatrix}_{(C_a^t, R_a^t)} \tag{4.14}$$

$$MAC_{34} = C\big(M_3 \parallel M_4 \parallel N_a^{t+1} \parallel R_a^{t+1}\big) \tag{4.15}$$

The session key is calculated as:

$$SK_{az} = H(N_a^{t+1}) \oplus H(N_z^{t+1}) \tag{4.16}$$

The drone sends:

$$\langle M_3, M_4, MAC_{34} \rangle$$

Step 6 (ZSP):

- Extracts $N_a^{t+1'}$ from M_3 and $R_a^{t+1'}$ from M_4;
- Computes and verifies:

$$MAC_{34}' = C\big(M_3 \parallel M_4 \parallel N_a^{t+1'} \parallel R_a^{t+1'}\big) \tag{4.17}$$

$$C_a^{t+1} = S\big(N_z^{t+1'} \parallel N_a^{t+1'}\big)_{(C_a^t, R_a^t)} \tag{4.18}$$

$$PID_a^{t+1} = H\big(ID_a \parallel R_a^{t+1'}\big) \tag{4.19}$$

The session key is calculated as:

$$SK_{az} = H(N_a^{t+1'}) \oplus H(N_z^{t+1}) \tag{4.20}$$

4.3.2 Drone-Drone Communication (PMAP D2D)

In the Drone-Drone scenario, the ZSP acts as a mediator between drone A (D_a) and drone B (D_b). The process begins when D_a requests session initiation by providing the pseudonym of D_b, allowing the ZSP to retrieve the corresponding CRP and validate both entities. However, in the original article, the authors do not specify how D_a obtains the pseudonym of D_b, implicitly assuming that this information is already known beforehand. This lack of detail suggests that obtaining this identifier depends on mechanisms external to the protocol or on prior configuration.

Once this implicit assumption of prior knowledge of the pseudonym is addressed, the ZSP proceeds to distribute specific nonces to each drone, ensuring that each party updates its pseudonym and CRP, and providing the necessary elements for deriving the shared session key.

Figure 22 presents a simplified overview of the PMAP D2D message flow. The figure summarizes the main operations performed by the drone A, drone B and the ZSP.

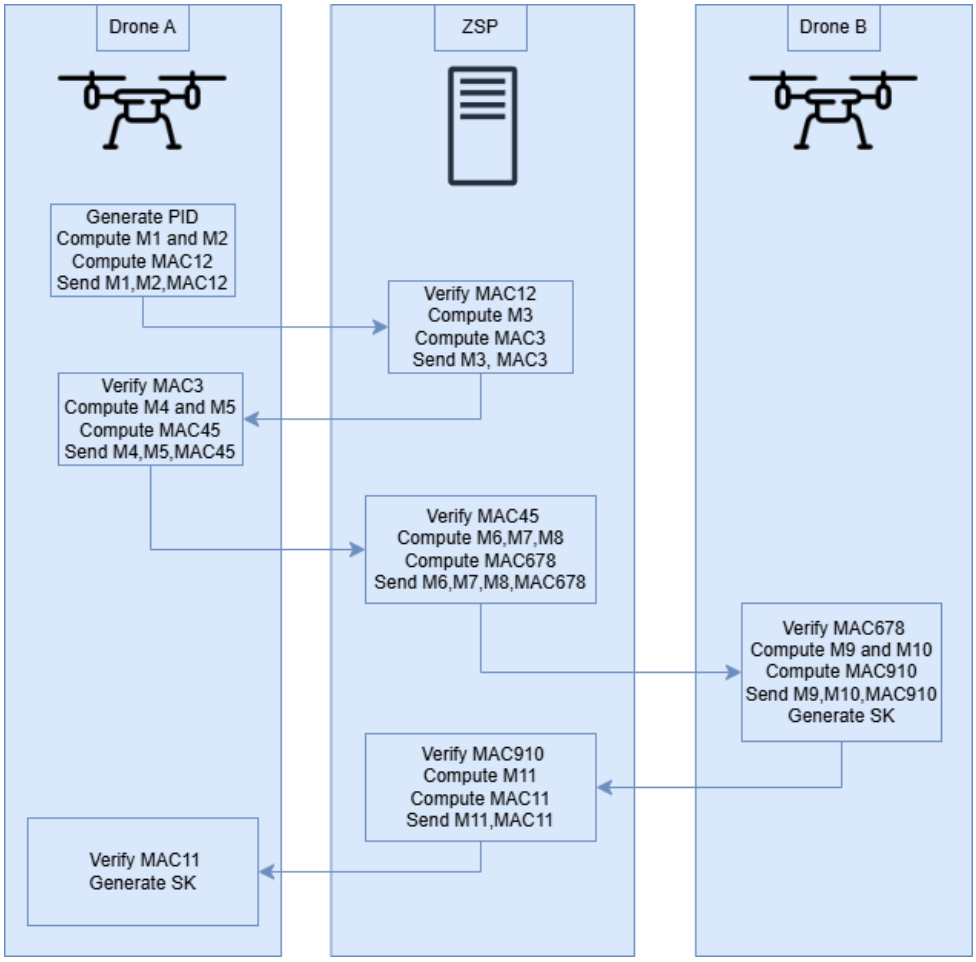

Figure 22 – Simplified overview of the PMAP D2D message flow.

Unlike the D2Z case, the D2D scenario requires a greater number of messages, as it involves three participants and includes additional forwarding and validation steps. Nevertheless, the core logic remains the same: mutual authentication, credential updates, and session key derivation through the combination of nonces and hash functions, with redundancy in certain messages serving as a mechanism for robustness against tampering.

The complete flow of the PMAP protocol in the Drone-Drone scenario is presented below, organized into sequential steps that detail the interactions among Drone A, the ZSP, and Drone B, as well as the updates of pseudonyms, CRP pairs, and the establishment of the shared session key.

Step 1 (Drone A → ZSP):

$$PID_a^t = H(ID_a \parallel R_a^t) \tag{4.21}$$

$$N_a^t \leftarrow \text{Random Nonce} \tag{4.22}$$

$$M_1 = S\big(PID_a^t \parallel ID_z \parallel N_a^t\big)_{(C_a^t, R_a^t)} \tag{4.23}$$

$$M_2 = S\big(PID_a^t \parallel ID_z \parallel N_a^t \parallel PID_b^t\big)_{(C_a^t, R_a^t)} \tag{4.24}$$

$$MAC_{12} = C\big(M_1 \parallel M_2 \parallel N_a^t \parallel PID_b^t\big) \tag{4.25}$$

The drone sends:

$$\langle M_1, M_2, MAC_{12} \rangle$$

Step 2 (ZSP):

- Verifies PID_a^t and retrieves (C_a^t, R_a^t);
- Extracts $N_a^{t'}$ from M_1;
- Computes and verifies:

$$MAC'_{12} = C\big(M_1 \parallel M_2 \parallel N_a^{t'} \parallel PID_b^{t'}\big) \tag{4.26}$$

Step 3 (ZSP → Drone A):

$$N_{az}^{t+1},\ N_{bz}^{t+1} \leftarrow \text{Random Nonces} \tag{4.27}$$

$$M_3 = S\begin{pmatrix} PID_a^t \| ID_z \| PID_b^{t'} \\ \| N_a^{t'} \| N_{az}^{t+1} \end{pmatrix}_{(C_a^t, R_a^t)} \tag{4.28}$$

$$MAC_3 = C\big(M_3 \parallel N_a^{t'} \parallel N_{az}^{t+1}\big) \tag{4.29}$$

The ZSP sends:

$$\langle M_3, MAC_3 \rangle$$

Step 4 (Drone A)

- Extracts N_{az}^{t+1} from M_3;
- Computes and verifies:

$$N_a^{t+1} \leftarrow \text{New Nonce} \tag{4.30}$$

$$C_a^{t+1} = S\big(N_{az}^{t+1} \parallel N_a^{t+1}\big) \tag{4.31}$$

$$R_a^{t+1} = P\big(C_a^{t+1}\big) \tag{4.32}$$

$$PID_a^{t+1} = H\big(ID_a \parallel R_a^{t+1}\big) \tag{4.33}$$

Step 5 (Drone A $\rightarrow$ ZSP):

$$M_4 = S\begin{pmatrix} PID_a^t \| ID_z \| PID_b^{t'} \\ \| N_{az}^{t+1} \| N_a^{t+1} \end{pmatrix}_{(C_a^t, R_a^t)} \tag{4.34}$$

$$M_5 = S\begin{pmatrix} PID_a^t \| ID_z \| PID_b^{t'} \\ \| N_{az}^{t+1} \| N_a^{t+1} \\ \| R_a^{t+1} \end{pmatrix}_{(C_a^t, R_a^t)} \tag{4.35}$$

$$MAC_{45} = C\big(M_4 \parallel M_5 \parallel N_a^{t+1} \parallel R_a^{t+1}\big) \tag{4.36}$$

The drone A sends:

$$\langle M_4, M_5, MAC_{45} \rangle$$

Step 6 (ZSP):

- Extracts $N_a^{t+1'}$ from M_4 and $R_a^{t+1'}$ from M_5;
- Computes and verifies:

$$MAC'_{45} = C\big(M_4 \parallel M_5 \parallel N_a^{t+1'} \parallel R_a^{t+1'}\big) \tag{4.37}$$

Step 7 (ZSP → Drone B):

$$M_6 = S\big(PID_b^t \parallel ID_z \parallel N_{bz}^{t+1}\big) \tag{4.38}$$

$$M_7 = S\big(PID_b^t \parallel ID_z \parallel N_{bz}^{t+1} \parallel N_a^{t+1'}\big)_{(C_b^t, R_b^t)} \tag{4.39}$$

$$M_8 = S\begin{pmatrix} PID_b^t \| ID_z \| N_{bz}^{t+1} \\ \| N_a^{t+1'} \| PID_a^t \end{pmatrix}_{(C_b^t, R_b^t)} \tag{4.40}$$

$$MAC_{678} = C\begin{pmatrix} M_6 \| M_7 \| M_8 \\ \| N_{bz}^{t+1} \| N_a^{t+1'} \\ \| PID_a^t \end{pmatrix} \tag{4.41}$$

The ZSP sends:

$$\langle M_6, M_7, M_8, MAC_{678} \rangle$$

Step 8 (Drone B):

- Extracts $N_b z^{t+1'}$ from M_6, $N_a^{t+1''}$ from M_7 and $PID_i^{t'}$ from M_8;
- Computes and verifies:

$$MAC'_{678} = C\begin{pmatrix} M_6 \| M_7 \| M_8 \\ \| N_{bz}^{t+1'} \| N_a^{t+1''} \\ \| PID_a^{t'} \end{pmatrix} \tag{4.42}$$

Step 9 (Drone B → ZSP):

$$N_b^{t+1} \leftarrow \text{New Nonce} \tag{4.43}$$

$$C_b^{t+1} = S\big(N_{bz}^{t+1} \parallel N_b^{t+1}\big)_{(C_b^t, R_b^t)} \tag{4.44}$$

$$R_b^{t+1} = P\big(C_b^{t+1}\big) \tag{4.45}$$

$$PID_b^{t+1} = H\big(ID_b \parallel R_b^{t+1}\big) \tag{4.46}$$

$$M_9 = S\begin{pmatrix} PID_b^t \| ID_z \| PID_a^t \\ \| N_{bz}^{t+1} \| N_b^{t+1} \end{pmatrix}_{(C_b^t, R_b^t)} \tag{4.47}$$

$$M_{10} = S\begin{pmatrix} PID_b^t \| ID_z \| PID_a^t \\ \| N_{bz}^{t+1} \| N_b^{t+1} \\ \| R_b^{t+1} \end{pmatrix}_{(C_b^t, R_b^t)} \tag{4.48}$$

$$MAC_{910} = C\big(M_9 \parallel M_{10} \parallel N_b^{t+1} \parallel R_b^{t+1}\big) \tag{4.49}$$

The session key is calculated as:

$$SK_{ba} = H(N_b^{t+1}) \oplus H(N_a^{t+1'}) \tag{4.50}$$

The drone B sends:

$$\langle M_9, M_{10}, MAC_{910} \rangle$$

Step 10 (ZSP):

- Extracts $N_b^{t+1'}$ from M_9 and $R_b^{t+1'}$ from M_{10};
- Computes and verifies:

$$MAC'_{910} = C\big(M_9 \parallel M_{10} \parallel N_b^{t+1'} \parallel R_b^{t+1'}\big) \tag{4.51}$$

Step 11 (ZSP $\rightarrow$ Drone A):

$$C_b^{t+1} = S\big(N_{bz}^{t+1'} \parallel N_b^{t+1}\big)_{(C_b^t, R_b^t)} \tag{4.52}$$

$$PID_b^{t+1} = H\big(ID_b \parallel R_b^{t+1'}\big) \tag{4.53}$$

$$M_{11} = S\begin{pmatrix} PID_b^t \| ID_z \| PID_a^t \\ \| N_a^{t+1} \| N_b^{t+1} \end{pmatrix}_{(C_b^t, R_b^t)} \tag{4.54}$$

$$MAC_{11} = C\big(M_{11} \parallel N_a^{t+1} \parallel R_b^{t+1}\big) \tag{4.55}$$

The ZSP sends:

$$\langle M_{11}, MAC_{11} \rangle$$

Step 12 (Drone A)

- Extracts $N_b^{t+1'}$ from M_11;
- Computes and verifies:

$$MAC'_{11} = C\big(M_{11} \parallel N_a^{t+1} \parallel R_b^{t+1'}\big) \tag{4.56}$$

The session key is calculated as:

$$SK_{ab} = H(N_a^{t+1}) \oplus H(N_b^{t+1}) \tag{4.57}$$

4.4 Threat Model

The PMAP protocol adopts an adversary model widely used in wireless communication systems, assuming that all channels are insecure and subject to interception. In this scenario, an adversary can eavesdrop, duplicate, modify, reorder, delay, or retransmit messages exchanged between drones and ZSPs.

The model also considers the possibility of physical capture of drones. However, the original work assumes that each device is equipped with an ideal and noise resistant PUF. Under this assumption, any attempt to probe or manipulate the integrated circuit would irreversibly alter the challenge–response mapping, preventing the extraction of internal secrets. Thus, the security of the protocol strongly depends on the assumption of the hardware's physical tamper resistance.

However, this assumption does not explicitly cover more realistic scenarios of partial physical compromise, in which an adversary may gain access to the captured device and extract data stored in non-volatile memory, intermediate states, or auxiliary parameters used during protocol execution.

Therefore, in this work, the threat model is extended to consider the possibility of partial physical data extraction from a compromised drone, while still preserving the assumed tamper resistance properties of the PUF. Based on this scenario, the impact of such partial exposure on authentication and session key secrecy properties in the PMAP protocol is evaluated.

The adversary's objective remains to illegitimately authenticate itself to a ZSP or another drone, as well as to interfere with communication in order to cause strategic consequences, such as route redirection or unauthorized control of aerial vehicles. To mitigate such risks, PMAP was designed to satisfy requirements such as authenticity, integrity, confidentiality, secure session key establishment, and resistance to different classes of attacks [6].

4.5 Formal Security Analysis

As observed in the systematic literature review, formal security evaluation in IoD authentication protocols is commonly supported by two complementary approaches: proof-based models and automated verification tools. Among the identified formal models, ROR, BAN, and ROM appear as recurrent methods, while AVISPA, ProVerif, and Scyther stand out among the most frequently used automated tools.

Based on this observation, PMAP is evaluated in this section considering the ROR model [10], which is used to analyze session key security. In addition, the automated tools AVISPA [13], Scyther [15], and ProVerif [14] are employed to support the verification of authentication and secrecy properties under complementary formal specifications.

4.5.1 ROR Model

The RoR model [10] is used to verify the security of the session key established by the PMAP protocol. In this model, security is evaluated based on the ability of an adversary to distinguish whether a given session key is the real key derived from the protocol execution or a completely random value.

In general, the RoR model defines an experiment in which the adversary $\mathcal{A}$ receives a session key and must decide whether it is real or random. If the adversary cannot distinguish between them, its behavior is no better than a random guess.

The adversary's advantage is formally defined as:

$$\mathrm{Adv}_{\mathcal{A}}^{PMAP}(plt) = \left|2\Pr[\mathcal{A} \text{ succeeds }] - 1\right|.$$

If the adversary merely guesses, its probability of success is $\frac{1}{2}$, resulting in zero advantage. On the other hand, if it can perfectly distinguish between the two cases, its probability of success will be 1, resulting in a maximum advantage of 1. Thus, we have:

$$0 \leq \mathrm{Adv}_{\mathcal{A}}^{PMAP}(plt) \leq 1.$$

Therefore, the protocol is considered secure when this advantage is negligible, that is, when the adversary cannot distinguish the real key from a random value with probability significantly greater than a random guess.

In the context of this protocol, the participants can be represented as $Part_{Drone}$, corresponding to the drone acting as the field entity, and $Part_{ZSP}$, representing the trusted ZSP.

The adversary $\mathcal{A}$ follows the DY model [5], assuming full control over the public communication channel. Thus, $\mathcal{A}$ is capable of intercepting, modifying, delaying, reordering, or deleting the messages exchanged between the protocol participants.

In the RoR model, the adversary is granted access to queries that represent different attack capabilities. The EXECUTE query allows observing honest executions of the protocol, while the SEND query enables modifying or injecting messages. The CORRUPT query grants access to secret information stored in the drone. Finally, the TEST query provides the adversary with a session key that may be either real or random, and is used to define its advantage in the model.

Theorem 01 The value $\mathrm{Adv}_{\mathcal{A}}^{PMAP}(plt)$ represents the advantage of a polynomial time adversary $\mathcal{A}$ in distinguishing the real session key from a random value in the PMAP protocol under the RoR model. This advantage is bounded by the sum of three terms that reflect the security of the primitives used in the protocol.

$$\begin{aligned} \mathrm{Adv}_{\mathcal{A}}^{PMAP}(plt) \leq\ & 2\mathrm{Adv}_{\mathcal{A}}^{MAC}(plt) \\ & + 2\mathrm{Adv}_{\mathcal{A}}^{PRP}(plt) \\ & + \frac{q_{\mathrm{CRP}}^2}{|\mathrm{PUF}|} \end{aligned} \tag{4.58}$$

The term $2\mathrm{Adv}_{\mathcal{A}}^{MAC}(plt)$ is related to the security of the message authentication mechanism (MAC), capturing the adversary's ability to produce or distinguish valid authenticated messages. This term directly reflects the MAC's resistance to existential forgery attacks.

The term $2\mathrm{Adv}_{\mathcal{A}}^{PRP}(plt)$ corresponds to the adversary's advantage in distinguishing the shuffling function employed in the protocol, based on a pseudorandom construction, from a truly random permutation. This term is associated with the indistinguishability property of the primitive used in the shuffling process.

Finally, the term $\frac{q_{\mathrm{CRP}}^2}{|\mathrm{PUF}|}$ represents the probabilistic contribution associated with the number of queries performed over the finite space of PUF challenge–response pairs, reflecting the impact of the indirect reuse of this space across protocol executions.

Proof According to the RoR security model, five games G_n (with $n = 0, 1, 2, 3, 4$) are defined. Thus, the probability of success of an adversary $\mathcal{A}$ in distinguishing the session key is expressed by $\mathrm{Adv}_{\mathcal{A}}^{PMAP}(plt)$.

Game G0 In this game, which corresponds to the real attack scenario, all queries are answered faithfully according to the protocol specifications. Based on the definition of security in the RoR model, the advantage of a polynomial time adversary is given by:

$$Adv_{\mathcal{A}}^{PMAP}(plt) = |2 \cdot Adv_{\mathcal{A}}^{G0}(plt) - 1| \tag{4.59}$$

Game G1 In this game, the adversary $\mathcal{A}$ can only capture messages transmitted over the public channel by issuing EXECUTE queries. Based on the observed transcripts, the

goal of $\mathcal{A}$ is to derive the session key SK, constructed from the nonces generated in the previous steps of the protocol. Since $\mathcal{A}$ does not have knowledge of the shuffling performed by the Hénon map nor of the PUF challenge–response pair (C_i, R_i), it cannot compute SK. Thus, games $G0$ and $G1$ are indistinguishable:

$$Adv_{\mathcal{A}}^{G0}(plt) = Adv_{\mathcal{A}}^{G1}(plt) \tag{4.60}$$

Game G2 In this game, the possibility of forging messages along with their corresponding authentication codes is considered. The adversary's probability of success is bounded by the security of the MAC. Therefore, the difference between $G1$ and $G2$ is given by:

$$|Adv_{\mathcal{A}}^{G1}(plt) - Adv_{\mathcal{A}}^{G2}(plt)| \;\leq\; Adv_{\mathcal{A}}^{MAC}(plt). \tag{4.61}$$

Game G3 In this game, we replace the shuffling function $S(\cdot)_{(C_i,R_i)}$ based on the Hénon map with a truly random permutation $\pi(\cdot)$ defined over the same domain, restricted to fresh sessions in which the pair (C_i, R_i) remains secret (i.e., not compromised). For the adversary $\mathcal{A}$, any attempt to extract nonces from the transcripts reduces to distinguishing S from π. Thus, the difference between $G2$ and $G3$ is bounded by the advantage of distinguishing the pseudorandom permutation:

$$|Adv_{\mathcal{A}}^{G2}(plt) - Adv_{\mathcal{A}}^{G3}(plt)| \;\leq\; Adv_{\mathcal{A}}^{PRP}(plt). \tag{4.62}$$

Game G4. In the final game, the adversary $\mathcal{A}$ issues a CORRUPT query to the drone, obtaining the parameters stored in memory $\{ID_i, C_i\}$. If the corruption occurs before or during the target session, the session is no longer considered *fresh*, and consequently, the TEST query returns null. On the other hand, if the corruption occurs only after the session has been completed (thus preserving freshness), the knowledge of $\{ID_i, C_i\}$ is not sufficient to reconstruct the session key, since it would still be necessary to obtain the value $R_i = P(C_i)$ in order to reverse the shuffling and recover the nonces used in key derivation.

Assuming that the PUF behaves as an unpredictable physical function over a finite space of CRPs, the adversary cannot predict the corresponding response to unseen challenges beyond negligible probability. Thus, any additional advantage would depend on obtaining repetitions or relations among challenges within the observed CRP space.

Applying the birthday bound [84] to the PUF challenge–response pair space, the difference in advantage between the games can be bounded by:

$$\left| Adv_{\mathcal{A}}^{G3}(plt) - Adv_{\mathcal{A}}^{G4}(plt) \right| \;\leq\; \frac{q_{\mathrm{CRP}}^2}{2\,|\mathrm{PUF}|}\,. \tag{4.63}$$

The term $\frac{q_{\text{CRP}}^2}{2|\text{PUF}|}$ arises from the birthday bound and represents an upper bound on the probability that the adversary observes repetitions of challenges or exploits statistical dependencies within the PUF CRP space after q_{CRP} queries, considering a finite space of size $|\text{PUF}|$.

After performing all allowed queries, the adversary $\mathcal{A}$ issues the TEST query, whose goal is to determine whether the returned session key corresponds to the real protocol key or to a random value.

In game $G4$, all information that could assist the adversary has already been replaced by independent or idealized values. As a result, the session key returned in the TEST query is computationally indistinguishable from a random value from the perspective of any polynomial time adversary.

Thus, $\mathcal{A}$ does not have any efficient strategy better than a random guess, and its probability of success is approximately $\frac{1}{2}$. Therefore, we have:

$$Adv_{\mathcal{A}}^{G4}(plt) = \frac{1}{2}. \tag{4.64}$$

Eqs. (G0-G4) together yield the following derivations:

$$\begin{aligned}
\frac{1}{2}Adv_{\mathcal{A}}^{PMAP}(plt) &= \left|Adv_{\mathcal{A}}^{G0}(plt) - \tfrac{1}{2}\right| \\
&= \left|Adv_{\mathcal{A}}^{G0}(plt) - Adv_{\mathcal{A}}^{G4}(plt)\right| \\
&\leq |Adv_{\mathcal{A}}^{G0}(plt) - Adv_{\mathcal{A}}^{G1}(plt)| + |Adv_{\mathcal{A}}^{G1}(plt) - Adv_{\mathcal{A}}^{G2}(plt)| \\
&\quad + |Adv_{\mathcal{A}}^{G2}(plt) - Adv_{\mathcal{A}}^{G3}(plt)| + |Adv_{\mathcal{A}}^{G3}(plt) - Adv_{\mathcal{A}}^{G4}(plt)| \\
&\leq Adv^{MAC}(plt) + Adv_{\mathcal{A}}^{PRP}(plt) + \frac{q_{\text{CRP}}^2}{2\,|\text{PUF}|}.
\end{aligned} \tag{4.65}$$

Thus, the desired result is obtained by multiplying both sides of the above expression by 2:

$$\begin{aligned}
\text{Adv}_{\mathcal{A}}^{PMAP}(plt) \;\leq\; & 2\text{Adv}_{\mathcal{A}}^{MAC}(plt) \\
& + 2\text{Adv}_{\mathcal{A}}^{PRP}(plt) \\
& + \frac{q_{\text{CRP}}^2}{|\text{PUF}|}
\end{aligned} \tag{4.66}$$

Inequality 4.66 establishes an upper bound on the advantage of a polynomial time adversary in distinguishing the real session key from a random value. Thus, assuming that the MAC is existentially unforgeable, the shuffling mechanism is indistinguishable from a truly random permutation, and the PUF's CRP space is sufficiently large, the adversary's advantage remains negligible.

Consequently, it is concluded that the PMAP protocol is secure under the RoR model given the stated assumptions, since any efficient adversary capable of distinguishing the session key would imply the violation of at least one of these assumptions.

4.5.2 Automated Tools

Automated tools play a fundamental role in identifying vulnerabilities and validating security properties in cryptographic protocols. In the original PMAP paper, the formal evaluation was conducted exclusively using the AVISPA tool, in which the protocol was modeled and yielded results classified as secure under the assumptions of the adopted model.

In this work, a more comprehensive reassessment of PMAP is proposed, expanding the scope of the formal analysis. In addition to reproducing the experiments in AVISPA for consistency and comparison purposes, the protocol was also modeled and analyzed using two additional tools widely adopted in the formal verification of security protocols: Scyther Tool and ProVerif.

This complementary approach enables the protocol to be analyzed under different internal verification mechanisms, distinct state exploration algorithms, and diverse modeling capabilities, thereby increasing the reliability of the evaluation and reducing dependence on a single analysis environment.

4.5.2.1 AVISPA

The AVISPA tool [13] was employed to formally verify the PMAP protocol under the DY symbolic model. The specification was written in HLPSL (High-Level Protocol Specification Language), explicitly modeling the roles of the participating entities, the message flow, and the intruder's capabilities.

The analysis was carried out using the OFMC and CL-AtSe back-ends. OFMC (On-the-Fly Model Checker) performs verification through dynamic state exploration, simulating possible protocol executions in the presence of an adversary. In contrast, CL-AtSe (Constraint-Logic-based Attack Searcher) adopts a constraint based approach, translating the security problem into a set of logical constraints.

The use of these back-ends enables the automatic verification of security properties such as mutual authentication and session key confidentiality, considering both protocol scenarios: $PMAP^{D2Z}$ and $PMAP^{D2D}$. The experiments were conducted in an Ubuntu 10.04 environment.

The generated reports indicate that, under the assumptions of the adopted symbolic model and considering a bounded number of sessions, no vulnerabilities were identified

in the analyzed executions. This result suggests that the protocol preserves the specified security properties against classical interception and message manipulation attacks over a public channel.

Figures 23 and 24 present the results obtained for each scenario.

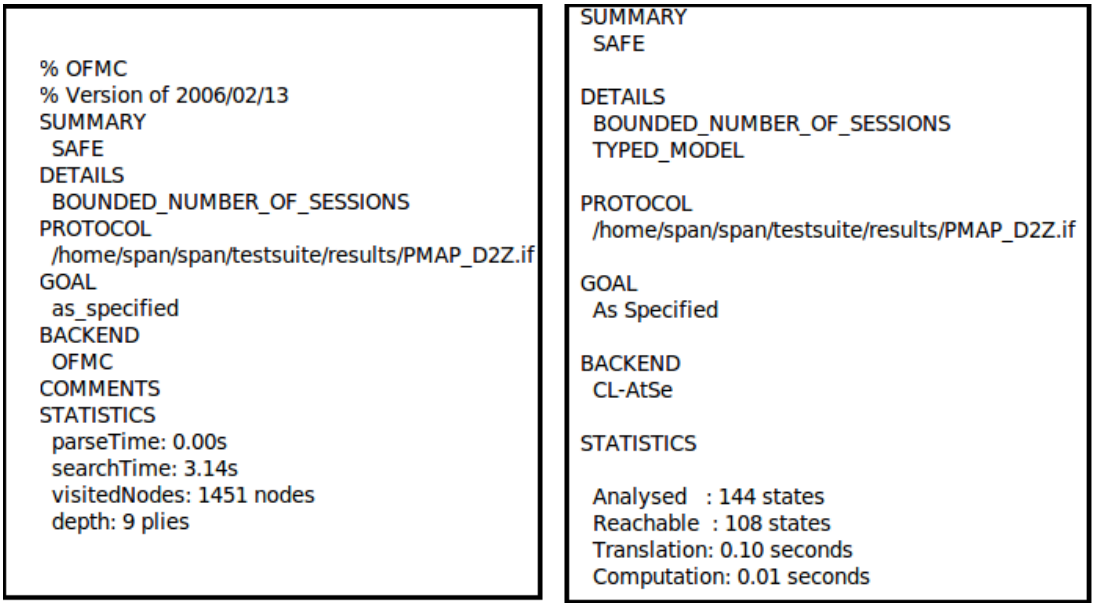

Figure 23 – Verification results of the $PMAP^{D2Z}$ protocol using the AVISPA tool.

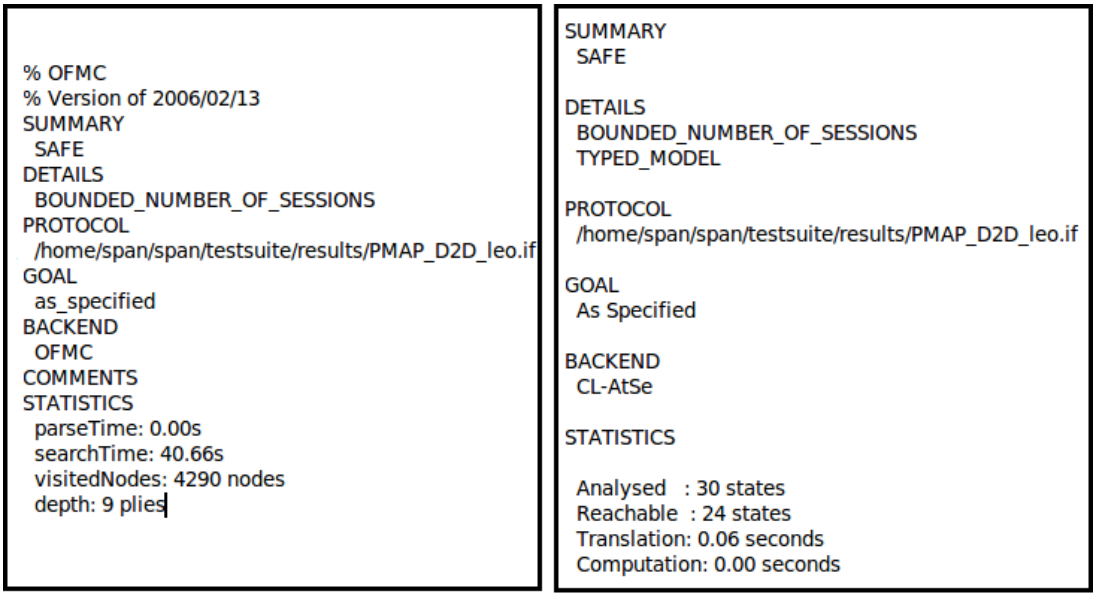

Figure 24 – Verification results of the $PMAP^{D2D}$ protocol using the AVISPA tool.

4.5.2.2 SCYTHER

The Scyther tool [15] was used for the formal verification of the PMAP protocol under the DY symbolic model, in which the adversary has full control over the communication channel and is capable of intercepting, modifying, and injecting messages.

The protocol was modeled using SPDL (Symbolic Protocol Description Language), where the roles of the drone and the ZSP were defined, including the message flow, cryptographic parameters, and the security properties to be verified, such as confidentiality, authentication, and freshness. These properties ensure, respectively, that sensitive information is not disclosed to unauthorized entities, that the involved parties are properly authenticated, and that sessions are not improperly reused.

The experiments were conducted in an Ubuntu 16.04 environment, considering multiple concurrent sessions in order to evaluate the protocol under conditions closer to real-world applications. The automated analysis performed by the tool did not identify any violations of the specified security properties in the $PMAP^{D2Z}$ and $PMAP^{D2D}$ scenarios.

Additionally, the results indicate that the protocol preserves its security properties even in the presence of an active adversary, reinforcing its robustness against attacks in the symbolic model. Figures 25 and 26 present the reports generated by the tool, highlighting the absence of attacks in the analyzed executions.

Scyther results : verify

Claim				Status	Comments
pmap_d2z	D	pmap_d2z,D1	Secret Nd	Ok	No attacks within bounds.
		pmap_d2z,D2	Secret Ns	Ok	No attacks within bounds.
		pmap_d2z,D3	Secret Rd	Ok	No attacks within bounds.
		pmap_d2z,D4	Secret Nd2	Ok	No attacks within bounds.
		pmap_d2z,D5	Niagree	Ok	No attacks within bounds.
		pmap_d2z,D6	Nisynch	Ok	No attacks within bounds.
		pmap_d2z,D7	Alive	Ok	No attacks within bounds.
		pmap_d2z,D8	Weakagree	Ok	No attacks within bounds.
	Z	pmap_d2z,Z1	Secret Ns	Ok	No attacks within bounds.
		pmap_d2z,Z2	Secret Nd	Ok	No attacks within bounds.
		pmap_d2z,Z3	Secret Nd2	Ok	No attacks within bounds.
		pmap_d2z,Z4	Secret Rd	Ok	No attacks within bounds.
		pmap_d2z,Z5	Niagree	Ok	No attacks within bounds.
		pmap_d2z,Z6	Nisynch	Ok	No attacks within bounds.
		pmap_d2z,Z7	Alive	Ok	No attacks within bounds.
		pmap_d2z,Z8	Weakagree	Ok	No attacks within bounds.

Done.

Figure 25 – Verification results of the $PMAP^{D2Z}$ protocol using the Scyther tool.

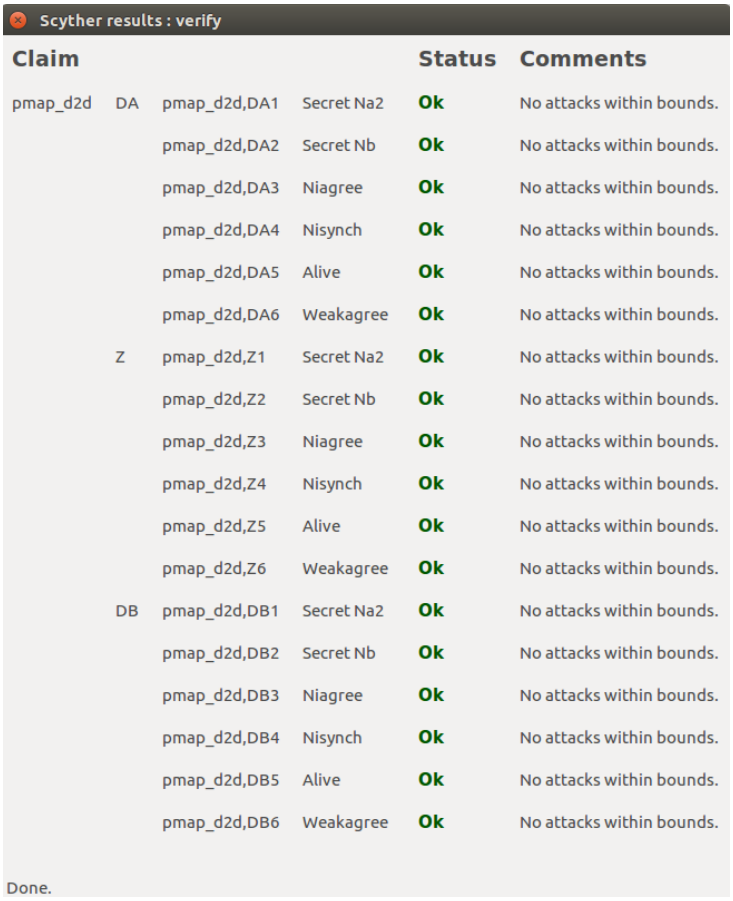

Figure 26 – Verification results of the $PMAP^{D2D}$ protocol using the Scyther tool.

4.5.2.3 PROVERIF

The ProVerif tool [14] was employed to formally verify the PMAP protocol under the DY symbolic model, with support for multiple sessions and concurrent executions. The protocol was modeled using the applied pi-calculus, enabling the automatic analysis of security properties through its translation into logical clauses.

Properties related to session key secrecy and mutual authentication between the parties were specified, covering both $PMAP^{D2Z}$ and $PMAP^{D2D}$ scenarios. The experiments were conducted in an Ubuntu 20.04 environment.

The results indicate that, under the assumptions of the adopted model, no violations of the specified properties were identified. Therefore, the protocol preserves secrecy and authenticity even in the presence of multiple concurrent sessions.

Figures 27 and 28 present the reports generated by the tool.

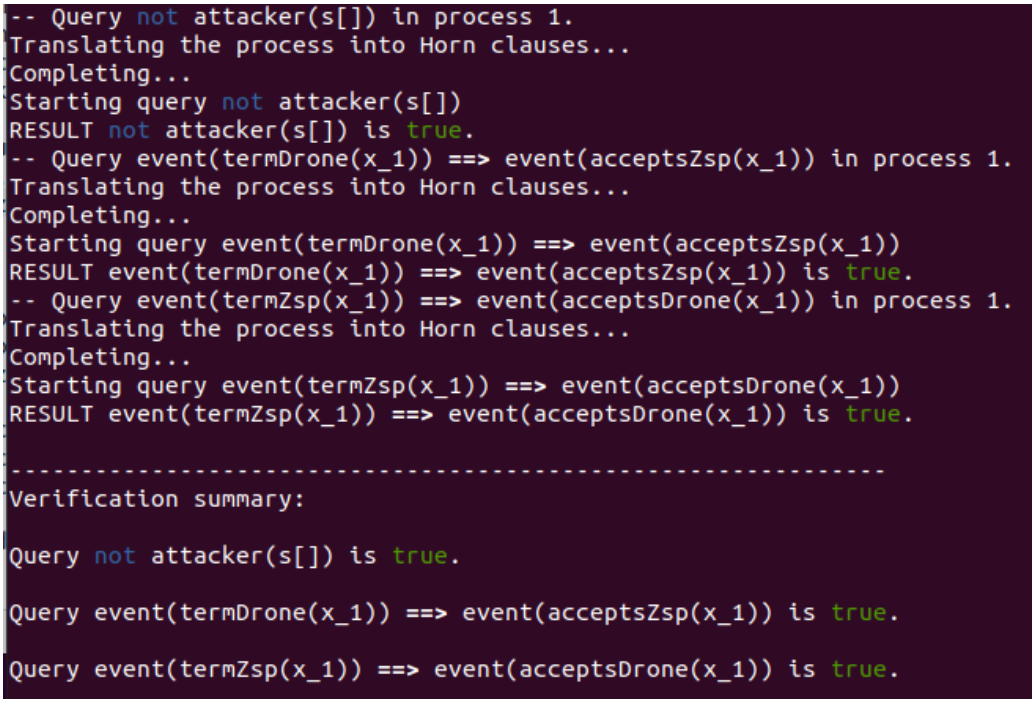

Figure 27 – Verification results of the $PMAP^{D2Z}$ protocol using the ProVerif tool.

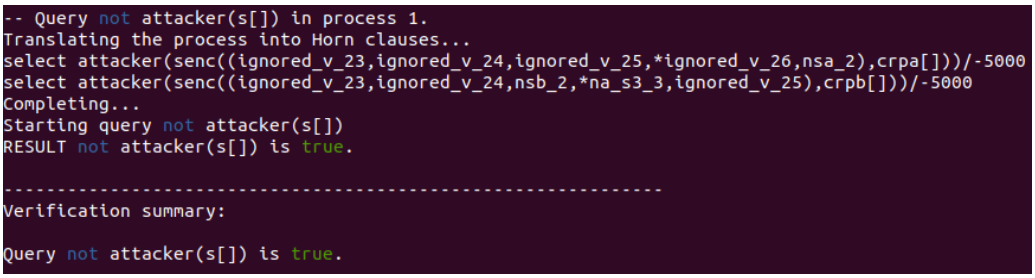

Figure 28 – Verification results of the $PMAP^{D2D}$ protocol using the ProVerif tool.

4.6 Informal Security Analysis

In this section, an informal analysis of PMAP is conducted to complement the assessment of the protocol with respect to security requirements commonly discussed in IoD environments [19, 20, 36, 8].

1. Anonymity: In PMAP, each drone ID_i uses a dynamic pseudonym $PID_i^t = H(ID_i \| R_i^t)$ for communication. Thus, the real identity of the device is not directly transmitted over the public channel and is known only to the ZSP, which maintains the complete database of CRPs. Therefore, even if an external adversary collects mul-

tiple protocol transcripts, it is not possible to directly map the pseudonym PID_i^t to the real identity ID_i.

However, the value ID_i remains stored in the drone's memory. Consequently, in a physical capture scenario, an adversary may extract this information and compromise the drone's anonymity. Therefore, PMAP provides anonymity against external adversaries observing the communication channel, but does not guarantee protection against physical capture attacks.

2. Untraceability: Since the pseudonym PID_i^t changes in each session (depending on new PUF values and nonces), different sessions of the same drone cannot be easily correlated by an external adversary. As a result, the protocol prevents direct tracking of the same entity across multiple interactions. Thus, PMAP satisfies untraceability in the presence of external adversaries.

3. Perfect Forward Secrecy: Perfect Forward Secrecy ensures that even if long term secrets are compromised in the future, previously established session keys cannot be reconstructed. Protocols that provide PFS typically rely on ephemeral exchanges (e.g., ephemeral Diffie-Hellman), such that only the participants of a given session can derive the session key.

   In PMAP, the session key is derived as

   $$SK_{i,s} = H(N_i^{t+1}) \oplus H(N_s^{t+1}),$$

   where N_i^{t+1} and N_s^{t+1} are random values exchanged and protected by long term secrets (CRPs/PUF). If these secrets are compromised, an adversary may reconstruct the random values of past sessions from recorded transcripts, thereby obtaining previous session keys. Hence, PMAP does not guarantee Perfect Forward Secrecy.

4. Privileged Insider Attack: Privileged insiders at the ZSP have access to real identities, CRPs, and pseudonyms, and can therefore break anonymity and link sessions to specific drones. The protocol does not include mechanisms to restrict insider capabilities. Thus, PMAP is vulnerable to privileged insider attacks.

5. Ephemeral Secret Leakage: If temporary values (nonces N_i^t, N_s^t) are exposed during a session, the session key can be immediately reconstructed. However, since these nonces change in each session, such exposure does not affect past or future sessions. Therefore, PMAP provides partial resilience against ephemeral secret leakage, limiting its impact to a single session.

6. Denial of Service (DoS): The protocol requires shuffling operations (Hénon map) and MAC verifications before rejecting invalid messages. An adversary may send multiple forged requests to overload the drone or the ZSP. There is no explicit mitigation mechanism. Thus, PMAP is vulnerable to denial-of-service attacks.

7. Session Key Disclosure: If a session key $SK_{i,s}^{(t)}$ is revealed, the adversary cannot derive session keys from other sessions. This is because each session key is generated from independent nonces (N_i^t, N_s^t) and one way hash functions, ensuring that the disclosure of one key does not compromise past or future sessions. Therefore, PMAP preserves session isolation and is resistant to session key disclosure attacks.

4.7 Simulation and Performance

4.7.1 Communication Cost

Communication cost is one of the primary efficiency metrics in authentication protocols for the IoD, especially due to the bandwidth constraints, mobility, and energy consumption inherent to UAVs. In distributed and dynamic environments, the total volume of transmitted data directly impacts latency, network scalability, and the operational autonomy of drones.

In the context of PMAP, the communication cost is determined by the number of messages exchanged between entities and the total size of the parameters transmitted during the authentication and key establishment phases. Therefore, analyzing this metric allows the assessment of the protocol's impact on network overhead and its practical feasibility in IoD scenarios with multiple nodes and concurrent sessions.

4.7.1.1 Communication Cost (PMAP D2Z)

The communication cost of the $PMAP^{D2Z}$ scenario was estimated based on the total volume of data transmitted during the complete authentication phase between the drone and the ZSP. For this measurement, a fixed prefix of 2 Bytes per field was considered, which is required to ensure the correct decoding of the shuffled structures.

The following sizes were adopted for the cryptographic parameters:

- $|PID| = 32\,\text{Bytes}$
- $|R| = 32\,\text{Bytes}$
- $|MAC| = 32\,\text{Bytes}$

- $|N| = 16\,\text{Bytes}$
- $|Z_s| = 8\,\text{Bytes}$

Table 3 details the individual size of each exchanged message, including their respective authentication codes.

Table 3 – Communication cost of $PMAP^{D2Z}$

Message	**Size (Bytes)**
M_1	62
MAC_1	32
M_2	80
MAC_2	32
M_3	80
M_4	114
MAC_{34}	32
Total D→ZSP	320
Total ZSP→D	112
Total	**432**

The total communication cost of $PMAP^{D2Z}$ is 432 Bytes, with approximately 74% of the traffic originating from the drone. This result is directly related to the redundancy in message exchanges, as discussed previously, which contributes to the increased communication overhead. Consequently, a higher load is imposed on the UAV, potentially impacting its energy consumption, especially in scenarios involving multiple sessions.

4.7.1.2 Communication Cost (PMAP D2D)

In the $PMAP^{D2D}$ scenario, the communication cost was estimated by considering all messages exchanged among Drone A, Drone B, and the ZSP during the complete collaborative authentication process. As in the previous scenario, a fixed prefix of 2 Bytes per field was assumed, along with the same cryptographic parameter sizes.

Table 4 presents the detailed breakdown of the individual messages.

Table 4 – Communication cost of $PMAP^{D2D}$

Message	**Size (Bytes)**
M_1	62
M_2	80
MAC_{12}	32
M_3	96
MAC_3	32
M_4	96
M_5	130
MAC_{45}	32
M_6	62
M_7	80
M_8	114
MAC_{678}	32
M_9	96
M_{10}	130
MAC_{910}	32
M_{11}	96
MAC_{11}	32
Total Drone A	538
Total Drone B	516
Total ZSP	636
Total	**1690**

The total communication cost of $PMAP^{D2D}$ is 1 690 Bytes, which is approximately 3.9 times higher than the D2Z scenario. This increase is not only due to the coordination among three entities, but also to the redundancy in message exchanges across multiple protocol phases, leading to the replication of authenticated structures.

It is also observed that the communication load is relatively balanced between the two drones (538 and 516 Bytes), while the ZSP accounts for 636 Bytes of the total traffic, highlighting its central role in session key establishment and session synchronization. This centralization, combined with the overall communication overhead, may impact the protocol's efficiency in scenarios with a higher density of devices.

4.7.2 Computational Cost

Computational efficiency is a critical factor in authentication protocols for the IoD, given the energy and processing constraints inherent to UAVs. Therefore, in addition to the security analysis, it is also relevant to evaluate the impact of the cryptographic and permutation primitives on the overall execution time of the protocol.

To measure the cost of the elementary operations employed in PMAP, two execution environments were considered. The operations executed by the drones were measured on an ESP32 platform using ESP-IDF, operating at a CPU frequency of 160 MHz. Each operation was executed $n = 100$ times, preceded by 5 warm-up executions, and the results were collected in milliseconds (ms). The benchmark considered messages of 512 bytes, HMAC keys of 32 bytes, and XOR operations over 32 bytes. During the benchmark, the task watchdog was disabled to avoid interference during the sequential execution of long-running operations.

For the ZSP, the execution times were measured in a desktop environment composed of Windows 11, an AMD Ryzen 7 5700G processor, and 16 GB of RAM. This distinction reflects the architectural assumption that drones are resource-constrained devices, whereas the ZSP is executed on a more capable infrastructure.

Table 5 presents the complete descriptive statistics obtained from the ESP32 benchmark, including mean, median, standard deviation, minimum, and maximum values. These measurements are used to characterize the cost of the operations executed by the drones.

Table 5 – Execution time of basic operations on the ESP32

Operation	**Mean (ms)**	**Median (ms)**	**Std. dev. (ms)**	**Min (ms)**	**Max (ms)**
HASH (SHA-256)	0.167080	0.167000	0.001707	0.159000	0.177000
HMAC (SHA-256)	0.312940	0.313000	0.002222	0.305000	0.324000
XOR (32B)	0.008350	0.008000	0.000477	0.008000	0.009000
HENON_MAP (perm)	34.465280	34.462000	0.005783	34.456000	34.474000
HENON_MAP (shuffle)	34.580490	34.578000	0.005740	34.571000	34.590000
HENON_MAP (unshuffle)	34.599910	34.597000	0.005676	34.591000	34.609000

The Hénon permutation is reported separately to show the cost of generating the permutation. However, in the aggregated PMAP estimates, the shuffle and unshuffle operations are used directly, since they already include both the generation and application of the corresponding permutation.

Since the ZSP is assumed to run on a more capable infrastructure, the average execution times measured in the desktop environment are also considered. Table 6 summarizes the average values adopted for each environment in the aggregated cost estimates.

Table 6 – Average execution time adopted for each execution environment

Operation	**Drone / ESP32 (ms)**	**ZSP / Desktop (ms)**
HASH (SHA-256)	0.167080	0.001025
HMAC (SHA-256)	0.312940	0.002672
XOR (32B)	0.008350	0.002111
HENON_MAP (shuffle)	34.580490	0.329701
HENON_MAP (unshuffle)	34.599910	0.333947

Based on these values, the total execution cost can be modeled according to the entity that executes each operation. For the drones, the computational cost is estimated as:

$$\begin{aligned} T_{\text{Drone}} = {} & n_{\text{hash}} \cdot 0.167080 + n_{\text{hmac}} \cdot 0.312940 \\ & + n_{\text{xor}} \cdot 0.008350 + n_{\text{shuf}} \cdot 34.580490 \\ & + n_{\text{unsh}} \cdot 34.599910 \ \ (\text{ms}) \end{aligned}$$

For the ZSP, the computational cost is estimated as:

$$\begin{aligned} T_{\text{ZSP}} = {} & n_{\text{hash}} \cdot 0.001025 + n_{\text{hmac}} \cdot 0.002672 \\ & + n_{\text{xor}} \cdot 0.002111 + n_{\text{shuf}} \cdot 0.329701 \\ & + n_{\text{unsh}} \cdot 0.333947 \ \ (\text{ms}) \end{aligned}$$

where $n_\bullet$ represents the number of executions of each operation throughout the protocol flow. This parametric model enables the estimation of the computational cost for each operating mode of PMAP, considering the different computational capacities of drones and the ZSP.

4.7.2.1 Computational Cost (PMAP D2Z)

In the $PMAP^{D2Z}$ scenario, only two entities participate in the authentication process, namely the Drone and the ZSP. In this evaluation, the Drone is modeled using the ESP32 measurements, while the ZSP is modeled using the desktop measurements.

Table 7 presents the aggregated computational cost per participant.

Table 7 – Aggregated computational cost of $PMAP^{D2Z}$

Operation	**Drone (ms)**	**ZSP (ms)**
HASH	0.501240	0.003075
HMAC	0.938820	0.008016
XOR	0.008350	0.002111
Shuffle	138.321960	0.659402
Unshuffle	34.599910	1.001841
Total	**174.370280**	**1.674445**

The aggregated computational cost for a complete $PMAP^{D2Z}$ session is approximately 176.04 ms. It can be observed that the overall cost is strongly dominated by the operations executed by the drone, especially the shuffle and unshuffle operations based on the Hénon map. In contrast, the cost associated with the ZSP remains comparatively low due to the higher processing capacity of the desktop environment.

This result indicates that the protocol's performance is mainly determined by the cost imposed on the constrained drone platform. Although SHA-256, HMAC-SHA256, and XOR operations have low execution times, the repeated application of Hénon-based transformations becomes the main source of computational latency.

4.7.2.2 Computational Cost (PMAP D2D)

In the $PMAP^{D2D}$ mode, three participants execute the protocol: Drone A, Drone B, and the ZSP. This mode requires additional authentication and coordination steps, resulting in a higher number of cryptographic and permutation operations.

Table 8 presents the aggregated computational cost per participant in the D2D mode.

Table 8 – Aggregated computational cost of $PMAP^{D2D}$

Operation	Drone A (ms)	Drone B (ms)	ZSP (ms)
HASH	0.668320	0.334160	0.001025
HMAC	1.251760	0.625880	0.016032
XOR	0.008350	0.008350	0.000000
Shuffle	172.902450	103.741470	1.648505
Unshuffle	69.199820	103.799730	2.003682
Total	**244.030700**	**208.509590**	**3.669244**

The aggregated computational cost for a complete $PMAP^{D2D}$ session is approximately 456.21 ms, which is about 2.6 times higher than in the D2Z scenario. This increase is mainly due to the higher number of shuffle and unshuffle operations executed by the drones during the coordination of the three participating entities.

Similarly to the D2Z mode, the computational cost is largely dominated by operations based on the Hénon map. However, when the ZSP is evaluated in a desktop environment, its contribution to the total execution time becomes comparatively small. Therefore, the main computational bottleneck of PMAP is concentrated on the drone side, especially in the Hénon-based shuffle and unshuffle procedures.

When compared with the values identified in the systematic literature review, the computational cost obtained for PMAP is considerably higher than the average execution time reported by the analyzed studies, which was approximately 37 ms. However, this comparison should be interpreted with caution, since there is no standardized methodology for reporting computational cost in IoD authentication protocols. As a result, the hardware platforms, implementation strategies, measured primitives, and aggregation criteria may vary significantly among the studies.

In this work, the operations executed by the drones were measured on an ESP32, which is a constrained embedded platform with considerably lower processing power than desktop or server-class environments commonly used in experimental evaluations. Therefore, the higher execution time observed for PMAP may be partly explained by the use of a

low-power hardware platform, as well as by the explicit measurement of the Hénon-based shuffle and unshuffle operations. Since these operations dominate the computational cost of the protocol, their execution on the ESP32 has a significant impact on the final latency.

Thus, although PMAP presents a computational cost above the average identified in the literature, this result should not be interpreted only as a direct disadvantage of the protocol. Rather, it also reflects the lack of standardization in how computational costs are measured and reported, especially regarding the hardware used in the experiments and the set of operations included in the final cost estimation. Overall, the results indicate that PMAP is executable on a constrained embedded platform, but its latency is strongly influenced by the cost of the chaotic permutation mechanism, particularly in the D2D mode.

4.8 Synthesis of Identified Limitations

Based on the reassessment conducted throughout this chapter, the main limitations of the PMAP protocol can be organized according to the analytical dimension from which they were identified. This synthesis is important because not all limitations arise from the same type of analysis. Some aspects are related to the formal security model, while others are associated with the protocol architecture, the adopted threat model, privacy assumptions, or performance evaluation.

The main limitations identified in PMAP are summarized as follows:

1. **Absence of Perfect Forward Secrecy.** PMAP does not employ an ephemeral key exchange mechanism, such as ECDH. As a result, if long term secrets or PUF-related values are compromised, previously recorded sessions may become vulnerable, since the session key is derived from nonces protected by these long term elements.

2. **Dependence on the ZSP as a central entity.** PMAP relies on the ZSP to validate identities, manage pseudonyms and CRP-related information, and coordinate authentication. This simplifies credential management, but also means that the unavailability of the ZSP may prevent the establishment of new authenticated sessions, characterizing it as a potential single point of failure.

3. **Exposure to privileged insider attacks.** In PMAP, the ZSP stores sensitive information required for protocol execution, including real identities, pseudonyms, and CRP-related values. Therefore, a privileged insider with access to the ZSP database may be able to associate pseudonyms with real drone identities and correlate different sessions. This limitation affects the privacy guarantees of the protocol,

since anonymity depends not only on the use of dynamic pseudonyms, but also on the trustworthiness and access control mechanisms of the ZSP.

4. **High communication cost due to message redundancy.** PMAP presents redundancy in the exchanged messages in both D2Z and D2D modes, since several structures are transmitted with repeated authentication and validation information. This redundancy increases the overall communication cost of the protocol, especially in the D2D scenario, where the number of participants and exchanged messages is higher.

5. **Limitations under physical capture scenarios.** Although PMAP assumes the use of tamper-resistant PUFs, more realistic physical compromise scenarios may involve the extraction of values stored in non-volatile memory, intermediate states, or auxiliary parameters. These cases are not fully captured by classical network-oriented adversary models.

These limitations do not necessarily invalidate the formal results obtained for PMAP under the adopted models. Instead, they delimit the scope of the security guarantees provided by the protocol and indicate aspects that require additional attention when considering practical IoD deployments.

4.8.1 Origin of the Identified Limitations

To clarify the scope of each limitation, Table 9 distinguishes whether the identified aspect originates from informal analysis, architectural analysis, threat model discussion, or performance evaluation. This distinction is relevant because not all limitations arise from the same type of assessment, and some of them do not follow directly from the automated formal verification.

Table 9 – Origin of the main limitations identified in the PMAP reassessment

Identified limitation	**Origin of analysis**	**Observation**
Absence of Perfect Forward Secrecy	Informal analysis / threat model	Associated with the absence of an independent ephemeral key exchange mechanism.
Dependence on the ZSP	Architectural analysis	Related to the centralized role of the ZSP in identity validation, pseudonym management, and session coordination.
High communication cost due to message redundancy	Performance analysis	Associated with redundant message structures in both D2Z and D2D modes, with greater impact in the D2D scenario.
Partial physical compromise of drones	Threat model extension / informal analysis	Classical network-oriented models do not fully cover practical physical extraction scenarios.
Potential privileged insider exposure	Informal analysis	Related to the ZSP's access to real identities, pseudonyms, and CRP-related information.
Potential Denial-of-Service exposure	Informal and performance analysis	Associated with the cost of processing forged or invalid requests before rejection.

Therefore, the reassessment indicates that PMAP presents limitations when broader architectural, physical, privacy, and performance aspects are considered. Based on this synthesis, the next section discusses mitigation strategies for the weaknesses that can be directly addressed at the protocol level.

4.9 Possible mitigation strategies

The analysis conducted throughout this chapter has highlighted three relevant limitations of the PMAP protocol in its original form: the lack of PFS, exposure to privileged insider attacks at the ZSP, and susceptibility to DoS attacks.

This section discusses possible mitigation directions based on the limitations identified during the reassessment.

4.9.1 Ensuring Perfect Forward Secrecy

In PMAP, the session key is derived from *nonces* protected by hash functions, according to the expression:

$$SK = H(N_i^{t+1}) \oplus H(N_s^{t+1}).$$

However, the potential exposure of long term secrets (such as PUF-related material or stored CRPs) may compromise the confidentiality of previously established session keys, since the protocol does not incorporate independent ephemeral components into the key derivation process, thus violating the PFS property.

A classical approach to achieving PFS consists of incorporating an ephemeral key exchange mechanism based on Diffie-Hellman. In this model, each session establishes a shared secret derived from ephemeral values independently generated by the parties, such that the session key does not directly depend on long term secrets. [84]

As a result, even if an adversary later obtains the long term keys of the entities, it will not be able to reconstruct previously established session keys, since the ephemeral parameters used in their derivation are not reused. For resource constrained scenarios, such as in the IoD context, elliptic curve based variants are recommended due to their lower computational cost and higher efficiency in shared secret generation [19].

A simple and low cost modification to PMAP consists of including ephemeral public keys X and Y (derived from ephemeral secrets x and y) within already protected protocol messages (e.g., in M_2 and M_3/M_4 in the D2Z mode, and in M_3 and M_{11} in the D2D mode). The session key can then be derived as:

$$SK \;=\; H\big(\mathrm{ECDH}(x, Y) \parallel N_i^{t+1} \parallel N_s^{t+1}\big)\,. \tag{4.67}$$

In this way, even if long term secrets are revealed after protocol execution, the ephemeral value $\mathrm{ECDH}(x, Y)$ remains inaccessible to the adversary, preserving the confidentiality of past session keys. This modification maintains the freshness mechanism based on nonces while adding the PFS property.

4.9.2 Reducing the Impact of Privileged Insiders

In the original PMAP design, the ZSP stores both the real identities of drones (ID_i) and the complete database of CRPs. As a result, an internal operator with access to this database could associate pseudonyms with real identities and correlate different communication sessions of the same drone.

One way to mitigate this risk is to prevent the ZSP from simultaneously holding both authentication data and the real identities of drones. Instead, the server should retain only the information strictly necessary for protocol execution, while the mapping between pseudonyms and real identities is stored separately and securely, accessible only under authorized conditions. This approach aligns with the principles of privacy preservation and identity exposure minimization discussed in the literature on anonymous and privacy-preserving authentication in IoD and UAV environments [56, 59, 70, 71, 44]. Although this strategy does not fully eliminate the need for institutional trust in the ZSP, it reduces the potential impact of malicious insiders and limits the ability to directly link sessions to real identities.

Additionally, the adoption of the principle of least privilege is recommended for access to system databases. This implies that ZSP operators should have only the permissions strictly necessary for their roles, while sensitive operations such as resolving the real identity of a drone should be restricted and logged through auditing mechanisms.

Although these measures do not completely remove the need for institutional trust in the ZSP, they significantly reduce the risks associated with malicious insiders and limit the impact of potential privilege abuse.

4.9.3 Possible Mitigation of Denial-of-Service Attacks

The vulnerability to DoS attacks arises from the fact that PMAP may perform computationally expensive operations (such as unshuffle and MAC verification) before rejecting invalid messages. An adversary can exploit this behavior by sending multiple forged requests, increasing the CPU and energy consumption of the involved devices.

A simple countermeasure consists of applying early lightweight filtering [84]. Upon receiving a request, the ZSP should initially perform only low cost checks:

- validation of message format and size;
- freshness verification based on a temporal window T;
- replay detection for recent pairs (PID_i^t, N_i^t) within the same window.

Messages that fail any of these checks should be discarded before executing more expensive operations, such as unshuffle or MAC computation. This strategy may reduce the impact of denial-of-service attacks without modifying the fundamental structure of the protocol.

5 Conclusion

This work presented a comprehensive analysis of authentication protocols applied to the IoD, combining two complementary perspectives: a systematic literature review and an in-depth case study of the PMAP protocol.

The literature review, based on 61 articles selected according to defined methodological criteria, revealed that despite the diversity of existing proposals, research in this area tends to follow a recurring structure. In general, studies describe the system topology, the employed cryptographic primitives, the adopted threat model, the application of formal or informal verification methods, and a performance analysis. However, a significant methodological heterogeneity was identified, particularly in the evaluation of security and performance. In many cases, different threat models, metrics, hardware platforms, and experimental parameters are used without standardization, which hinders direct comparison between protocols and compromises the reproducibility of results.

As a complement to the literature review, the PMAP protocol was analyzed from multiple perspectives. From a formal security standpoint, an upper bound on the adversary's advantage in the RoR model was established, relating it to the possibility of MAC forgery, the distinguishability between the Hénon map based shuffling mechanism and a truly random permutation, and the birthday bound applied to the CRP space of the PUF. Furthermore, formal verification using AVISPA, Scyther, and ProVerif did not identify violations of the specified confidentiality and authentication properties in both PMAP D2Z and PMAP D2D scenarios, under the assumptions of the respective models. The informal analysis revealed that PMAP does not guarantee PFS and remains vulnerable to privileged insider attacks at the ZSP. Additionally, the protocol does not include specific mechanisms to mitigate DoS attacks, leaving it potentially exposed to such threats.

Regarding performance, the conducted experiments enabled the development of a parametric model for computational cost and the estimation of the aggregated execution time per participating entity. In this evaluation, the operations executed by the drones were measured on an ESP32 platform, while the operations executed by the ZSP were measured in a desktop environment. The results indicated an aggregated computational cost of approximately 176.04 ms for a complete $PMAP^{D2Z}$ session and 456.21 ms for a complete $PMAP^{D2D}$ session. These values are mainly influenced by the Hénon map based shuffle and unshuffle operations executed by the drones, indicating that the chaotic permutation mechanism represents the main computational bottleneck of the protocol in constrained embedded platforms.

When compared with the average execution time identified in the systematic literature review, approximately 37 ms, the computational cost obtained for PMAP is considerably higher. However, this comparison should be interpreted with caution, since the analyzed studies do not follow a standardized methodology for reporting computational cost. The hardware platforms, implementation strategies, measured primitives, and aggregation criteria vary significantly among the studies. Thus, the higher latency observed in this work may be partly explained by the use of the ESP32 as a low-power embedded platform and by the explicit measurement of the Hénon-based operations in the protocol flow.

From a communication perspective, the estimated cost was 432 Bytes per session in the D2Z mode and 1690 Bytes in the D2D mode. These values remain compatible with low-rate links typical of UAV environments, although message redundancy and aggregated MACs represent a significant portion of the communication overhead.

In summary, PMAP presents relevant mechanisms for authentication and privacy in IoD environments, but the results of this study indicate that the protocol can benefit from design improvements. In terms of security, the main limitations are related to the absence of Perfect Forward Secrecy, the possibility of privileged insider attacks at the ZSP, and the lack of specific mechanisms for denial-of-service mitigation. In terms of performance, the main limitation is associated with the computational cost of the Hénon-based shuffle and unshuffle operations when executed on constrained drone hardware.

Finally, the results obtained in this research reinforce the importance of systematic analyses and more detailed evaluations in the study of authentication protocols for the Internet of Drones. The combination of literature review, formal analysis, automated verification, and performance evaluation provided a deeper understanding of the properties and limitations of the PMAP protocol. It is expected that the findings presented here will contribute to the development of more robust and efficient authentication protocols for IoD environments, as well as support future research aimed at improving security mechanisms and evaluation methodologies in this context.

5.1 Future Work

The results obtained in this dissertation indicate several directions for future work. First, the proposed mitigation mechanisms should be implemented and experimentally evaluated in order to assess their impact on the security and performance of the PMAP protocol. In particular, the incorporation of ephemeral key exchange mechanisms for providing PFS should be analyzed in terms of computational cost, communication overhead, and feasibility in resource-constrained IoD environments.

Another relevant direction is the practical validation of PMAP on additional embedded

platforms, drone simulation frameworks, and real UAV testbeds. Although the performance analysis conducted in this work provides an initial estimation of the protocol's cost, future experiments should consider energy consumption, computational cost and communication cost. In addition, adversarial experiments could be conducted in controlled and authorized test environments. Initially, protocol messages collected from real executions could be analyzed to evaluate the behavior of PMAP against an adversary with full control of the communication channel, considering actions such as replay, modification, injection, and correlation of messages. Subsequently, a stronger scenario could be considered by assuming partial physical access to a drone, in which locally stored information or auxiliary parameters are obtained and then used to reassess the protocol under network-level adversarial conditions. This two-stage evaluation would provide a broader understanding of the protocol's behavior under practical attack scenarios.

Future studies may also focus on optimizing the PMAP message structure and computational distribution. As identified in the reassessment, both D2Z and D2D modes present redundant message exchanges, and the D2D scenario introduces considerably higher communication overhead. Therefore, future work should investigate whether the protocol flow can be simplified without compromising authentication, integrity, freshness, and session key establishment properties. In parallel, the computational cost associated with Hénon map operations should be further analyzed, considering possible optimizations or partial offloading strategies to the ZSP, in order to reduce processing time and energy consumption on the drone side.

Another possible direction is the investigation of dynamic service discovery mechanisms for D2D communication. The original protocol assumes that a drone already knows the pseudonym of the target drone before initiating the authentication process. However, in highly mobile and ad-hoc IoD scenarios, this assumption may not always hold. Future work could investigate mechanisms that allow drones to identify nearby devices and obtain the necessary information to initiate D2D authentication, while evaluating the impact of this process on latency, scalability, and communication overhead.

Future research should also investigate alternative architectural models to reduce the dependence on the ZSP as a central entity. This analysis should consider redundant edge nodes, distributed coordination, cached credentials, or limited fallback mechanisms that allow drones to maintain secure communication for restricted periods when the ZSP is unavailable. Such approaches could improve resilience against failures or denial-of-service conditions, especially in larger drone swarms.

Finally, future work may explore decentralized mechanisms for storing and validating drone-related information. Blockchain or distributed ledger technologies could be considered for integrity verification, auditability, access control, or revocation records. However,

due to the latency, storage, and energy constraints of IoD environments, future studies should evaluate lightweight and permissioned approaches, avoiding excessive overhead on UAV devices.

Bibliography

[1] BOCCADORO, P.; STRICCOLI, D.; GRIECO, L. A. An extensive survey on the internet of drones. *Ad Hoc Networks*, v. 122, p. 102600, nov. 2021.

[2] MEKDAD, Y. et al. A comprehensive security and performance assessment of UAV authentication schemes. *Security and Privacy*, v. 7, n. 4, p. e338, 2023.

[3] CHOO, K.-K. R.; VINEL, A.; HUANG, X. Security and privacy for the internet of drones: Challenges and solutions. *IEEE Communications Magazine*, v. 56, n. 1, p. 64–69, 2018.

[4] PU, C. et al. A lightweight and privacy-preserving mutual authentication and key agreement protocol for internet of drones environment. *IEEE Internet of Things Journal*, v. 9, n. 12, p. 9918–9933, 2022.

[5] DOLEV, D.; YAO, A. C. On the security of public key protocols. *IEEE Transactions on Information Theory*, v. 29, n. 2, p. 198–208, Mar. 1983.

[6] CANETTI, R.; KRAWCZYK, H. *Universally Composable Notions of Key Exchange and Secure Channels*. [S.l.], 2002. Disponível em: <https://eprint.iacr.org/2002/059>.

[7] STALLINGS, W. *Cryptography and Network Security: Principles and Practice*. 5. ed. Upper Saddle River, NJ: Prentice Hall, 2011. ISBN 9780136097043.

[8] TANVEER, M. et al. Seaf-iod: Secure and efficient user authentication framework for the internet of drones. *Computer Networks*, v. 247, p. 110449, 2024.

[9] HOLZMANN, G. J. *The SPIN Model Checker: Primer and Reference Manual*. 1. ed. Boston, MA: Addison-Wesley, 2004.

[10] ABDALLA, M.; FOUQUE, P.-A.; POINTCHEVAL, D. Password-based authenticated key exchange in the three-party setting. In: VAUDENAY, S. (Ed.). *Public Key Cryptography – PKC 2005*. [S.l.]: Springer, 2005. (Lecture Notes in Computer Science, v. 3386), p. 65–84.

[11] BURROWS, M.; ABADI, M.; NEEDHAM, R. A logic of authentication. *ACM Transactions on Computer Systems*, v. 8, n. 1, p. 18–36, Feb. 1990.

[12] BELLARE, M.; BOLDYREVA, A.; PALACIO, A. *An Uninstantiable Random-Oracle-Model Scheme for a Hybrid Encryption Problem*. [S.l.], 2003. Disponível em: <https://eprint.iacr.org/2003/077>.

[13] ARMANDO, A. et al. The avispa tool for the automated validation of internet security protocols and applications. In: *Computer Aided Verification (CAV 2005)*. [S.l.]: Springer, 2005. (Lecture Notes in Computer Science, v. 3576), p. 281–285.

[14] BLANCHET, B.; SMYTH, B. Automated reasoning for equivalences in the applied pi calculus with barriers. In: *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*. [S.l.: s.n.], 2016. p. 310–324.

[15] CREMERS, C. J. F. The scyther tool: Verification, falsification, and analysis of security protocols. In: *Computer Aided Verification (CAV 2008)*. [S.l.]: Springer, 2008. (Lecture Notes in Computer Science, v. 5123), p. 414–418.

[16] MAO, W.; BOYD, C. Towards formal analysis of security protocols. In: *Proceedings of the Computer Security Foundations Workshop VI*. [S.l.: s.n.], 1993. p. 147–158.

[17] GONG, L.; NEEDHAM, R.; YAHALOM, R. Reasoning about belief in cryptographic protocols. In: *Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy*. [S.l.: s.n.], 1990. p. 234–248.

[18] BASIN, D. et al. Tamarin: Verification of large-scale, real-world, cryptographic protocols. *IEEE Security & Privacy*, v. 20, n. 3, p. 24–32, maio 2022.

[19] DAS, A. K. et al. igcacs-iod: An improved certificate-enabled generic access control scheme for internet of drones deployment. *IEEE Access*, v. 9, p. 87024–87048, 2021.

[20] YU, S. et al. Slap-iod: Secure and lightweight authentication protocol using physical unclonable functions for internet of drones in smart city environments. *IEEE Transactions on Vehicular Technology*, v. 71, n. 10, p. 10374–10388, 2022.

[21] Parsifal Ltd. *Parsifal: A web-based tool to support systematic literature reviews*. `https://parsif.al/`. Accessed: 2025-05-07.

[22] PAGE, M. J. et al. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, v. 372, p. n71, 2021.

[23] BERINI, A. D. E. et al. Hcala: Hyperelliptic curve-based anonymous lightweight authentication scheme for internet of drones. *Pervasive and Mobile Computing*, v. 92, p. 101798, 2023.

[24] TANVEER, M. et al. Paf-iod: Puf-enabled authentication framework for the internet of drones. *IEEE Transactions on Vehicular Technology*, v. 73, n. 7, p. 9560–9574, 2024.

[25] LEI, Y. et al. A lightweight authentication protocol for uav networks based on security and computational resource optimization. *IEEE Access*, v. 9, p. 53769–53785, 2021.

[26] ZHANG, Z. et al. Prlap-iod: A puf-based robust and lightweight authentication protocol for internet of drones. *Computer Networks*, v. 238, p. 110118, 2024.

[27] SON, S. et al. A zero-trust authentication scheme with access control for 6g-enabled iot environments. *IEEE Access*, v. 12, p. 154066–154079, 2024.

[28] LOUNIS, K.; DING, S. H. H.; ZULKERNINE, M. D2d-map: A drone to drone authentication protocol using physical unclonable functions. *IEEE Transactions on Vehicular Technology*, v. 72, n. 4, p. 5079–5093, 2023.

[29] ZHANG, Y. et al. A novel and efficient authentication scheme based on uav-uav environment. *Wireless Communications and Mobile Computing*, v. 2023, 2023.

[30] SHARIQ, M. et al. Design of provably secure and lightweight authentication protocol for unmanned aerial vehicle systems. *Computer Communications*, v. 228, p. 107971, 2024.

[31] XIA, T. et al. A quantum-resistant identity authentication and key agreement scheme for uav networks based on kyber algorithm. *Drones*, v. 8, n. 8, p. 359, 2024.

[32] ZHANG, L. et al. A puf-based lightweight authentication and key agreement protocol for smart uav networks. *IET Communications*, v. 16, n. 10, p. 1142–1159, 2022.

[33] ALGARNI, A. D.; INNAB, N.; ALGARNI, F. A verifiably secure and robust authentication protocol for synergistically-assisted iod deployment drones. *PLOS ONE*, v. 20, n. 3, p. e0314475, 2025.

[34] RAHMAN, K. et al. An efficient authentication and access control protocol for securing uav networks against anomaly-based intrusion. *IEEE Access*, v. 12, p. 62750–62764, 2024.

[35] KARMAKAR, R.; KADDOUM, G.; AKHRIF, O. A puf and fuzzy extractor-based uav-ground station and uav-uav authentication mechanism with intelligent adaptation of secure sessions. *IEEE Transactions on Mobile Computing*, v. 23, n. 5, p. 3858–3875, 2024.

[36] ZHANG, Z. et al. Tagka: Threshold authenticated group key agreement protocol against member disconnect for uanet. *IEEE Transactions on Vehicular Technology*, v. 72, n. 11, p. 14987–15001, 2023.

[37] TAN, Y. et al. Blockchain-assisted distributed and lightweight authentication service for industrial unmanned aerial vehicles. *IEEE Internet of Things Journal*, v. 9, n. 18, p. 16928–16940, 2022.

[38] AKRAM, M. A. et al. Blockchain-based privacy-preserving authentication protocol for uav networks. *Computer Networks*, v. 224, p. 109638, 2023.

[39] ZHANG, J. et al. An enhanced certificateless blockchain-assisted authentication and key agreement protocol for internet of drones. *IEEE Transactions on Network Science and Engineering*, 2025. Early Access.

[40] SHAMSHAD, S.; BELGUITH, S.; ORACEVIC, A. Securing the skies: A cutting-edge authenticated key establishment protocol for the internet of drones. *IEEE Internet of Things Journal*, 2025. Early Access.

[41] BERA, B.; CHATTARAJ, D.; DAS, A. K. Designing secure blockchain-based access control scheme in iot-enabled internet of drones deployment. *Computer Communications*, v. 153, p. 229–249, 2020.

[42] LIU, Y. et al. An authentication and signature scheme for uav-assisted vehicular ad hoc network providing anonymity. *Journal of Systems Architecture*, v. 142, p. 102935, 2023.

[43] SON, S. et al. Design of secure and lightweight authentication scheme for uav-enabled intelligent transportation systems using blockchain and puf. *IEEE Access*, v. 11, p. 60240–60253, 2023.

[44] KHAN, M. A.; et al. A provable and privacy-preserving authentication scheme for uav-enabled intelligent transportation systems. *IEEE Transactions on Industrial Informatics*, v. 18, n. 5, p. 3416–3425, 2022.

[45] EL-ZAWAWY, M. A.; BRIGHENTE, A.; CONTI, M. Authenticating drone-assisted internet of vehicles using elliptic curve cryptography and blockchain. *IEEE Transactions on Network and Service Management*, v. 20, n. 2, p. 1775–1789, 2023.

[46] KUMAR, V.; ALI, R.; SHARMA, P. K. Ioepm+: A secured and lightweight 6g-enabled pollution monitoring authentication framework using iot and blockchain technology. *Computer Networks*, v. 250, p. 110554, 2024.

[47] WEN, K. et al. A secure authentication protocol supporting efficient handover for uav. *Mathematics*, v. 12, n. 5, p. 716, 2024.

[48] LI, L. et al. Csecmas: An efficient and secure certificate signing based elliptic curve multiple authentication scheme for drone communication networks. *Applied Sciences*, v. 12, n. 18, p. 9203, 2022.

[49] JAN, S. U.; QAYUM, F.; KHAN, H. U. Design and analysis of lightweight authentication protocol for securing iod. *IEEE Access*, v. 9, p. 69287–69306, 2021.

[50] KHALID, H. et al. Hoopoe: High performance and efficient anonymous handover authentication protocol for flying out of zone uavs. *IEEE Transactions on Vehicular Technology*, v. 72, n. 8, p. 10906–10920, 2023.

[51] CHEN, L. et al. Puf-based dynamic secret-key strategy with hierarchical blockchain for uav swarm authentication. *Computer Communications*, v. 218, p. 31–43, 2024.

[52] ALGARNI, F.; JAN, S. U. Pslaps-iod: A provable secure and lightweight authentication protocol for securing internet-of-drones (iod) environment. *IEEE Access*, v. 12, p. 45948–45960, 2024.

[53] TANVEER, M. et al. Ruam-iod: A robust user authentication mechanism for the internet of drones. *IEEE Access*, v. 10, p. 19836–19851, 2022.

[54] YU, S. et al. Laka-uav: Lightweight authentication and key agreement scheme for cloud-assisted unmanned aerial vehicle using blockchain in flying ad-hoc networks. *Computer Networks*, v. 224, p. 109612, 2023.

[55] JU, S. et al. Blockchain-assisted secure and lightweight authentication scheme for multi-server internet of drones environments. *Mathematics*, v. 12, n. 24, p. 3965, 2024.

[56] TANVEER, M. et al. Saaf-iod: Secure and anonymous authentication framework for the internet of drones. *IEEE Transactions on Vehicular Technology*, v. 73, n. 1, p. 232–244, 2024.

[57] CHOI, J. et al. A puf-based secure authentication and key agreement scheme for the internet of drones. *Sensors*, v. 25, n. 3, p. 982, 2025.

[58] SRINIVAS, J. et al. Tcalas: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment. *IEEE Transactions on Vehicular Technology*, v. 68, n. 7, p. 6903–6916, 2019.

[59] LI, C.-T. et al. An efficient authenticated key agreement scheme supporting privacy-preservation for internet of drones communications. *Sensors*, v. 22, n. 23, p. 9534, 2022.

[60] SHARMA, M. et al. Psecas: A physical unclonable function based secure authentication scheme for internet of drones. *Computers and Electrical Engineering*, v. 108, p. 108662, 2023.

[61] PARK, Y. et al. Provably secure mutual authentication and key agreement scheme using puf in internet of drones deployments. *Sensors*, v. 23, n. 4, p. 2034, 2023.

[62] WU, T. et al. Amassing the security: An enhanced authentication protocol for drone communications over 5g networks. *Drones*, v. 6, n. 1, p. 10, 2022.

[63] EL-ZAWAWY, M. A.; BRIGHENTE, A.; CONTI, M. Setcap: Service-based energy-efficient temporal credential authentication protocol for internet of drones. *Computer Networks*, v. 206, p. 108804, 2022.

[64] SHARMA, J.; MEHRA, P. S. Hcfaiun: A novel hyperelliptic curve and fuzzy extractor-based authentication for secure data transmission in iot-based uav networks. *Vehicular Communications*, v. 49, p. 100834, 2024.

[65] ALZAHRANI, A. A. Vskap-iod: A verifiably secure key agreement protocol for securing iod environment. *IEEE Access*, v. 12, p. 58039–58056, 2024.

[66] JAN, S. U.; ABBASI, I. A.; ALGARNI, F. A mutual authentication and cross verification protocol for securing internet-of-drones (iod). *Computers, Materials & Continua*, v. 72, n. 3, p. 5845–5869, 2022.

[67] CHANDRAN, I.; VIPIN, K. A puf secured lightweight mutual authentication protocol for multi-uav networks. *Computer Networks*, v. 253, p. 110717, 2024.

[68] SHARMA, J.; MEHRA, P. S. G2caiun: A novel genus-2 curve-based authentication for secure data transmission in iot-based uav networks. *Physical Communication*, v. 71, p. 102647, 2025.

[69] CHOE, H.; KANG, D. Ecc-based authentication protocol for military internet of drone (iod): A holistic security framework. *IEEE Access*, v. 13, p. 21503–21519, 2025.

[70] LEE, T.-F.; YE, X.; HUANG, W.-J. Lightweight privacy-preserving authenticated key agreements using physically unclonable functions for internet of drones. *Journal of Information Security and Applications*, v. 87, p. 103915, 2024.

[71] IRSHAD, A. et al. Dac-md: A privacy-preserving drone-access control scheme for last-mile delivery. *Transactions on Emerging Telecommunications Technologies*, v. 35, n. 3, p. e4958, 2024.

[72] WAZID, M. et al. Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment. *IEEE Internet of Things Journal*, v. 6, n. 2, p. 3572–3584, 2019.

[73] TANVEER, M. et al. Ramp-iod: A robust authenticated key management protocol for the internet of drones. *IEEE Internet of Things Journal*, v. 9, n. 2, p. 1339–1353, 2022.

[74] HUSSAIN, S. et al. An efficient and reliable user access protocol for internet of drones. *IEEE Access*, v. 11, p. 59688–59700, 2023.

[75] MALL, P. et al. Comsec++: Puf-based secured light-weight mutual authentication protocol for drone-enabled wsn. *Computer Networks*, v. 199, p. 108476, 2021.

[76] WANG, D. et al. Authentication and key agreement based on three factors and puf for uav-assisted post-disaster emergency communication. *IEEE Internet of Things Journal*, v. 11, n. 11, p. 20457–20472, 2024.

[77] PU, C. et al. litegap: Lightweight group authentication protocol for internet of drones systems. *IEEE Transactions on Vehicular Technology*, v. 73, n. 4, p. 5849–5860, 2024.

[78] KWON, D. et al. Design of secure handover authentication scheme for urban air mobility environments. *IEEE Access*, v. 10, p. 42529–42541, 2022.

[79] YAHUZA, M. et al. An edge assisted secure lightweight authentication technique for safe communication on the internet of drones network. *IEEE Access*, v. 9, p. 31420–31440, 2021.

[80] LAMACCHIA, B.; LAUTER, K.; MITYAGIN, A. Stronger security of authenticated key exchange. In: SUSILO, W.; LIU, J. K.; MU, Y. (Ed.). *Provable Security (ProvSec 2007)*. [S.l.]: Springer, 2007. (Lecture Notes in Computer Science, v. 4784), p. 1–16.

[81] OUAFI, K.; PHAN, R. C.-W. Privacy of recent rfid authentication protocols. In: CHEN, L.; MU, Y.; SUSILO, W. (Ed.). *Information Security Practice and Experience (ISPEC 2008)*. [S.l.]: Springer, 2008. (Lecture Notes in Computer Science, v. 4991), p. 263–277.

[82] CHUANG, K.-H. et al. A physically unclonable function using soft oxide breakdown featuring 0 native ber and 51.8 fj/bit in 40-nm cmos. *IEEE Journal of Solid-State Circuits*, v. 54, n. 10, p. 2765–2776, 2019.

[83] HÉNON, M. A two-dimensional mapping with a strange attractor. *Communications in Mathematical Physics*, v. 50, p. 69–77, 1976.

[84] MENEZES, A. J.; OORSCHOT, P. C. van; VANSTONE, S. A. *Handbook of Applied Cryptography*. [S.l.]: CRC Press, 1996.

A Comparison Among Protocols

This section presents all the data extracted during the data extraction phase of the methodological process. Tables 10 and 11 summarize the topology, threat model, formal tools, formal models, and performance metrics for each article. Tables 12 and 13 present the cryptographic primitives for each study. Finally, Tables 14 and 15 present the informal security analyses conducted across the selected articles.

Table 10 – Topology, threat models, formal verification tools, models, and performance metrics used in the analyzed studies (Part 1)

Article	Topology											Threat Model			Formal Tools					Formal Models						Performance	
	Drone	GSS	User	TA	Blockchain	CR	Vehicle	RSU	ZSP	AP	Sensor	DY	CK	eCK	AVISPA	ProVerif	Scyther	SPIN	Tamarin	ROR	BAN	ROM	Mao-Boyd	Ouafi-Phan	GNY	Comp	Comm
[19]	✔	✔	×	×	×	✔	×	×	×	×	×	✔	✔	×	✔	×	×	×	×	✔	×	×	×	×	×	81.896	2272
[41]	✔	✔	×	×	✔	✔	×	×	×	×	×	✔	✔	×	✔	×	×	×	×	✔	×	×	×	×	×	63.589	1888
[29]	✔	✔	×	×	×	×	×	×	×	×	×	✔	×	×	✔	×	×	×	×	✔	×	×	×	×	×	42.22	1760
[37]	✔	✔	×	×	✔	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	✔	×	×	×	×	23	1280
[56]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	✔	×	×	×	✔	×	×	✔	×	×	×	×	×	9.54	1664
[8]	✔	✔	✔	×	×	✔	×	×	×	×	×	✔	✔	×	×	×	✔	×	×	×	✔	×	×	×	×	21.63	2272
[42]	✔	×	×	✔	×	×	✔	✔	×	×	×	✔	×	×	×	✔	×	×	×	×	×	✔	×	×	×	2.10671	6176
[4]	✔	✔	×	×	×	×	×	×	×	×	×	✔	×	×	✔	×	×	×	×	×	×	×	✔	×	×	3576	-
[78]	✔	✔	×	×	×	✔	×	×	✔	×	×	✔	✔	×	✔	×	×	×	×	✔	✔	×	×	×	×	11.0001	2560
[57]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	✔	×	✔	×	×	×	×	✔	✔	×	×	×	×	0.2046	2112
[28]	✔	×	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	×	✔	×	×	×	×	×	×	340.0	1744
[25]	✔	✔	×	×	×	×	×	×	×	✔	✔	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	2.0622	1568
[77]	✔	×	×	×	×	×	×	×	✔	×	×	✔	×	×	✔	×	×	×	×	×	×	×	✔	×	×	-	-
[58]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	✔	×	✔	×	×	×	×	✔	×	×	×	×	×	26.7	1536
[38]	✔	✔	×	×	✔	×	×	×	×	×	×	✔	×	×	✔	×	×	×	×	×	×	✔	×	×	×	0.44819	1152
[30]	✔	✔	×	×	×	×	×	×	×	×	×	✔	×	×	✔	×	✔	×	×	×	×	×	×	✔	×	1600	-
[75]	✔	✔	✔	×	×	×	×	×	×	×	✔	✔	×	×	×	×	✔	×	×	×	×	×	×	×	×	25.47	4000
[59]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	✔	×	×	✔	×	×	×	×	✔	×	×	×	×	1.022	2176
[47]	✔	✔	×	✔	×	×	×	×	×	×	×	✔	×	×	×	✔	×	×	×	✔	×	×	×	×	×	0.784	1856
[43]	✔	×	✔	×	×	×	✔	✔	×	×	×	✔	✔	×	✔	×	×	×	×	✔	✔	×	×	×	×	6.861	2656
[31]	✔	✔	×	×	×	×	×	×	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	54.379	1836
[60]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	✔	×	✔	×	×	×	×	✔	×	×	×	×	×	0.154634	2688
[46]	✔	×	×	✔	✔	×	×	×	×	×	✔	✔	✔	×	×	×	✔	×	×	✔	×	×	×	×	×	48.544	3712
[48]	✔	✔	×	✔	×	×	×	×	×	×	×	✔	×	×	×	×	×	✔	×	×	×	×	×	×	×	18.298	2944
[61]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	×	×	✔	×	×	×	×	✔	✔	×	×	×	×	2.138	2560
[62]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	×	×	×	✔	×	×	×	✔	×	×	×	×	×	0.135	2176
[32]	✔	✔	×	×	×	×	×	×	×	×	×	✔	×	×	✔	×	×	×	×	✔	×	×	×	×	×	0.0676	2688
[54]	✔	✔	✔	✔	✔	×	×	×	×	×	×	✔	✔	×	✔	×	×	×	×	✔	×	×	×	×	×	4.377	4160
[63]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	✔	×	✔	×	×	×	×	✔	✔	×	×	×	×	25.42	2432
[26]	✔	×	✔	✔	×	×	×	×	×	✔	×	✔	✔	×	✔	×	×	×	×	✔	×	×	×	×	×	0.67	1760

Table 11 – Topology, threat models, formal verification tools, models, and performance metrics used in the analyzed studies (Part 2)

Article	Topology											Threat Model			Formal Tools					Formal Models						Performance	
	Drone	GSS	User	TA	Blockchain	CR	Vehicle	RSU	ZSP	AP	Sensor	DY	CK	eCK	AVISPA	ProVerif	Scyther	SPIN	Tamarin	ROR	BAN	ROM	Mao-Boyd	Ouafi-Phan	GNY	Comp	Comm
[49]	✔	✔	×	✔	×	×	×	×	×	×	×	✔	×	×	×	✔	×	×	×	×	×	✔	×	×	×	17.7939	3720
[64]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	✔	×	×	×	✔	×	×	×	×	✔	×	×	×	3.832	1456
[51]	✔	✔	×	✔	✔	×	×	×	×	×	×	✔	✔	×	✔	×	×	×	×	×	×	×	×	×	×	1.8	800
[65]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×	×	×	✔	12.447	3232
[66]	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	✔	×	×	×	×	31.158	2728
[23]	✔	✔	✔	×	×	✔	×	×	×	×	×	✔	✔	×	✔	×	×	×	×	×	×	✔	×	×	×	3.3873	1536
[67]	✔	✔	✔	×	×	×	×	×	×	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×	×	×	1.1628	3872
[55]	✔	×	✔	×	✔	×	×	×	×	×	×	✔	×	×	✔	×	×	×	×	×	✔	×	×	×	×	0.143	2048
[33]	✔	✔	×	×	×	×	×	×	×	×	×	✔	✔	×	×	✔	×	×	×	×	✔	×	×	×	×	13.5	1344
[68]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	✔	×	×	×	✔	×	×	×	×	✔	×	×	×	2.5	1456
[27]	✔	×	✔	×	✔	×	×	×	×	✔	×	✔	×	×	✔	×	×	×	×	✔	✔	×	×	×	×	61.607	3520
[69]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	×	×	×	✔	×	×	×	×	✔	×	×	×	×	902.0	3280
[70]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	×	×	✔	×	×	×	×	✔	✔	×	×	×	×	5.54	3744
[34]	✔	✔	×	×	×	×	×	×	×	×	×	✔	✔	×	✔	×	×	×	×	✔	×	×	×	×	×	7.5695	832
[71]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	✔	×	×	✔	×	×	×	✔	×	×	×	×	×	0.54	2176
[72]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	×	×	✔	×	×	×	×	×	×	×	×	×	×	27.02	1696
[39]	✔	✔	×	×	×	✔	×	×	×	×	×	×	×	✔	×	×	✔	×	×	×	×	×	×	×	×	1.576	1728
[73]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	×	×	×	×	✔	×	×	×	×	✔	×	×	×	21.055	1856
[52]	✔	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	✔	×	×	×	4.504	1728
[24]	✔	✔	✔	×	×	✔	×	×	×	×	×	✔	✔	×	×	×	✔	×	×	✔	×	×	×	×	×	11.628	2400
[74]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	0.11	1856
[35]	✔	✔	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	✔	×	×	✔	×	×	✔	×	×	130.15	1024
[79]	✔	✔	×	✔	×	×	×	×	✔	×	×	×	✔	×	×	✔	×	×	×	×	×	×	×	×	×	13.66	1760
[50]	✔	✔	×	✔	×	×	×	×	×	×	×	×	✔	×	×	✔	×	×	×	✔	×	×	×	×	×	8.343	1280
[45]	✔	×	×	✔	✔	×	✔	✔	×	×	×	✔	✔	×	✔	×	×	×	×	×	✔	×	×	×	×	3.094	2560
[36]	✔	✔	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	×	✔	×	×	✔	×	×	×	2.87	864
[76]	✔	✔	✔	×	×	×	✔	×	×	×	×	✔	✔	×	✔	×	×	×	×	✔	×	×	×	×	×	3.852	2720
[40]	✔	✔	×	×	×	✔	×	×	×	×	×	✔	✔	✔	×	×	✔	×	×	×	×	✔	×	×	×	0.51591	1664
[20]	✔	✔	✔	×	×	×	×	×	×	×	×	✔	×	×	✔	×	×	×	×	✔	×	×	×	×	×	12.371	2048
[53]	✔	✔	✔	✔	×	×	×	×	×	×	×	✔	✔	×	×	×	✔	×	×	×	×	✔	×	×	×	25.282	2240
[44]	✔	×	×	✔	×	×	✔	✔	×	×	×	✔	✔	×	×	✔	×	×	×	✔	×	×	×	×	×	1.92	336

Table 12 – Cryptographic primitives used in the analyzed studies (Part 1)

Article	Hash Function	XOR Operation	PUF	ECC	Fuzzy Extractor	Symmetric Enc.	Digital Signature	HCC	Digital Certificates	Asymmetric Enc.	Chaotic Map	SSS	Symmetric Bivariate Poly.	Kyber algorithm	Polynomial-based Key Dist.	BCH
[19]	✔	✔	×	✔	×	×	×	×	✔	×	×	×	×	×	×	×
[41]	✔	×	×	✔	×	×	✔	×	×	×	×	×	✔	×	×	×
[29]	✔	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×
[37]	✔	✔	×	✔	×	✔	×	×	×	×	×	×	×	×	×	×
[56]	✔	✔	×	×	✔	✔	×	×	×	×	✔	×	×	×	×	×
[8]	✔	✔	✔	×	✔	✔	×	×	×	×	×	×	×	×	×	×
[42]	✔	✔	×	✔	×	×	✔	×	×	×	×	×	×	×	×	×
[4]	✔	✔	✔	×	×	×	×	×	×	×	✔	×	×	×	×	×
[78]	✔	✔	×	✔	×	×	×	×	×	×	×	×	×	×	×	×
[57]	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×
[28]	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×
[25]	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×
[77]	✔	✔	✔	×	✔	×	×	×	×	×	×	×	×	×	×	×
[58]	✔	✔	×	×	✔	×	×	×	×	×	×	×	×	×	×	×
[38]	✔	✔	×	✔	✔	×	✔	×	×	×	×	×	×	×	×	×
[30]	✔	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×
[75]	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×
[59]	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[47]	✔	✔	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×
[43]	✔	✔	✔	✔	×	×	✔	×	×	×	×	×	×	×	×	×
[31]	✔	×	×	✔	×	×	×	×	×	×	×	×	×	✔	×	×
[60]	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×
[46]	✔	✔	×	✔	×	×	✔	×	×	×	×	×	×	×	×	×
[48]	✔	✔	×	✔	×	×	×	×	✔	×	×	×	×	×	×	×
[61]	✔	✔	✔	×	✔	×	×	×	×	×	×	×	×	×	×	×
[62]	✔	✔	×	×	✔	×	×	×	×	×	×	×	×	×	×	×
[32]	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×
[54]	✔	✔	×	✔	×	×	×	×	✔	×	×	×	×	×	×	×
[63]	✔	✔	×	×	✔	×	×	×	×	×	×	×	×	×	×	×
[26]	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×

Table 13 – Cryptographic primitives used in the analyzed studies (Part 2)

Article	Hash Function	XOR Operation	PUF	ECC	Fuzzy Extractor	Symmetric Enc.	Digital Signature	HCC	Digital Certificates	Asymmetric Enc.	Chaotic Map	SSS	Symmetric Bivariate Poly.	Kyber algorithm	Polynomial-based Key Dist.	BCH
[49]	✔	✔	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×
[64]	✔	✔	×	×	✔	×	×	✔	×	×	×	×	×	×	×	×
[51]	✔	×	✔	✔	✔	×	×	×	×	×	×	✔	×	×	×	×
[65]	✔	✔	×	×	×	×	×	×	×	✔	×	×	×	×	×	×
[66]	✔	✔	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×
[23]	✔	✔	×	×	×	×	×	✔	×	×	×	×	×	×	×	×
[67]	✔	✔	✔	×	✔	✔	×	×	×	×	×	×	×	×	×	×
[55]	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×
[33]	✔	✔	×	✔	×	×	×	×	×	×	×	×	×	×	×	×
[68]	✔	✔	✔	×	×	×	×	✔	×	×	×	×	×	×	×	×
[27]	✔	✔	×	✔	✔	✔	×	×	×	×	×	×	×	×	×	×
[69]	✔	✔	×	✔	×	×	✔	×	×	×	×	×	×	×	×	×
[70]	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×
[34]	✔	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×
[71]	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[72]	✔	✔	×	×	✔	×	×	×	×	×	×	×	×	×	✔	×
[39]	✔	×	×	×	×	✔	×	×	×	×	✔	×	×	×	×	×
[73]	✔	✔	×	✔	✔	✔	×	×	×	×	×	×	×	×	×	×
[52]	✔	✔	×	×	✔	×	×	×	×	×	×	×	×	×	×	×
[24]	✔	✔	✔	×	✔	✔	×	×	×	×	×	×	×	×	×	×
[74]	✔	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	×
[35]	✔	✔	✔	×	✔	×	×	×	×	×	×	×	×	×	×	×
[79]	✔	×	×	✔	×	×	✔	×	×	×	×	×	×	×	×	×
[50]	✔	×	×	×	×	✔	✔	×	×	✔	×	×	×	×	×	×
[45]	✔	✔	×	✔	×	×	✔	×	×	×	×	×	×	×	×	×
[36]	✔	✔	×	×	×	✔	×	×	×	×	✔	✔	×	×	×	×
[76]	✔	✔	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×
[40]	✔	✔	✔	×	✔	✔	×	×	×	×	×	×	×	×	×	✔
[20]	✔	✔	✔	×	✔	×	×	×	×	×	×	×	×	×	×	×
[53]	✔	✔	×	✔	✔	✔	×	×	×	×	×	×	×	×	×	×
[44]	✔	×	×	×	×	×	×	✔	✔	×	×	×	×	×	×	×

Table 14 – Informal security properties and attacks considered in the analyzed studies (Part 1)

Art.	Replay Attack	Impersonation	MitM	Anonymity	Untraceability	Drone Capture	Forward Secrecy	Insider attack	Mutual Auth.	ESL	DoS	Smart Device Stolen	Sess. Key Disclosure	Stolen Verifier	Desynchronization	Backward Secrecy	Password Update	Modification	Session Agreement	Password Guessing	Biometric Update	Cloning	Tampering	Privacy Protection	Eavesdropping	Session-key sec.	Spoofing	Unlinkability	Integrity	Brute Force	Private Key Leakage	Confidentiality	Side Channel	Unforgeability	ML Attack
[19]	✔	✔	✔	✔	✔	✔	×	✔	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[41]	✔	✔	✔	×	×	✔	×	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[29]	✔	✔	✔	×	×	✔	✔	✔	×	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✔	×	×	×	×
[37]	✔	✔	×	✔	×	×	✔	×	✔	×	✔	×	×	×	×	✔	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	✔	×	×	×	×
[56]	✔	✔	✔	✔	✔	×	×	✔	×	✔	×	×	×	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[8]	✔	✔	✔	✔	✔	✔	×	✔	×	✔	✔	×	×	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[42]	✔	×	×	✔	✔	×	×	×	✔	×	×	×	×	×	✔	×	×	✔	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×
[4]	✔	✔	✔	×	×	✔	×	×	×	×	×	×	✔	×	×	×	×	✔	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×
[78]	✔	✔	✔	✔	✔	×	✔	✔	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[57]	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[28]	✔	×	✔	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✔	✔	×	×	×	✔
[25]	✔	✔	✔	✔	✔	×	✔	×	×	×	✔	×	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	×
[77]	✔	✔	✔	✔	×	✔	×	×	×	×	×	×	✔	×	×	×	×	✔	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×
[58]	✔	✔	✔	✔	✔	✔	×	×	✔	✔	×	✔	×	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[38]	✔	✔	✔	✔	✔	×	✔	×	✔	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×
[30]	✔	✔	✔	×	✔	×	✔	×	✔	×	×	×	✔	×	×	✔	×	×	×	×	×	✔	×	✔	×	×	×	×	×	×	×	×	×	×	×	×
[75]	×	✔	×	×	×	✔	×	✔	×	×	×	✔	✔	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[59]	✔	✔	×	✔	✔	✔	×	×	✔	×	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[47]	✔	✔	✔	✔	✔	✔	✔	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[43]	✔	✔	✔	✔	✔	✔	✔	×	×	✔	✔	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[31]	✔	×	×	×	×	×	×	×	✔	×	×	×	✔	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[60]	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	×	✔	✔	×	×	×	×	✔	✔	✔	×	✔	✔	×	✔	×	×	×	✔	×	×	×	×	×	×	×
[46]	✔	✔	✔	✔	✔	✔	×	✔	×	✔	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[48]	✔	×	✔	×	×	✔	✔	✔	✔	×	✔	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[61]	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	×	×	×	×	×	✔	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[62]	✔	×	✔	✔	✔	✔	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[32]	✔	✔	✔	✔	✔	×	✔	×	✔	×	×	✔	×	×	×	✔	×	×	✔	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×
[54]	✔	✔	✔	×	✔	×	×	×	✔	✔	×	✔	✔	×	✔	×	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[63]	✔	✔	✔	✔	✔	✔	✔	×	×	✔	×	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[26]	×	✔	×	✔	✔	✔	✔	✔	✔	×	×	✔	×	×	✔	✔	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[49]	✔	✔	✔	×	×	✔	✔	✔	✔	×	✔	✔	×	✔	×	×	×	✔	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×

Table 15 – Informal security properties and attacks considered in the analyzed studies (Part 2)

Art.	Replay Attack	Impersonation	MitM	Anonymity	Untraceability	Drone Capture	Forward Secrecy	Insider attack	Mutual Auth.	ESL	DoS	Smart Device Stolen	Sess. Key Disclosure	Stolen Verifier	Desynchronization	Backward Secrecy	Password Update	Modification	Sess. Key Agreement	Password Guessing	Biometric Update	Cloning	Tampering	Privacy Protection	Eavesdropping	Session-key sec.	Spoofing	Unlinkability	Integrity	Brute Force	Private Key Leakage	Confidentiality	Side Channel	Unforgeability	ML Attack
[64]	✔	✔	✔	×	✔	✔	✔	×	✔	×	✔	✔	✔	×	×	×	✔	×	✔	×	✔	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×
[51]	✔	✔	✔	×	✔	✔	✔	×	×	×	×	×	×	×	✔	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	✔
[65]	✔	✔	✔	✔	×	×	✔	✔	×	×	✔	×	×	✔	✔	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×
[66]	✔	×	×	✔	✔	✔	×	✔	×	×	✔	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✔	×	×
[23]	✔	✔	✔	✔	✔	✔	✔	×	×	×	✔	✔	✔	×	×	×	×	✔	✔	×	×	×	×	✔	×	×	×	×	✔	×	×	×	×	×	×
[67]	✔	✔	✔	✔	✔	×	✔	✔	×	×	×	✔	×	✔	×	✔	×	×	×	×	×	×	×	×	✔	✔	×	×	×	×	×	×	×	×	×
[55]	✔	✔	✔	✔	✔	✔	×	✔	×	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[33]	✔	×	×	✔	×	×	✔	✔	×	×	×	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[68]	✔	✔	✔	✔	✔	✔	✔	×	✔	×	✔	✔	✔	×	×	×	×	✔	×	×	×	×	✔	×	✔	×	×	×	✔	×	×	×	×	×	×
[27]	✔	✔	✔	✔	✔	×	×	×	×	✔	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[69]	✔	×	×	✔	×	✔	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✔	✔	✔	×	×	×	×	×	×	×
[70]	✔	✔	✔	×	×	✔	✔	✔	✔	✔	×	×	×	✔	×	×	×	×	×	✔	×	×	×	✔	×	✔	×	×	×	×	×	×	×	×	×
[34]	✔	✔	✔	×	×	×	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	✔	×	✔	×
[71]	✔	✔	×	✔	×	✔	✔	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	×	×
[72]	×	✔	×	✔	✔	✔	×	✔	×	×	✔	✔	×	×	×	×	✔	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[39]	✔	✔	×	✔	×	×	✔	×	✔	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	×	×
[73]	✔	✔	✔	✔	✔	✔	×	✔	×	✔	✔	✔	×	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[52]	✔	×	✔	×	✔	×	×	✔	×	×	✔	×	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✔	×	×
[24]	✔	✔	✔	✔	✔	✔	×	×	×	✔	✔	×	×	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[74]	✔	✔	✔	✔	✔	×	✔	×	✔	×	✔	✔	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[35]	✔	×	✔	✔	✔	×	✔	×	✔	✔	×	×	×	×	✔	×	×	×	✔	×	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×
[79]	✔	✔	✔	×	×	×	✔	✔	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✔	×	×	×	×
[50]	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	×	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✔	×	×	✔	×	×	×
[45]	✔	✔	✔	✔	✔	✔	×	✔	×	✔	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[36]	✔	✔	×	×	×	×	✔	×	✔	×	×	×	×	×	×	✔	×	✔	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×	×
[76]	✔	×	×	✔	×	✔	✔	✔	✔	×	×	✔	×	×	×	×	✔	×	×	×	✔	×	×	×	×	✔	×	×	×	✔	×	×	×	✔	×
[40]	×	✔	×	✔	✔	✔	✔	×	×	✔	×	×	×	✔	✔	✔	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×
[20]	✔	✔	✔	✔	×	✔	×	×	✔	✔	×	×	✔	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[53]	✔	✔	✔	✔	✔	✔	×	✔	×	✔	✔	✔	×	×	×	×	✔	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×	×
[44]	✔	✔	✔	✔	×	×	×	✔	×	✔	×	×	×	×	×	×	×	×	×	×	×	✔	×	×	×	×	×	×	×	×	×	×	×	×	×

B Automated Analysis Tool Code

B.1 AVISPA

B.1.1 PMAP D2Z

```
role drone(
 DronePID, ZSP : agent,
 CRP  : symmetric_key,
 Hash  : hash_func,
 Snd, Rcv : channel(dy))
played_by DronePID
def=
 local
  State     : nat,
  Nd,Ns,PID    : text,
  Cd,Rd,M1,M2,M3,M4,MAC1,MAC2,MAC34,SK : message

 init
  State := 1

 transition
  1. State = 1 /\ Rcv(start) =|>
     State':= 3 /\ Nd'   := new()
    /\ M1'   := {DronePID.ZSP.Nd'}_CRP
    /\ MAC1' := Hash(M1'.Nd')
    /\ Snd(M1'.MAC1')

  3. State = 3 /\ Rcv({DronePID.ZSP.Nd.Ns'}_CRP.MAC2') =|>
     State':= 5 /\ Nd' := new()
    /\ Cd' := {Ns'.Nd'}_CRP
    /\ Rd' := Hash(Cd')
    /\ CRP' := (Cd'.Rd')
    /\ M3' := {DronePID.ZSP.Ns'.Nd'}_CRP
    /\ M4' := {DronePID.ZSP.Ns'.Nd'.Rd'}_CRP
    /\ MAC34':= Hash(M3'.M4'.Nd'.Rd')
```

```
    /\ SK' := xor(Hash(Ns'), Hash(Nd'))
    /\ Snd(M3'.M4'.MAC34')
    /\ secret(Ns',sec_1,{DronePID,ZSP})
    /\ secret(Nd',sec_1,{DronePID,ZSP})
    /\ secret(Rd',sec_2,{DronePID,ZSP})
    /\ request(DronePID,ZSP,auth_1,Nd)
    /\ witness(DronePID,ZSP,auth_2,Ns')
    /\ witness(DronePID,ZSP,auth_3,SK')

end role

role zsp(
 ZSP,DronePID : agent,
 CRP  : symmetric_key,
 Hash  : hash_func,
 Snd, Rcv : channel(dy))
played_by ZSP
def=
 local
  State     : nat,
  Nd,Ns,PID    : text,
  Cd,Rd,M1,M2,M3,M4,MAC1,MAC2,MAC34,SK : message
 init
  State := 2

 transition
  2. State  = 2  /\ Rcv({DronePID.ZSP.Nd'}_CRP.MAC1') =|>
     State':= 4 /\ Ns'  := new()
    /\ M2'   := {DronePID.ZSP.Nd'.Ns'}_CRP
    /\ MAC2' := Hash(M2'.Nd'.Ns')
    /\ Snd(M2'.MAC2')
    /\ witness(ZSP,DronePID,auth_1,Nd')

  4. State   = 4  /\ Rcv({DronePID.ZSP.Ns.Nd'}_CRP.{DronePID.ZSP.Ns.Nd'.Rd'}_CRP.MA
     State' := 6 /\ Cd'    := {Ns.Nd'}_CRP
    /\ Rd'   := Hash(Cd')
    /\ CRP'   := (Cd'.Rd')
```

```
    /\ SK'   := xor(Hash(Ns), Hash(Nd'))
    /\ request(ZSP,DronePID,auth_2,Ns)
    /\ request(ZSP,DronePID,auth_3,SK')
end role

role session(
 DronePID,ZSP : agent,
 CRP  : symmetric_key,
 Hash  : hash_func)
def=
 local
  SND2,RCV2,SND1,RCV1:channel(dy)

 composition
  drone(DronePID,ZSP,CRP,Hash,SND1,RCV1)
  /\ zsp(ZSP,DronePID,CRP,Hash,SND2,RCV2)
end role

role environment()
def=
 const
  dpid,gs     :agent,
  crp     :symmetric_key,
  h     :hash_func,
  auth_2,auth_1,auth_3,sec_1,sec_2 :protocol_id

 intruder_knowledge = {dpid,gs,h,crp}

 composition
  session(dpid,gs,crp,h)
  /\ session(dpid,gs,crp,h)
  /\ session(i,gs,crp,h)
  /\ session(dpid,i,crp,h)

end role

goal
```

```
 secrecy_of sec_1
 secrecy_of sec_2
 authentication_on auth_1
 authentication_on auth_2
 authentication_on auth_3
end goal

environment()
```

B.1.2 PMAP D2D

```
role droneA(
 DronePIDa,DronePIDb, ZSP : agent,
 CRPa    : symmetric_key,
 Hash    : hash_func,
 Snd, Rcv   : channel(dy))
played_by DronePIDa
def=
 local
  State     : nat,
  Na,Nb,Nsa,Ra,Ca    : text,
  M1,M2,M4,M5,MAC12,MAC3,MAC45,MAC11,SK : message
 init
  State := 1

 transition
  1. State = 1  /\ Rcv(start) =|>
     State':= 3  /\ Na' := new()
    /\ M1' := {DronePIDa.ZSP.Na'}_CRPa
    /\ M2' := {DronePIDa.ZSP.Na'.DronePIDb}_CRPa
    /\ MAC12' := Hash(M1'.M2'.Na'.DronePIDb)
    /\ Snd(M1'.M2'.MAC12')

  3. State = 3 /\ Rcv({DronePIDa.ZSP.DronePIDb.Na.Nsa'}_CRPa.MAC3') =|>
     State':= 7 /\ Na' := new()
    /\ Ca' := {Nsa'.Na'}_CRPa
```

```
    /\ Ra' := Hash(Ca')
    /\ CRPa':= (Ca'.Ra')
    /\ M4' := {DronePIDa.ZSP.DronePIDb.Nsa'.Na'}_CRPa
    /\ M5' := {DronePIDa.ZSP.DronePIDb.Nsa'.Na'.Ra'}_CRPa
    /\ MAC45' := Hash(M4'.M5'.Na'.Ra')
    /\ Snd(M4'.M5'.MAC45')

  7. State = 7 /\ Rcv({DronePIDa.ZSP.DronePIDb.Na.Nb'}_CRPa.MAC11') =|> SK' := xor(
    /\ request(DronePIDa,DronePIDb,auth_1,SK')

end role

role zsp(
 DronePIDa,DronePIDb, ZSP : agent,
 CRPa, CRPb   : symmetric_key,
 Hash    : hash_func,
 Snd, Rcv   : channel(dy))
played_by ZSP
def=
 local
  State     : nat,
  Na,Nb,Nsa,Nsb,Ra,Ca,Rb,Cb  : text,
  M3,M6,M7,M8,M11,
  MAC12,MAC3,MAC45,MAC678,MAC11,MAC910 : message
 init
  State := 2

 transition
  2. State = 2 /\ Rcv({DronePIDa.ZSP.Na'}_CRPa.{DronePIDa.ZSP.Na'.DronePIDb}_CRPa.M
     State':= 4 /\ Nsa' := new()
    /\ Nsb' := new()
    /\ M3' := {DronePIDa.ZSP.DronePIDb.Na'.Nsa'}_CRPa
    /\ MAC3':= Hash(M3'.Na'.Nsa')
    /\ Snd({DronePIDa.ZSP.DronePIDb.Na'.Nsa'}_CRPa.MAC3')

  4. State = 4 /\ Rcv({DronePIDa.ZSP.DronePIDb.Nsa.Na'}_CRPa.{DronePIDa.ZSP.DronePI
     State':= 6 /\ M6' := {DronePIDb.ZSP.Nsb}_CRPb
```

```
    /\ M7' := {DronePIDb.ZSP.Nsb.Na'}_CRPb
    /\ M8' := {DronePIDb.ZSP.Nsb.Na'.DronePIDa}_CRPb
    /\ MAC678' := Hash(M6'.M7'.M8'.Nsb.Na'.DronePIDa)
    /\ Ca' := {Nsa.Na'}_CRPa
    /\ Ra' := Hash(Ca')
    /\ CRPa':= (Ca'.Ra')
    /\ Snd(M6'.M7'.M8'.MAC678')

  6. State = 6 /\ Rcv({DronePIDb.ZSP.DronePIDa.Nsb.Nb'}_CRPb.{DronePIDb.ZSP.DronePI
    /\ Rb' := Hash(Cb')
    /\ CRPb':= (Cb'.Rb')
    /\ M11' := {DronePIDa.ZSP.DronePIDb.Na.Nb'}_CRPa
    /\ MAC11' := Hash(M11'.Na.Nb')
    /\ Snd(M11'.MAC11')

end role

role droneB(
 DronePIDa,DronePIDb, ZSP : agent,
 CRPb    : symmetric_key,
 Hash    : hash_func,
 Snd, Rcv   : channel(dy))
played_by DronePIDb
def=
 local
  State   : nat,
  Na,Nb,Nsa,Nsb,Rb,Cb : text,
  M9,M10,MAC678,MAC910,SK : message
 init
  State := 5

 transition
  5. State = 5 /\ Rcv({DronePIDb.ZSP.Nsb'}_CRPb.{DronePIDb.ZSP.Nsb'.Na'}_CRPb.{Dron
    /\ Cb' := {Nsb.Nb'}_CRPb
    /\ Rb' := Hash(Cb')
    /\ CRPb':= (Cb'.Rb')
    /\ M9' := {DronePIDb.ZSP.DronePIDa.Nsb'.Nb'}_CRPb
```

```
    /\ M10' := {DronePIDb.ZSP.DronePIDa.Nsb'.Nb'.Rb'}_CRPb
    /\ MAC910' := Hash(M9'.M10'.Nb'.Rb')
    /\ Snd(M9'.M10'.MAC910')
    /\ SK' := xor(Hash(Nb'), Hash(Na'))
    /\ witness(DronePIDb,DronePIDa,auth_1,SK')
end role

role session(
 DronePIDa, DronePIDb,ZSP : agent,
 CRPa,CRPb   : symmetric_key,
 Hash    : hash_func)
def=
 local
  SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy)

 composition
  droneA(DronePIDa,DronePIDb,ZSP,CRPa,Hash,SND1,RCV1)
  /\ zsp(DronePIDa,DronePIDb,ZSP,CRPa,CRPb,Hash,SND2,RCV2)
  /\ droneB(DronePIDa,DronePIDb,ZSP,CRPb,Hash,SND3,RCV3)
end role

role environment()
def=
 const
  dpida,dpib,gs :agent,
  crpa,crpb :symmetric_key,
  h  :hash_func,
  auth_1  :protocol_id

 intruder_knowledge = {dpida,dpib,gs,h}

 composition
  session(dpida,dpib,gs,crpa,crpb,h)
  /\ session(dpida,dpib,gs,crpa,crpb,h)
  /\ session(i,dpib,gs,crpa,crpb,h)
  /\ session(dpida,i,gs,crpa,crpb,h)
  /\ session(dpida,dpib,i,crpa,crpb,h)
```

```
end role

goal
 authentication_on auth_1
end goal

environment()
```

B.2 PROVERIF

B.2.1 PMAP D2Z

```
free c: channel.
free s: bitstring [private].
free crp: bitstring [private].

fun PUF(bitstring, bitstring): bitstring [private].
fun senc(bitstring, bitstring): bitstring.
fun hash(bitstring): bitstring.

reduc forall m: bitstring, k: bitstring; sdec(senc(m,k), k) = m.

query attacker(s).

event acceptsDrone(bitstring).
event acceptsZsp(bitstring).
event termDrone(bitstring).
event termZsp(bitstring).

query x: bitstring; event(termDrone(x)) ==> event(acceptsZsp(x)).
query x: bitstring; inj-event(termZsp(x)) ==> event(acceptsDrone(x)).

let drone(pid: bitstring, zspid: bitstring) =
 (* step 01 *)
 new nd_1: bitstring;
 let m1 = senc((pid, zspid, nd_1), crp) in
 let mac1 = hash((m1, nd_1)) in
```

```
 out(c, (m1, mac1));

 (* step 2 *)
 in(c, m:bitstring);
 let (m2: bitstring, mac2: bitstring) = m in
 let (_: bitstring, _: bitstring, nd_1_r: bitstring, ns_1: bitstring) = sdec(m2, cr
 if mac2 = hash((m2, nd_1_r, ns_1)) then
 new nd_2: bitstring;
 let cd = PUF((ns_1, nd_2), crp) in
 let rd = hash(cd) in
 let crp_new = (cd, rd) in
 let m3 = senc((pid, zspid, ns_1, nd_2), crp_new) in
 let m4 = senc((pid, zspid, ns_1, nd_2, rd), crp_new) in
 let mac34 = hash((m3, m4, nd_2, rd)) in
 out(c, (m3,m4,mac34));
 let k = (hash(ns_1), hash(nd_2)) in

 (* mutual authentication *)
 event acceptsDrone(k);
 in(c, x:bitstring);
 let z = sdec(x,k) in
 event termDrone(k);

 0.

let zsp(zspid: bitstring, pid: bitstring) =
 (* step 01 *)
 in(c, m: bitstring);
 let (m1: bitstring, mac1: bitstring) = m in
 let (_: bitstring, _:bitstring, nd_1:bitstring) = sdec(m1, crp) in
 if mac1 = hash((m1, nd_1)) then
 new ns_1: bitstring;
 let m2 = senc((pid, zspid, nd_1, ns_1), crp) in
 let mac2 = hash((m2, nd_1, ns_1)) in
 out(c, (m2, mac2));

 (* step 02 *)
```

```
 in(c, m_2: bitstring);
 let (m3:bitstring, m4:bitstring, mac34:bitstring) = m_2 in
 let (pid_2: bitstring, zspid_2: bitstring, ns_1_r: bitstring, nd_2: bitstring, rd_
 if mac34 = hash((m3, m4, nd_2, rd_r)) then
 let cd = senc((ns_1_r, nd_2), crp) in
 let rd = hash(cd) in
 let crp_new = (cd, rd) in
 let k = (hash(ns_1_r), hash(nd_2)) in

 (* mutual authentication *)
 event acceptsZsp(k);
 out(c, senc(s,k));
 event termZsp(k);
 0.

process
 new dronePID: bitstring;
 new zspid: bitstring;

 out(c, dronePID);
 out(c, zspid);

 ( (!drone(dronePID, zspid)) | (!zsp(zspid, dronePID)) )
```

B.2.2 PMAP D2D

```
free c: channel.
free s: bitstring [private].
free crpa: bitstring [private].
free crpb: bitstring [private].

fun PUF(bitstring, bitstring): bitstring [private].
fun senc(bitstring, bitstring): bitstring.
fun hash(bitstring): bitstring.

reduc forall m: bitstring, k: bitstring; sdec(senc(m,k), k) = m.
```

```
query attacker(s).

let droneA(pida: bitstring, pidb: bitstring, zspid: bitstring) =
 (* step 01 *)
 new na_s1: bitstring;
 let m1 = senc((pida, zspid, na_s1), crpa) in
 let m2 = senc((pida, zspid, na_s1, pidb), crpa) in
 let mac12 = hash((m1,m2,na_s1,pidb)) in
 out(c, (m1,m2,mac12));

 (* step 03 *)
 in(c, m_s3: bitstring);
 let (m3: bitstring, mac3: bitstring) = m_s3 in
 let (_: bitstring, _: bitstring, _: bitstring, _: bitstring, nsa: bitstring) = sde
 new na_s3: bitstring;
 let ca = senc((nsa,na_s3), crpa) in
 let ra = hash(ca) in
 let crpa_new = (ca, ra) in
 let m4 = senc((pida, zspid, pidb, nsa, na_s3), crpa) in
 let m5 = senc((pida, zspid, pidb, nsa, na_s3, ra), crpa) in
 let mac45 = hash((m4, m5, na_s3, ra)) in
 out(c, (m4,m5,mac45));

 (* step 07 *)
 in(c, m_s7: bitstring);
 let (m11:bitstring,mac11:bitstring) = m_s7 in
 let (_:bitstring,_:bitstring,_:bitstring,_:bitstring,nb:bitstring) = sdec(m11, crp
 let k = (hash(nb), hash(na_s3)) in

 (* authentication *)
 out(c, senc(s, k));
 0.


let zsp(pida: bitstring, pidb: bitstring, zspid: bitstring) =
 (* step 02 *)
```

```
 in(c, m_s2: bitstring);
 let (m1:bitstring,m2:bitstring,mac12:bitstring) = m_s2 in
 let (_:bitstring, _:bitstring, na:bitstring, _:bitstring) = sdec(m2, crpa) in
 new nsa: bitstring;
 new nsb: bitstring;
 let m3 = senc((pida, zspid, pidb, na, nsa), crpa) in
 let mac3 = hash((m3, na, nsa)) in
 out(c, (m3,mac3));

 (* step 04 *)
 in(c, m_s4: bitstring);
 let (m4:bitstring,m5:bitstring,mac45:bitstring) = m_s4 in
 let (_:bitstring, _:bitstring, _:bitstring, _:bitstring, na_s3:bitstring, ra:bitst
 let m6 = senc((pidb, zspid, nsb), crpb) in
 let m7 = senc((pidb, zspid, nsb, na_s3), crpb) in
 let m8 = senc((pidb, zspid, nsb, na_s3, pida), crpb) in
 let mac678 = hash((m6, m7, m8, nsb, na_s3, pida)) in
 let ca = senc((nsa,na_s3), crpa) in
 let ra = hash(ca) in
 let crpa_new = (ca, ra) in
 out(c, (m6, m7, m8, mac678));

 (* step 06 *)
 in(c, m_s6: bitstring);
 let (m9: bitstring,m10: bitstring, mac910: bitstring) = m_s6 in
 let (_: bitstring,_: bitstring,_: bitstring,_: bitstring, nb: bitstring, _: bitstr
 let cb = senc((nsb, nb), crpb) in
 let rb = hash(cb) in
 let crpb_new = (cb, rb) in
 let m11 = senc((pida, zspid, pidb, na_s3, nb), crpa_new) in
 let mac11 = hash((m11, na_s3, nb)) in
 out(c, (m11, mac11));
 0.


let droneB(pida: bitstring, pidb: bitstring, zspid: bitstring) =
 (* step 05 *)
```

```
 in(c, m_s5: bitstring);
 let (m6:bitstring,m7:bitstring,m8:bitstring,mac678:bitstring) = m_s5 in
 let (_:bitstring, _:bitstring, nsb:bitstring, na_s3:bitstring, _:bitstring) = sdec
 new nb: bitstring;
 let cb = senc((nsb,nb), crpb) in
 let rb = hash(cb) in
 let crpb_new = (cb, rb) in
 let m9 = senc((pidb,zspid,pida,nsb,nb), crpb) in
 let m10 = senc((pidb,zspid,pida,nsb,nb, rb), crpb) in
 let mac910 = hash((m9, m10, nb, rb)) in
 out(c, (m9,m10, mac910));
 let k = (hash(nb), hash(na_s3)) in

 (* authentication *)
 in(c, x:bitstring);
 let z = sdec(x,k) in
 0.

process
 new pida: bitstring;
 new pidb: bitstring;
 new zspid: bitstring;

 out(c, pida);
 out(c, pidb);
 out(c, zspid);

 ( (!droneA(pida, pidb, zspid)) | (!zsp(pida, pidb, zspid)) | (!droneB(pida, pidb,
```

B.3 SCYTHER

B.3.1 PMAP D2Z

```
hashfunction h;

protocol pmap-d2z(D, Z) {
```

```
role D {
 fresh Nd: Nonce;
 fresh Rd: Ticket;
 var Ns: Nonce;

 send_1(D,Z,{D,Z,Nd}k,h({D,Z,Nd}k,Nd));
 recv_2(Z,D,{D,Z,Nd,Ns}k,h({D,Z,Nd,Ns}k,Nd,Ns));

 fresh Nd2: Nonce;

 send_3(D,Z,{D,Z,Ns,Nd2}k, {D,Z,Ns,Nd2,Rd}k, h({D,Z,Ns,Nd2}k, {D,Z,Ns,Nd2,Rd}k, Nd

 claim(D, Secret, Nd);
 claim(D, Secret, Ns);
 claim(D, Secret, Rd);
 claim(D, Secret, Nd2);
 claim(D, Niagree);
 claim(D, Nisynch);
 claim(D, Alive);
 claim(D, Weakagree);
}

role Z {
 fresh Ns: Nonce;
 var Nd: Nonce;
 var Nd2: Nonce;
 var Rd: Ticket;

 recv_1(D,Z,{D,Z,Nd}k,h({D,Z,Nd}k,Nd));
 send_2(Z,D,{D,Z,Nd,Ns}k,h({D,Z,Nd,Ns}k,Nd,Ns));
 recv_3(D,Z,{D,Z,Ns,Nd2}k, {D,Z,Ns,Nd2,Rd}k, h({D,Z,Ns,Nd2}k, {D,Z,Ns,Nd2,Rd}k, Nd

 claim(Z, Secret, Ns);
 claim(Z, Secret, Nd);
 claim(Z, Secret, Nd2);
 claim(Z, Secret, Rd);
```

```
  claim(Z, Niagree);
  claim(Z, Nisynch);
  claim(Z, Alive);
  claim(Z, Weakagree);
 }
}
```

B.3.2 PMAP D2D

```
hashfunction h;

protocol pmap-d2d(DA, DB, Z) {
 role DA {
  var Nsa, Nb: Nonce;
  fresh Na, Na2, Ra: Nonce;

  send_1(DA,Z,{DA,Z,Na}k(DA,Z),{DA,Z,Na,DB}k(DA,Z),h({DA,Z,Na}k(DA,Z),{DA,Z,Na,DB}k
  recv_2(Z,DA,{DA,Z,DB,Na,Nsa}k(Z,DA),h({DA,Z,DB,Na,Nsa}k(Z,DA),Na,Nsa));
  send_3(DA,Z, {DA,Z,DB,Nsa,Na2}k(DA,Z), {DA,Z,DB,Nsa,Na2,Ra}k(DA,Z), h({DA,Z,DB,Ns
  recv_6(Z,DA, {DA,Z,DB,Na2,Nb}k(Z,DA),h({DA,Z,DB,Na2,Nb}k(Z,DA),Na2,Nb));

  claim(DA, Secret, Na2);
  claim(DA, Secret, Nb);
  claim(DA, Niagree);
  claim(DA, Nisynch);
  claim(DA, Alive);
  claim(DA, Weakagree);
 }

 role Z {
  var Na,Na2, Ra, Nb, Rb: Nonce;
  fresh Nsa,Nsb: Nonce;

  recv_1(DA,Z,{DA,Z,Na}k(DA,Z),{DA,Z,Na,DB}k(DA,Z),h({DA,Z,Na}k(DA,Z),{DA,Z,Na,DB}k
  send_2(Z,DA,{DA,Z,DB,Na,Nsa}k(Z,DA),h({DA,Z,DB,Na,Nsa}k(Z,DA),Na,Nsa));
  recv_3(DA,Z, {DA,Z,DB,Nsa,Na2}k(DA,Z), {DA,Z,DB,Nsa,Na2,Ra}k(DA,Z), h({DA,Z,DB,Ns
  send_4(Z,DB,{DB,Z,Nsb}k(Z,DB),{DB,Z,Nsb,Na2}k(Z,DB),{DB,Z,Nsb,Na2,DA}k(Z,DB),h({D
```

```
 recv_5(DB,Z,{DB,Z,DA,Nsb,Nb}k(DB,Z),{DB,Z,DA,Nsb,Nb,Rb}k(DB,Z),h({DB,Z,DA,Nsb,Nb}
 send_6(Z,DA, {DA,Z,DB,Na2,Nb}k(Z,DA),h({DA,Z,DB,Na2,Nb}k(Z,DA),Na2,Nb));

 claim(Z, Secret, Na2);
 claim(Z, Secret, Nb);
 claim(Z, Niagree);
 claim(Z, Nisynch);
 claim(Z, Alive);
 claim(Z, Weakagree);
}

role DB {
 var Nsb,Na2: Nonce;
 fresh Nb, Rb: Nonce;

 recv_4(Z,DB,{DB,Z,Nsb}k(Z,DB),{DB,Z,Nsb,Na2}k(Z,DB),{DB,Z,Nsb,Na2,DA}k(Z,DB),h({D
 send_5(DB,Z,{DB,Z,DA,Nsb,Nb}k(DB,Z),{DB,Z,DA,Nsb,Nb,Rb}k(DB,Z),h({DB,Z,DA,Nsb,Nb}

 claim(DB, Secret, Na2);
 claim(DB, Secret, Nb);
 claim(DB, Niagree);
 claim(DB, Nisynch);
 claim(DB, Alive);
 claim(DB, Weakagree);
}
}
```