ACADEMIA MILITAR

MILITARY ACADEMY

Secure Command, Control and Communications Systems (C3) for Army UxVs

Alferes-Aluno Tomás Neto Rebolo

Thesis to obtain the Master of Science Degree in

Military Electrical Engineering

Supervisors: Prof. António Manuel Raminhos Cordeiro Grilo
Doutor Carlos Nuno da Cruz Ribeiro

Examination Committee

Chairperson: Prof. Name of the Chairperson
Supervisor: Prof. António Manuel Raminhos Cordeiro Grilo
Members of the Committee: Prof. André Zuquete
TC Pedro Miguel Simões Roque Pena Madeira

October 2025

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Military Academy.

Acknowledgments

The success of this work was made possible by the help of many individuals. First, I thank my advisors, Professor António Grilo and Professor Carlos Ribeiro, for their availability, guidance, and for consistently pointing me in the right direction. I also thank the entire SIC-T team, especially Lt. Carlos Gomes, for welcoming me into their facilities, providing the HR-5000H radios, and working alongside me for three consecutive weeks during the experimental testing of the protocol. Their availability, expertise, and generosity were crucial to the results obtained. I am deeply grateful to my entire family, especially my parents, my sister, and my girlfriend, for their unwavering support throughout this year, always giving me the strength to go further and do my best, and for their patience and understanding during the long hours of the NC2S design and implementation. To my friends and comrades, thank you for the encouragement, companionship, and good humor that kept me going. Above all, I thank God for the strength and clarity that sustained me throughout this work.

Abstract

Unmanned Vehicles (UxVs) are increasingly used in modern military operations for reconnaissance, surveillance, and strike missions, enhancing situational awareness while reducing risk to personnel. Their affordability and rapid deployment have encouraged the adoption of commercial solutions. However, many rely on insecure protocols such as MAVLink, which lack authentication and encryption mechanisms.

To address these limitations and overcome the operational gap in the Portuguese Army Battlefield Management System (BMS), its current C4I system, which does not support secure UxV control or control handover, this thesis designed, implemented, and evaluated a new secure command-and-control architecture that ensures confidentiality, integrity, and authentication (CIA) while supporting real-time control delegation between Ground Control Stations (GCSs).

The proposed solution, named New Command and Control System (NC2S), enforces a zero-trust model integrating hierarchical credential-based privileges to regulate access and control among Tactical Commanders (TC), GCSs, and UxVs. It employs mutual Transport Layer Security (mTLS) with Elliptic Curve Digital Signature Algorithm (ECDSA) certificates and Elliptic Curve Diffie–Hellman (ECDH) key exchange, while message integrity is ensured through Hash-based Message Authentication Codes (HMAC). Multiple lightweight protocols were developed for credential management, key renewal, and control handover.

The NC2S prototype was experimentally validated over Wi-Fi and Rohde & Schwarz HR-5000H tactical radios. Results showed that HR-5000H links introduce latencies roughly two orders of magnitude higher than Wi-Fi but maintain stable communication with minimal message loss, making them suitable

for command-node links.

Keywords

Unmanned Vehicles (UxV); Secure Communication; Command and Control (C2); Mutual TLS (mTLS); Elliptic Curve Cryptography (ECC); Control Handover; C4I Systems; Digital Certificates; Battlefield Management System (BMS)

Resumo

Os Veículos Não Tripulados (UxV), acrónimo utilizado na tradução para a língua inglesa, têm vindo a tornar-se ativos essenciais nas operações militares devido às suas capacidades de reconhecimento, vigilância e ataque ao solo, que, em conjunto, reduzem a exposição dos soldados a ambientes de elevado risco. A sua relação custo/eficácia e rápida disponibilidade têm levado à adoção generalizada de soluções comerciais. No entanto, muitas destas soluções recorrem a protocolos inseguros, como o MAVLink, que não implementam mecanismos de autenticação e encriptação.

Com o objetivo de colmatar estas limitações e superar a lacuna operacional do Battlefield Management System (BMS), o sistema de Comando, Controlo, Comunicações, Computadores e Informações (C4I) atualmente utilizado pelo Exército Português, que não permite o controlo ou a transferência segura de controlo de UxVs, esta dissertação concebeu, implementou e avaliou uma nova arquitetura segura de comando e controlo que garante confidencialidade, integridade e autenticação (CIA), suportando a delegação de controlo em tempo real entre Estações de Controlo Terrestres (GCS).

A solução proposta, designada New Command and Control System (NC2S), aplica um modelo zero-trust com privilégios hierárquicos baseados em credenciais e utiliza mutual Transport Layer Security (mTLS) com certificados Elliptic Curve Digital Signature Algorithm (ECDSA) e troca de chaves Elliptic Curve Diffie–Hellman (ECDH). Foram desenvolvidos múltiplos protocolos leves para gestão de credenciais, renovação de chaves e transferência de controlo.

O protótipo do NC2S foi validado experimentalmente através de ligações Wi-Fi e rádios táticos Rohde & Schwarz HR-5000H. Os resultados demonstraram que os HR-5000H introduzem latências cerca de duas ordens de grandeza superiores às do Wi-Fi, mas mantêm comunicações estáveis e com perdas

mínimas, sendo adequados para ligações entre nós de comando.

Palavras Chave

Veículos Não Tripulados (UxV); Comunicação Segura; Comando e Controlo (C2); Mutual TLS (mTLS); Criptografia de Curvas Elípticas (ECC); Transferência de Controlo; Sistemas C4I; Certificados Digitais

Contents

List of Figures

List of Tables

Acronyms

AEAD Authenticated Encryption with Associated Data.

AES Advanced Encryption Standard.

ALE Automatic Link Establishment.

AM Amplitude Modulation.

ARP Address Resolution Protocol.

AS Access Stratum.

ATS Authentication Server.

BMS Battlefield Management System.

C2 Command and Control.

C3 Command, Control and Communications.

C4I Command, Control, Communications, Computers, and Intelligence.

CA Certificate Authority.

CBC Cipher Block Chaining.

CCM Counter with CBC-MAC.

CertRL Certificate Revocation List.

CIA Confidentiality, Integrity, and Authentication.

CLPKC Certificateless Group Key Agreement with Constant Rounds.

CN Common Name.

COMSEC Communication Security.

COP Common Operational Picture.

CPM Continuous Phase Modulation.

CPU Central Processing Unit.

CRC Cyclic Redundancy Check.

CredRL Credential Revocation List.

CRL Certificate Revocation List.

CT Commander Terminal.

CTR Counter Mode.

DES Data Encryption Standard.

DoD Department of Defense.

DoS Denial of Service.

DSS Dismounted Soldier System.

ECC Elliptic Curve Cryptography.

ECDH Elliptic Curve Diffie-Hellman.

ECDSA Elliptic Curve Digital Signature Algorithm.

EDA European Defense Agency.

FEC Forward Error Correction.

FM Frequency Modulation.

GCM Galois/Counter Mode.

GCS Ground Control Station.

GPS Global Positioning System.

GRS Ground Relay Station.

GSS Ground Station Server.

GUI Graphical User Interface.

HF High Frequency.

HIGHT High Security and Lightweight.

HITL Hardware In The Loop.

HKDF HMAC-based Key Derivation Function.

HMAC Hash-based Message Authentication Code.

HMS High Management System.

HTTP Hypertext Transfer Protocol.

ICMP Internet Control Message Protocol.

INTERACT Intelligent Network for Teams of Exploratory Robots with Adaptive Collaborative Technologies.

IP Internet Protocol.

IPsec Internet Protocol Security.

ISM Industrial, Scientific, and Medical.

JAUS Joint Architecture for Unmanned Systems.

JSON JavaScript Object Notation.

L-TIDS Link-Tactical Information Distribution System.

LAN Local Area Network.

LC2EVO Lightweight Command and Control Evolution.

LC2IS Land Command and Control Information Service.

LOI Level of Interoperability.

LTE Long-Term Evolution.

MAC Message Authentication Code.

MANET Mobile Ad hoc Network.

MAVLink Micro Air Vehicle Link.

MIL-STD-461E Military Standard for Electromagnetic Interference Characteristics.

MIL-STD-810E Military Standard for Environmental Engineering Considerations.

MITM Man-in-the-Middle.

MME Mobility Management Entity.

MOMU Multi-Operator Multi-UAV.

mTLS mutual Transport Layer Security.

MTU Maximum Transmission Unit.

NATO North Atlantic Treaty Organization.

NC2S New Command and Control System.

NETSEC Network Security.

NTP Network Time Protocol.

OCB Offset Codebook Mode.

OS Operating System.

PADR Preparatory Action for Defence Research.

PKI Public Key Infrastructure.

PUF Physically Unclonable Function.

RA Registration Authority.

RC4 Rivest Cipher 4.

RCE Remote Command Execution.

RDE Army Data Network.

RSA Rivest–Shamir–Adleman.

RTO Retransmission Timeout.

RTT Round-Trip Time.

SECOM H Secure Communications Hopping.

SECOM-V Secure Communications Mode V.

SHA-256 Secure Hash Algorithm 256-bit.

SICCE Sistema de Informação de Comando e Controlo do Exército.

SITL Software In The Loop.

SIV Synthetic Initialization Vector.

SSB Single Sideband.

STANAG Standardization Agreement.

TAK Tactical Assault Kit.

TC Tactical Commander.

TCP Transmission Control Protocol.

TDMA Time Division Multiple Access.

TGS Ticket-Granting Server.

TGT Ticket-Granting Ticket.

TLS Transport Layer Security.

TRANSEC Transmission Security.

UART Universal Asynchronous Receiver-Transmitter.

UAV Unmanned Aerial Vehicle.

UDP User Datagram Protocol.

UGV Unmanned Ground Vehicle.

UHF Ultra High Frequency.

USV Unmanned Surface Vehicle.

UUV Unmanned Underwater Vehicle.

UWB Ultra Wide Band.

UxS Unmanned Systems.

UxV Unmanned Vehicle.

VHF Very High Frequency.

VPN Virtual Private Network.

WAN Wide Area Network.

Wi-Fi Wireless Fidelity.

WPA2-Enterprise Wi-Fi Protected Access 2 – Enterprise.

ZSP Zone Service Provider.

1

Introduction

Contents

Unmanned Vehicle (UxV) have caused a paradigm shift in contemporary military operations by offering advanced reconnaissance, surveillance and attack capabilities, thereby minimising the direct exposure of military personnel to hazardous situations [2]. The integration of these vehicles into Command, Control and Communications (C3) systems is imperative for facilitating coordination and an informed and agile decision-making process by the command through the transmission of real-time information from the battlefield [3].

The adoption of efficient communication protocols is therefore fundamental to the performance of these systems. By publishing Standardization Agreement (STANAG) 4586 in 2002, North Atlantic Treaty Organization (NATO) began a process of standardising communication protocols to achieve interoperability between different types of Unmanned Aerial Vehicle (UAV)s from different countries deployed in joint operations. Although it sets clear technical requirements, such as message formats and command protocols, it gives member nations the freedom to implement specific solutions that meet these standards.

This work was developed within the scope of the Portuguese Army's Project EXE02 – Remote and Autonomous Systems, specifically in EXE03 – Unmanned Ground Systems, through the eMOVE – Robotisation of the M113 initiative.

1.1 Motivation and Challenges

The development of these C3 systems mostly leads to the acquisition of commercial solutions available on the market due to their ease of acquisition and high cost/quality ratio. However, with these acquisitions come the challenges of integrating these commercial systems with the NATO standards already implemented in other C3 systems in use in one country or in other alliance countries systems.

Among the commercial solutions is the open-source Micro Air Vehicle Link (MAVLink) protocol, widely used in civilian applications due to its low cost and simplicity, which has been considered for military use in recent years. However, this protocol has significant security vulnerabilities, such as the lack of robust authentication and encryption mechanisms, which can be exploited by adversaries to compromise the integrity of UxV operations [4]. To mitigate these flaws, it is imperative to implement solutions that guarantee the Confidentiality, Integrity, and Authentication (CIA) of messages, without introducing significant latencies or even increasing energy consumption, critical factors for the autonomy of UxVs [5].

The success of a C3 system is also measured by the operational flexibility it can offer commanders on the battlefield. This requires the development of authorisation systems that empower commanders to delegate control of UxVs to the appropriate units and possibly withdraw control from control stations that have been compromised.

This work is developed within the scope of the Portuguese Army's Project EXE02 – Remote and Au-

tonomous Systems, specifically in EXE03 – Unmanned Ground Systems, through the eMOVE – Robotisation of the M113 initiative, which aims to modernize and automate the M113 armoured vehicles, increasing their operational effectiveness and adaptability to current challenges [2]. The implementation of safe and efficient C3 systems is therefore essential for the success of this project and for the evolution of the armed forces' operational capabilities.

1.2 Objectives and Key Results

The primary objective of this investigation was to design and implement a new lightweight Command, Control, Communications, Computers, and Intelligence (C4I) system, similar to the Battlefield Management System (BMS), that addresses this C4I system operational limitation of not permitting UxV control or control handover. The proposed system, while ensuring CIA, should enable a combat unit commander to securely and efficiently transfer UxV control between Ground Control Stations (GCSs) while maintaining real-time access to UxV telemetry. Furthermore, it should operate within the constraints of military environments, supporting rapid control delegation without compromising security.

As a result of this thesis, a new system named New Command and Control System (NC2S) was designed, implemented, and evaluated under different operational architectures, comparing its performance over Wireless Fidelity (Wi-Fi) and HR-5000H tactical radios. The NC2S framework demonstrated strong alignment with the CIA principles, ensuring secure communication and control delegation between nodes.

Experimental validation confirmed that NC2S achieves secure handover and key renewal with manageable latency while maintaining stable communication under limited bandwidth conditions. A direct comparison revealed that the HR-5000H radios introduce latency approximately two orders of magnitude higher than Wi-Fi, significantly influencing connection establishment, control handover, and key renewal times. In terms of system reliability, both communication media exhibited high stability, with similarly low message loss percentages. These results indicate that, due to their limited bandwidth and high latency, the HR-5000H tactical radios are currently more suitable for communication between command nodes, while Wi-Fi remains the preferred medium for UxV control, where higher message rates are required.

1.3 Thesis Outline

This thesis is organized into six chapters. Chapter 1 introduces the motivation, challenges, objectives, and scope of the work, highlighting the importance of secure and interoperable communication protocols for UxV control within military C3 systems. Chapter 2 provides the necessary background, covering UxV and C3 concepts, relevant communication technologies, the MAVLink protocol, and NATO

STANAG 4586, along with a discussion of common cyber threats. Chapter 3 reviews related work on cryptographic, authentication, and control handover mechanisms, identifying their limitations and motivating the proposed approach. Chapter 4 presents the design of the proposed NC2S framework, describing its architecture, node interactions, and the developed security protocols for key management, credential renewal, and control delegation. Chapter 5 discusses the experimental setup and testing results obtained under different communication conditions, analysing performance metrics such as latency, throughput, connection and handover times. Finally, Chapter 6 concludes the thesis by summarizing the main contributions, assessing system limitations, and proposing directions for future work.

2

Background

Contents

This chapter provides the theoretical basis necessary to understand the context of this work, presenting the fundamental concepts, technologies, and standards that support the design and implementation of secure C3 architectures for unmanned systems.

2.1 Unmanned Vehicle (UxV) and Unmanned System (UxS)

UxVs are systems that operate without a human on board, relying instead on remote control, supervised control or autonomous control [6]. They combine sensors, communication systems, and onboard processing to execute tasks in environments that are dangerous, costly, or inaccessible to human operators. According to their operational domain, they can be classified as ground vehicles (Unmanned Ground Vehicle (UGV)), aerial vehicles (UAV), surface vehicles (Unmanned Surface Vehicle (USV)), or underwater vehicles (Unmanned Underwater Vehicle (UUV)), each designed to extend human reach and effectiveness in land, air, maritime, and subaquatic contexts.

While the term UxV refers strictly to the vehicle platform itself, it is important to recognize that these vehicles are usually integrated within a broader Unmanned Systems (UxS). An UxS includes not only the vehicle itself but also the supporting infrastructure and interfaces required for its effective operation. According to Hossain *et al.* [7], a typical UxS is composed of five main elements, the unmanned vehicle, the ground control station, the communication links, the payload, and the human operators.

2.2 Military application of UxVs

UxVs are now an irreplaceable part of military operations, improving combat capabilities in various domains as demonstrated above. The main areas of use of UxVs in the armed forces include:

- **Surveillance, reconnaissance, and intelligence gathering**: UxVs, particularly UAVs, are widely used for intelligence gathering, providing real-time data on enemy positions and movements via video streaming [8];
- **Combat operations**: UxVs are used in offensive strike missions, including precision attacks and combat support, reducing the risk to human personnel and increasing combat potential [9];
- **Logistics**: UxVs facilitate the transport of supplies, war material, and medical aid to frontline units, ensuring timely and efficient delivery in combat environments;
- **Electronic warfare**: UxVs are used in electronic warfare actions to disrupt enemy communications;

To ensure interoperability, safety, and efficiency, military UxVs adhere to specific standards and regulations. For instance, the US Department of Defense (DoD) established the Joint Architecture for Unmanned Systems (JAUS) as an interoperability standard for UGVs, focusing on communications and data processing for Command and Control (C2). Similarly, the Intelligent Network for Teams of Exploratory Robots with Adaptive Collaborative Technologies (INTERACT) project, developed by the European Defense Agency (EDA) and funded by the Preparatory Action for Defence Research (PADR), aims to establish standardization and interoperability guidelines for UxVs from various member countries, aligning with JAUS's objectives [10, 11].

2.3 Radio Technologies for UxV Control

This section explores various radio technologies employed in UxV control, including widely used wireless communication systems like Wi-Fi and proprietary radio solutions tailored for specific platforms. These technologies differ in their communication range, data throughput, and application scenarios, each playing a vital role in enhancing UxV operations in diverse environments.

2.3.1 Wi-Fi

Wi-Fi is a widely utilized wireless communication technology based on IEEE 802.11 standards, operating in the 2.4 GHz and 5 GHz frequency bands. Wi-Fi is distinguished by its high data transfer rates and cost efficiency, rendering it particularly well-suited for short to medium-range wireless communication. Its versatility lies in its ability to handle diverse data types with minimal latency, a feature that has proven instrumental in applications such as the control of UxVs [12].

In the context of UxV control, Wi-Fi serves as a critical enabler for C2 communication, facilitating the real-time exchange of telemetry data and operational commands between UxVs and GCS. Its high bandwidth capacity is particularly beneficial for supporting the transmission of sensor data and live video feeds, which are essential for tasks requiring reconnaissance and enhanced situational awareness. However, its communication range is the greatest limitation to its application in the UxV control scenario.

2.3.2 UxV-Specific Radio Technologies

Proprietary radio technologies can be described as specialized systems developed by a specific manufacturer for unique applications. They are often tailored to a particular product or platform and usually include custom hardware, such as antennas, transceivers, or modems, and unique communication protocols designed to optimize performance, enhance security, and ensure reliability in telemetry and control for UxVs.

The SIYI HM30, Herelink, and Holybro Telemetry Radio are three distinct hardware solutions for UAV communication, each with unique capabilities. The SIYI HM30 offers the longest communication range, reaching up to 30 km in the 2.4 GHz band, supporting full HD video at 1080p/60 fps with low latency. The Herelink system, designed for ArduPilot and PX4, provides a 20 km range in the same frequency band, integrating telemetry and video with ultra-low latency. In contrast, the Holybro Telemetry Radio is a low-cost alternative focused on bi-directional telemetry, operating in the 915 MHz or 433 MHz Industrial, Scientific, and Medical (ISM) bands, with a range of up to 10 km when using high-gain antennas [13–15].

2.4 MAVLink Protocol

The MAVLink protocol is a lightweight open-source communication protocol that is now widely used by various UxVs, particularly UAVs, to facilitate data exchange between the components on board the UxV and GCS [16]. This protocol's versatility, which supports customizable messages and bidirectional communication, along with its cross-platform compatibility and well-documented structure, makes it highly suitable for implementation in the framework developed in this investigation.

MAVLink uses a variety of communication protocols, including serial communication (Universal Asynchronous Receiver-Transmitter (UART)/RS-232) for high reliability and low latency, User Datagram Protocol (UDP) for efficient, low-overhead transmission in latency-sensitive applications, though without guaranteed delivery, Transmission Control Protocol (TCP) for reliable data transfer through acknowledgment and retransmission mechanisms, and Bluetooth for seamless short-range wireless connectivity, enabling rapid integration with portable devices such as tablets and smartphones [1].

MAVLink has two versions: an older one, known as MAVLink 1.0, adopted around 2013, and its current version, widely implemented by many users since early 2017, called MAVLink 2.0. These two versions present some differences in the structure of their messages. However, in the control system that was developed, only version 2.0 was implemented. A MAVLink 2.0 message's fields are represented in Figure 2.1 and consist of:

- **STX** - Marks the start of a message, beginning with 0xFD;
- **LEN** - Specifies the length of the message payload, encoded in 1 byte;
- **Incompatibility Flags**: Indicate whether the message contains specific features requiring special handling, such as message signing;
- **Compatibility Flags**: Specify fields that can be ignored if not interpreted, such as flags indicating packet priority;
- **SEQ** - Marks the message sequence to detect lost packets;

- **SYS ID** - Identifies the source system;
- **COMP ID** - Identifies the component (e.g., Global Positioning System (GPS) sensor) of the UxV sending the message;
- **MSG ID** - Specifies the type of message in the payload, allowing the receiver to correctly interpret the PAYLOAD;
- **PAYLOAD** - Contains the data to be transmitted. This may include telemetry data, state updates, or control commands;
- **CHECKSUM** - Ensures the integrity of the message using a Cyclic Redundancy Check (CRC) to detect transmission errors;
- **SIGNATURE**: Consists of a 13-byte Hash-based Message Authentication Code (HMAC) that ensures message integrity and authentication. It is composed of three fields:
  - **LINKID**: A 1-byte field identifying the channel ID used to send the message;
  - **Timestamp**: Encoded in 6 bytes, used to prevent replay attacks;
  - **Signature**: Encoded in 6 bytes, derived from the first 6 bytes of the Secure Hash Algorithm 256-bit (SHA-256) hash applied to the complete message, along with the timestamp and a 32-byte symmetric key shared between the GCS and the UxV.

The signature field enforces security measures by discarding packets if the received timestamp is older than the last message's timestamp or if the calculated signature differs from the packet's signature.

Figure 2.1: MavLink 2.0 header and Payload [1].

MAVLink supports a wide range of message types used for communication between UxVs and GCS. These message types can be categorized into two groups based on their function: State Messages and Command Messages [1]. State messages are sent from the UxV to the GCS and contain information such as location, speed, or altitude. Among the state messages are the `HEARTBEAT`, `System Status`, and `Global Position` messages, which respectively indicate that the UxV is present and active, provide information about the system state (e.g., battery level, sensors, and communication status), and transmit the UxV's GPS coordinates.

On the other hand, Command Messages are sent from the GCS to the UxV and contain tasks for the UxV to execute. Examples include the `COMMAND_LONG` message, which is used to control the UxV,

and the `MISSION_ITEM_COMMAND` message, which specifies the coordinates of a waypoint for the UxV to autonomously navigate toward.

2.5 UxVs Cyber-Attacks and MAVLink Security Threats

To design and implement a secure control delegation system for UxVs, it is necessary to study and understand the most common UxSs security flaws. This section explores various cyber-attack vectors targeting UxVs, focusing on the security challenges faced by communication protocols such as MAVLink. These threats are categorized into four types: confidentiality, integrity, availability, and authenticity attacks. Each of these threat categories presents unique risks to the safety and effectiveness of UxV operations, and understanding them is essential for developing secure systems capable of resisting malicious attempts to compromise their functionality.

2.5.1 Confidentiality attacks

In this type of attacks, the attacker gains access to the data transmitted between two nodes (for example, UxV and GCS), compromising its confidentiality. These attacks mostly consist of **Eavesdropping** and **Traffic Analysis** attacks.

- **Eavesdropping** occurs due to the absence of encryption, allowing the attacker to intercept and store data for future attacks, such as replay or fabrication. This can be performed using packet analyzers like `Wireshark` to passively capture telemetry, video streams, or control data [17].
- **Traffic Analysis** refers to the collection and inspection of metadata from intercepted packets, such as the Message Authentication Code (MAC) address, encryption type, or communication protocol used [18].

2.5.2 Integrity attacks

Integrity attacks involve modifying the content of messages transmitted between two nodes, thereby compromising the system's reliability. These attacks include **Man-in-the-Middle (MITM)**, **Hijacking**, **Replay**, **False Location Updates**, **Script Injection**, **Shell Injection**, and **Remote Command Execution** attacks.

- In a **MITM** attack, the attacker is placed between two nodes, enabling interception and manipulation of data. This attack can be separated into passive and active forms. In passive attacks, the attacker may only observe traffic (e.g., traffic analysis), while active attacks involve altering commands or telemetry, often combined with spoofing techniques [19].

- **Hijacking** or unauthorized command injection exploits the lack of authentication, enabling the attacker to take control of the UxV, misleading the GCS into believing it still has control of the vehicle [17].
- **Replay** attacks involve the retransmission of previously captured valid messages to the UxV, which may result in unintended behavior or execution of commands [17].
- **False Location Updates** use tools like `Scapy` to send spoofed messages to the GCS, misleading it about the UxV's actual location and potentially causing incorrect operational decisions [17].
- **Script Injection** refers to inserting malicious code into the target Operating System (OS), with the goal of altering or disabling some system functionality [20].
- **Shell Injection** allows the attacker to insert commands directly into the system shell, enabling arbitrary code execution. Once successful, this attack grants low-level access to the operating system and can be used to modify the target flight path and/or the system logs [20].
- **Remote Command Execution (RCE)** enables attackers to execute unauthorized commands remotely, mostly by targeting open services or vulnerabilities in protocols such as Telnet or FTP [20].

2.5.3 Availability Attacks

Availability attacks aim to disrupt or disable the communication between two nodes, potentially placing the vehicle in a lost-link or fail-safe state. These attacks include **Jamming**, **Denial of Service (DoS)**, **Flooding** and **Deauthentication** attacks.

- **Jamming** disrupts wireless communication by interfering with radio frequency signals through powerful noise or signals on the same frequency, preventing legitimate messages from being received. This attack can effectively block telemetry and control signals, causing the UxV to become unresponsive, as shown in [21].
- **DoS** attacks flood the communication link with unauthorized or malformed packets, overwhelming the system and preventing legitimate commands from being processed.
- **Flooding** attacks are a subset of DoS attacks that involve overwhelming the system with excessive packets of various types, such as TCP, UDP, Internet Control Message Protocol (ICMP), or Hypertext Transfer Protocol (HTTP) requests, ultimately causing the UxV to become unresponsive and possibly crash, as demonstrated in [21, 22].
- **Deauthentication** attacks are another subset of DoS attacks that exploit the lack of protected management frames in some Wi-Fi-based systems by sending spoofed deauthentication packets,

effectively breaking the connection between the UxV and the controller. This is typically performed using tools such as `Aircrack-ng`. If the system does not implement IEEE 802.11w (management frame protection), this attack can be highly effective [21].

2.5.4 Authenticity Attacks

Authenticity attacks target the trustworthiness of the communication between two nodes, aiming to cause one or both parties to believe that the data they are receiving is genuine. These attacks include **Data Fabrication**, **Spoofing**, **Dictionary Password Attacks**, and **Brute Force Attacks**.

- In a **Data Fabrication** attack, the adversary, after acquiring knowledge of the protocol, injects false or malicious data into the communication stream between the two nodes [1].
- **Spoofing** attacks refer to the attacker impersonating a legitimate node. **GCS spoofing**, **Address Resolution Protocol (ARP) spoofing** and **MAC/Internet Protocol (IP) spoofing** are subsets of spoofing attacks. In **GCS spoofing**, the attacker impersonates a legitimate GCS and sends false wireless control commands to the UxV. **ARP spoofing** exploits the ARP protocol to associate the attacker's MAC address with the IP address of the UxV, redirecting communication to the attacker's device [21]. Finally, **MAC/IP spoofing** aims to impersonate legitimate devices on the communication network by using stolen identifiers, such as MAC or IP addresses. This allows the attacker to insert themselves into the communication channel and mislead the UxV or GCS into accepting commands or data from the attacker's device [21].
- **Dictionary Password** attacks involve attempting to crack passwords by systematically testing a list of common words, phrases, and combinations until the correct one is found. This type of attack is commonly used on wireless networks with weak or commonly used passwords or keys [20].
- **Brute Force** attacks differ from dictionary attacks in that they involve trying all possible combinations of characters to crack the password. These attacks require substantial computing power, especially when dealing with long or complex passwords [20].

2.6 Portuguese C4I Systems

C4I systems are designed to enhance decision-making and operational efficiency in military and other domains. C4I systems are regarded as systems of systems (SoS), comprising multiple discrete constituent systems that interoperate to meet specific mission requirements. Each system within the C4I framework is capable of operating independently, yet is also required to function in conjunction with other systems to achieve broader goals [23].

Due to their relevance for this work, this section focuses on the C4I systems used by lower-echelon units in the Portuguese Army.

2.6.1 BMS

The Portuguese Army has incorporated the BMS into its C4I capabilities. The principal objective of the BMS is to equip lower-echelon units with the tools necessary to achieve a Common Operational Picture (COP) and enable effective mission coordination.

The BMS is integrated into the C2 hierarchy at the battalion level and below, interacting with higher-level systems such as the High Management System (HMS) (responsible for Corps-to-Brigade command levels), the transitioning Sistema de Informação de Comando e Controlo do Exército (SICCE) toward NATO's Land Command and Control Information Service (LC2IS), and the Dismounted Soldier System (DSS), a developing system that targets platoons and individual soldiers.

The key capabilities of the BMS are:

- **Real-time information sharing**: The BMS allows for real-time sharing of information between units deployed in operations or military exercises;
- **Advanced tactical tools**: Includes mapping operations, terrain analysis, and the creation of tactical symbols and graphics;
- **Integration with Army Data Networks**: Through the Link-Tactical Information Distribution System (L-TIDS), the BMS connects radio communications with the Army Data Network (RDE), facilitating broader interoperability;
- **NATO compliance**: The BMS meets NATO requirements, ensuring interoperability with allied operational communications networks through STANAG 5527 (Friendly Force Tracking), STANAG 5525 (standardized data exchange), APP-6B (NATO Joint Military Symbology) and other relevant standards such as APP-11D and STANAG 4677;

For its communication backbone, the BMS primarily employs the PRC-525 radio, although it is designed to support other communication technologies.

When employing the PRC-525 the BMS faces a number of challenges, including the bandwidth limitations of the PRC-525, which result in the prioritization of voice over data and can disrupt information transmission, and the lack of PRC-525 radios in the majority of army units that constrains the comprehensive deployment of the BMS across some brigades [24].

However, the most significant limitation of the BMS, and the one that motivated this thesis, lies in its lack of integration of UxV command and control, particularly regarding the secure handover of control between operators. To solve this gap the NC2S was developed as part of a new C2 framework that,

similary to the BMS, provides commanders with access to the COP and enables the delegation and transfer of UxV control in a secure and authenticated manner.

2.6.2 DSS

The DSS (Dismounted Soldier System) is the C2 system adopted by the Portuguese Army for unmounted combatants. It is classified as a BMS extension designed for platoon and section commanders when they are outside the vehicle, thus allowing them to continue accessing the COP. The DSS is interoperable with the BMS and can be accessed through a data terminal running the Android operating system [25].

The DSS has been developed in accordance with NATO standards, specifically referencing STANAG 4677, which outlines norms and protocols for interoperability among allied forces [25]. The DSS is compared with other systems such as *Fortion Soldier C2* and Lightweight Command and Control Evolution (LC2EVO), highlighting its adherence to NATO standards and its ability to operate with various types of radios [26].

For its communication backbone, the DSS employs mostly the Rohde & Schwarz HR-5000H handheld tactical radio, ensuring encrypted and resilient connectivity for dismounted soldiers operating within the BMS network. However just like the BMS it is able to use other communication techenologies.

2.6.3 Military Radios

This section aims to present the military radios that the portuguese army mostly employs in both the BMS and DSS.

2.6.3.A PRC-525

The PRC-525 is a Combat Net Radio that equips most of the units of the Portuguese armed forces in its Fixed, Vehicular, or Manpack configurations and acts as the backbone of the BMS C4I system [27].

The radio can operate across a wide frequency range from 1.5 to 512 MHz, it supports High Frequency (HF), Very High Frequency (VHF), and Ultra High Frequency (UHF) communications with multiple channel spacings and modulation types, including Amplitude Modulation (AM), Frequency Modulation (FM), Single Sideband (SSB), and digital modes. It also integrates advanced frequency hopping techniques such as Secure Communications Hopping (SECOM H), Secure Communications Mode V (SECOM-V), and HAVE QUICK II, along with data transmission standards including STANAG 4285, 4539, 4529, and others, ensuring robust, interoperable, and secure communications [27].

The PRC-525 can also operate with output power levels up to 20 W in HF and 10 W in VHF or UHF and with receiver sensitivity around –115 dBm, providing reliable long-range voice and data links.

Designed for modern military operations, it incorporates GPS integration, Automatic Link Establishment (ALE) (Automatic Link Establishment), and is designed accordingly to Military Standard for Environmental Engineering Considerations (MIL-STD-810E) and Military Standard for Electromagnetic Interference Characteristics (MIL-STD-461E) standards for temperature, vibration, shock, immersion, and electromagnetic compatibility [27]. Finally, its autonomy is only limited in its Manpack configuration, where powered by its Li-ion batteries it can operate for exceeding 20 hours. It is also capable of integrating Local Area Network (LAN) and Wide Area Network (WAN) digital networks, crucial for the deployment of UxVs.

Beyond its security features and range, it has a limited bandwidth of just 72 kb/s which can compromise the amount of data that can be transmitted between UxVs, and its size and weight are also a fracture feature that mostly impedes its application in UAVs. Due to these drawbacks, the PRC-525 is only an option for C3 between commanders and GCSs and not for the UxV control.

2.6.3.B HR-5000H

The R&S HR-5000H is the handheld version of the R&S HR-5000, a versatile tactical radio that is starting to be implemented in the Portuguese Army to meet the demanding needs of dismounted soldiers in field conditions under the Dismounted Soldier Communications System digitization program [28]. It operates in the frequency range of 30 MHz to 512 MHz without any frequency gaps and offers robust communication capabilities for both voice and IP-based data transmission. The radio provides two independent voice channels and integrates Ethernet and serial interfaces (RS-232) for external devices, making it suitable for portable, vehicular, and fixed configurations.

In the Portuguese Army, the HR-5000H operates with the SOVERON TNW50 narrowband waveform, which provides a Time Division Multiple Access (TDMA)-based Mobile Ad hoc Network (MANET), thus supporting dynamic slot assignment, relay operation, and integrated routing. The TNW50 offers two configurable modes. A Balanced Mode, optimized for higher throughput, up to approximately 22 kb/s, by employing lighter Forward Error Correction (FEC), and Robust Mode, optimized for reliability, up to approximately 10 kb/s, through stronger error correction and increased redundancy. Both modes use Continuous Phase Modulation (CPM) within a 50 kHz channel and can achieve communication ranges of up to 50 km under line-of-sight conditions. In terms of security, the TNW50 integrates Advanced Encryption Standard (AES)-256 encryption (Communication Security (COMSEC)) for data protection, uses anti-jam frequency hopping (Transmission Security (TRANSEC)) to ensure signal integrity in hostile environments and Network Security (NETSEC), which provides end-to-end encryption and authentication across the entire MANET.

Just like the PRC-525, the HR5000 is designed to endure harsh conditions, so it complies with newer versions of the MIL-STD-810E and MIL-STD-461E standards. Its compact form factor (less than 1.2 kg)

and long battery autonomy (over 12 hours) make it ideal for operations in demanding terrain, including urban and hilly environments [28]. Compared to the PRC-525, the HR-5000H main advantage lies in its reduced weight and size, making it more suitable for communication between command nodes that are distant from the command center.

2.7 UxV Simulation

There are two key concepts to consider in UxV simulation: Hardware In The Loop (HITL) and Software In The Loop (SITL). The main goal of SITL is to simulate both the environment and the vehicle entirely on a development machine, typically a PC, without physical hardware. It is commonly used in early development stages to validate software logic before hardware integration. It communicates with GCS software through protocols like MAVLink to simulate mission execution. The key benefits of SITL are its cost-effectiveness, quick setup, and the safety it provides when experimenting with new code [17].

Conversely, HITL simulation integrates real hardware components with simulated environments to test and validate control systems, allowing developers to assess how hardware interacts with software in real time and ensuring that the system behaves as expected under various conditions [17].

2.8 Summary

This chapter introduced the fundamental concepts and technologies that support this thesis. It started by defining UxVs and UxSs, outlining their core components and military applications in surveillance, combat, logistics, and electronic warfare. Wi-Fi and proprietary radios were presented as part of commercial solutions for UxV control. The chapter also reviewed communication standards, focusing on the MAVLink protocol, its message structure, communication modes, and security features.

In addition, common cyberattack vectors against UxVs were examined and grouped into confidentiality, integrity, availability, and authenticity threats, emphasizing the vulnerabilities of protocols such as MAVLink and the importance of robust countermeasures. The chapter also outlined Portuguese C4I systems, focusing on the BMS and DSS that support lower-echelon units and their main communication backbones (the PRC-525 and HR-5000H). Finally, simulation environments (SITL and HITL) were described as essential tools for testing and validating control systems. Together, these elements establish the operational, technological, and security foundations upon which the NC2S system is designed and evaluated in the following chapters.

3

Related Work

Contents

This chapter reviews the studies that most influenced the design of the proposed NC2S framework, focusing on approaches that enhance the security of UxVs in terms of packet protection (Section 3.1), key establishment and management (Section 3.2), and authentication with control handover (Section 3.3). The selected works were chosen for their direct relevance to improving confidentiality, integrity, and authentication (CIA) in tactical communication environments, providing the foundation for the design decisions adopted in this thesis.

3.1 Packet Protection and Cryptographic Protocols

To ensure confidentiality in a communication system, it is crucial to encrypt the message payload so that an attacker cannot access the transmitted information. However, along with the benefits of encryption come drawbacks, namely the increased time and computational resources required for encryption operations. Therefore, it is important to study and compare the wide range of cryptographic protocols available.

In [29], the authors sought to secure MAVLink packets by integrating symmetric encryption. AES-Counter Mode (CTR), AES-Cipher Block Chaining (CBC), Rivest Cipher 4 (RC4), and ChaCha20 were tested in an ArduPilot SITL environment and the results showed that AES-CBC had the highest processing cost and reduced packet transmission efficiency due to its sequential nature, while RC4 consumed the most memory and Central Processing Unit (CPU). AES-CTR performed better but still required more resources than ChaCha20. ChaCha20 achieved the best balance, with very low CPU and memory overhead, a packet rate close to the unsecured MAVLink, and suitability for real-time use in resource-constrained UAVs.

Tüfekci *et al.* [30] proposed enhancing MAVLink security by using Authenticated Encryption with Associated Data (AEAD), which encrypts the payload while also authenticating the packet header. They evaluated four AEAD schemes, ChaCha20-Poly1305, AES-Galois/Counter Mode (GCM)-Synthetic Initialization Vector (SIV), AES-Offset Codebook Mode (OCB)3, and AES-Counter with CBC-MAC (CCM), on a real drone incorporating a Raspberry Pi 4 and a Pixhawk flight controller. The results showed that ChaCha20-Poly1305 achieved the fastest execution time, AES-OCB3 was the least CPU-intensive, and AES-CCM required the least memory.

Table 3.1 summarizes the previously discussed cryptographic solutions. The analysis indicates that increased communication security leads to greater computational overhead, highlighting the need to balance security and performance in resource-constrained environments such as UxVs.

Table 3.1: Summary of packet protection and cryptographic protocols.

Reference	Method	Pros	Cons
Allouch *et al.* [29]	Symmetric encryption of MAVLink packets using AES-CTR, AES-CBC, RC4, ChaCha20	ChaCha20 achieved best balance with low CPU/memory usage and high packet rate; improved security over plain MAVLink	AES-CBC had high processing cost and reduced throughput; RC4 consumed most CPU/memory
Tüfekci *et al.* [30]	AEAD encryption (ChaCha20-Poly1305, AES-GCM-SIV, AES-OCB3, AES-CCM) securing payload and authenticating header	ChaCha20-Poly1305 fastest; AES-OCB3 least CPU usage; AES-CCM lowest memory usage	Increased computational overhead compared to unprotected MAVLink; trade-offs between speed, CPU, and memory per scheme

3.2 Key Establishment and Management

In a secure communication system where cryptographic algorithms are applied, it is essential to define how the keys used by these algorithms are established and managed. There are several approaches to this, from individualized keys to group keys. The following examples describe some of the solutions proposed in the literature.

Li and Pu [31] presented a lightweight key generation and authentication scheme where both the GCS and the drone derive identical keys using a chaotic Duffing map, eliminating the need for key exchange over the channel. The derived keys are used to sign and verify command messages, preventing MITM attacks. Experimental results showed significantly lower computation time, energy consumption, and memory usage compared to AES, Data Encryption Standard (DES), and 3DES, demonstrating the efficiency of the proposed approach.

In [32], the authors addressed the problem of weak or static key management in MAVLink communications by introducing a lightweight session key generation scheme based on a one-dimensional Chebyshev chaotic map. In this method, random keys are computed with high unpredictability while maintaining a low computational cost, making it suitable for UAVs with limited resources. The scheme was integrated with the High Security and Lightweight (HIGHT) block cipher to encrypt drone communication. Experimental results showed that the proposed approach generates keys very quickly, with minimal CPU and memory usage, while maintaining a high level of key randomness.

Another solution to the session key establishment problem was presented by Hashmi and Munir [33]. In their solution, the UAVs use the Elliptic Curve Diffie-Hellman (ECDH) algorithm to establish a shared session key between the drone and the GCS. Once the session key is set, the ChaCha20 algorithm is applied to encrypt the data. The proposed protocol was tested in a SITL environment, showing that it could generate secure session keys quickly with minimal CPU and memory usage.

To reduce the risk of physical capture, [34] proposed a group key management system where instead

of sharing the same group key for all drones, it assigns different session keys based on the hop distance from the GCS, with only drones at the same hop sharing a key. The keys are generated using a hash chain, which allows neighboring drones to derive each other's keys without storing large key sets. When the network topology changes, the GCS creates a new master key and securely distributes it hop-by-hop, with each node computing and forwarding the key to the next level. Experimental results showed that the authors solution achieved the lowest storage, computation, and communication overhead compared with other group key management schemes, while maintaining resistance to capture and supporting mobility.

In [35], the authors introduced a communication system where the UAV-Ground Station Server (GSS) session keys are computed via ECDH. These keys are short-lived and refreshed during handover operations. Also, to prevent physical attacks, each node's sensitive data, like its public and private keys, certificates, and Registration Authority (RA) shared secret, is linked to each user via a biometric fuzzy extractor and protected using a Physically Unclonable Function (PUF), where only the challenge values are stored in memory. In terms of performance, experimental results showed that the proposed protocol achieved lower computation time (around 0.784 ms) and communication overhead (only 1856 bits) than other schemes, while still providing stronger security guarantees such as resistance to physical capture, anonymity, and perfect forward secrecy.

Kwon *et al.* [36] proposed a system that utilizes Elliptic Curve Cryptography (ECC)-based key establishment mechanisms. In this system, prior to deployment, the GSS selects elliptic curve parameters, computes a master key, and, for each Zone Service Provider (ZSP), generates its identity, a handover shared secret with neighboring ZSPs, and the parameters required to derive its public and private keys. Then, during UAV registration, the GSS computes the UAV identification and public and private keys. The private keys of both ZSPs and UAVs are computed via an ECC-based process that binds elliptic curve parameters, identification parameters, and the GSS master key. During ZSP-UAV communication establishment, the session key is negotiated through an ECDH-based process using their private secrets. Experimental testing demonstrated that the system required only 2880 bits of communication and achieved lower computation time while maintaining strong security features.

Lin *et al.* [37] presented a solution for UAV communication based on four key types. In the GCS-UAV link a Master Key and a Refresh Key are negotiated via ECDH. The Session Key and Authentication Key are then derived from a hash combination of the Master Key and Refresh Key. The Session Key is used exclusively for encrypting messages, while the Authentication Key is used for computing HMACs to ensure authentication and integrity. In the key renewal protocol, the new keys are derived from the initial Master Key bounded to the Refresh Key. Forward secrecy is guaranteed by the Refresh Key, which is recomputed for each refreshment by hashing the previous one. Testing showed that the authors solution reduced communication overhead, improved throughput and scalability in UAV networks, and

still provided strong security guarantees, with only a slight trade-off in end-to-end delay compared to the established secure baseline.

Table 3.2: Summary of key establishment and management protocols

Ref.	Method	Pros	Cons
Ismael et al. [32]	Session key generation using Chebyshev chaotic map + HIGHT cipher	High key unpredictability; low computational cost; minimal CPU/memory usage	Limited validation on large-scale UAV networks; depends on chaotic map robustness
Hashmi and Munir [33]	ECDH for session key + ChaCha20 encryption	Strong protection against eavesdropping; lightweight; efficient in SITL tests	Secure ECDH parameter exchange required; no explicit resilience to physical capture
Bae et al. [34]	Saveless-based group key management with hop-distance keys via hash chain	Very low storage, computation, and communication overhead; supports mobility; capture resistant	More complex group rekeying; sensitive to frequent topology changes
Wen et al. [35]	ECDH with biometric fuzzy extractor + PUF for GSS private key	Resistance to physical capture; anonymity; forward secrecy; very low overhead (0.784 ms, 1856 bits)	Increased system complexity; depends on RA provisioning and biometric reliability
Kwon et al. [36]	ECC-based session keys with elliptic curve parameters, ID numbers and master keys	Low overhead (2880 bits); fast computation; strong security	Requires pre-deployment by GSS; scalability in dynamic topologies not fully tested
Lin et al. [37]	4-key scheme: Master+Refresh (ECDH), Session+Auth (hash-based)	Reduced overhead (2944 bits); improved throughput/scalability; forward secrecy via Refresh Key	Slight increase in end-to-end delay; requires frequent key updates
Khalid et al. [38]	Pre-shared secrets and nonces with AES–RSA hybrid for key establishment/encryption	Very low comm. cost (1280 bits); low computation (8.34 ms)	RSA adds computation in constrained UAVs; Pre-shared secret dependency remains
Semal et al. [39]	Certificateless PKC group key agreement with bilinear pairings	Group key in 2 rounds; scalable; strong AKE security	Higher computation; weaker mutual authentication and key control resistance
Li and Pu [31]	Chaotic Duffing map with pre-shared parameters, using byte substitution, matrix transformation, and random shuffling	No key exchange needed during operation; very low CPU, memory, and energy usage; outperforms AES/DES/3DES	Requires secure pre-configuration of Duffing map parameters; limited to authentication (no encryption)

The scheme proposed by Khalid *et al.* [38] is a hybrid AES-Rivest–Shamir–Adleman (RSA) system where the nodes authenticate via digital certificates and the session keys are derived from pre-shared secrets and exchanged nonces. The protocol demonstrated that AES provides lower computational and communication costs, with total end-to-end computation for the full protocol (certificate request + mutual authentication + key derivation) taking only around 8.34 ms and reducing the communication cost to

1280 bits, making it highly suitable for resource-constrained UAVs.

Finally, Semal *et al.* [39] presented a Certificateless Group Key Agreement with Constant Rounds (CLPKC) protocol designed to overcome the limitations of traditional Public Key Infrastructure (PKI) infrastructures. In this solution, the group key is established through a two-round broadcast process where each node shares ephemeral values, computes pairwise secrets with all others using certificateless keys and bilinear pairings, exchanges masked key contributions, and finally derives the same session key by hashing all members' contributions. When compared to other schemes, the authors' approach required slightly more computation but achieved group keys in only two rounds without signatures. It provides strong authenticated key exchange security, though with slightly weaker mutual authentication and key control resistance. Overall, it offers stronger guarantees and better scalability than existing certificateless group key agreement solutions.

The reviewed works present diverse approaches to key establishment and management, ranging from lightweight chaotic maps and ECDH-based schemes to PKI and certificateless protocols. Each solution balances performance, scalability, and security guarantees differently. Table 3.2 summarizes these methods, highlighting their main advantages and limitations.

3.3 Authentication and Handover Mechanisms

Authentication is fundamental to enabling secure control delegation between GCSs in UxV networks, ensuring that each command originates from an authorized and trusted node. This subsection reviews mechanisms that achieve mutual authentication and reliable handover, covering approaches based on public key infrastructures, ticket-based or hop-by-hop exchanges, lightweight trust anchors, blockchain frameworks, and pre-shared key models.

Several works rely on PKI and digital certificates to authenticate nodes and secure delegation.

A secure PKI-based communication framework is proposed in [40] for UAV networks with military applications, introducing two protocols and a handover mechanism. In the Drone-to-GCS protocol, authentication is achieved via the exchange and verification of digital certificates using Elliptic Curve Digital Signature Algorithm (ECDSA). The session key is negotiated via ECDH and, finally, key possession is confirmed with an HMAC exchange. In the Drone-to-Monitoring-Drone protocol, authentication is similar. However, to address constrained computational resources on a UAV, the authors optimize the handshake by attaching the HMAC earlier, saving one message. In the handover protocol, when the drone enters the GCS zone, it authenticates with the GCS through the Drone-to-GCS protocol and then deletes the Monitoring-Drone session. In experimental results, the authors showed that the Drone-to-GCS authentication protocol took around 213 ms, while the Drone-to-Drone authentication took around 29 ms, with formal verification (BAN-logic, Scyther) confirming strong guarantees. Although crypto-

graphically sound, the 213 ms latency could be problematic for real-time tactical scenarios if frequent handovers occur.

In [41], a global UAV authentication system based on PKI is also described. Authentication is based on digital certificates obtained via a manufacturing control certificate. These certificates are combined with UAV license certificates that cryptographically link the aircraft to its operator. Mutual authentication between entities is achieved using mutual Transport Layer Security (mTLS). Revocation is granted by Certificate Revocation Lists (CRLs) managed by the flight-control server. In the proof of concept, the system successfully performed certificate provisioning, mutual authentication, and secure communication, demonstrating feasibility, but also highlighted the need for redundancy to avoid single points of failure. From these solutions, it can be concluded that PKI-based approaches offer strong cryptographic guarantees but may introduce latency [40] or centralization risks [41].

Another group of works focuses on ticket-based authentication or hop-by-hop key renewal, inspired by Kerberos or lightweight key chains.

Ayati *et al.* [42] presents a Kerberos-based system that enables secure distributed control across multiple GCSs users and UAVs through timestamps, ticket lifetimes, and Blowfish encryption. The architecture relies on a Authentication Server (ATS) that issues Ticket-Granting Tickets (TGTs), which the Ticket-Granting Server (TGS) verifies for authenticity and freshness before generating service tickets for each UAV. These tickets allow multiple GCSs to authenticate and control UAVs without requiring modifications to them. The system also supports role-based access, where commanders can delegate control through ticket assignments, ensuring scalability and specific access management. The scheme was validated in simulation and showed lower authentication overhead and better scalability than other solutions. However, no real UAV testbed validation was provided. Another possible point of failure is the requirement for constant connectivity to the ATS/TGS, which is not always feasible in tactical UAV networks.

Bae *et al.* [34], also cited in Section 3.2, present a hop-by-hop node-withdrawal and re-participation protocol together with a drone delegation system. The GCS-UAV authentication is performed in a hash chain of keys that derive from a start point key passed from the GCS to the UAV pre-flight in a secure location. The GCSs, on the other side, authenticate each other via digital certificates. In the re-participation protocol, the UAV proves it is not compromised by sending encrypted flight data and timestamps collected during the absence, which the GCS verifies. For control delegation, the two GCSs are already mutually authenticated, and the in-control GCS sends a session key to the new GCS. Then sends a handover message to the UAVs with the new GCS ID and the session key to use. The UAVs authenticates with the new GCS via the shared key. Although very lightweight, the rejoining mechanism assumes the UAV's storage cannot be compromised during absence.

Wang *et al.* [43] propose a handover key-management system for Long-Term Evolution (LTE)-based

UAV networks supporting three scenarios: X2 handover (neighboring Ground Relay Stations (GRSs) exchange root-key data and derive new Access Stratum (AS) keys), S1 handover (non-connected GRSs use Mobility Management Entity (MME) mediation for key distribution), and inter-MME handover (switches to pre-distributed root-keys with counter reset). The system requires only 2880 bits communication, achieves faster computation than previous LTE solutions, and maintains key separation against compromised GRSs. However, its LTE-specific design limits applicability to *ad hoc* tactical networks.

These ticket/key-chaining systems reduce computational overhead compared to PKI-based systems, but still trade off with Certificate Authority (CA) reliance [42], deployment constraints [43], or assumptions about uncompromised UAV storage [34].

Some solutions reduce complexity by using lightweight trust anchors instead of full PKI.

Wen *et al.* [35], referenced in Section 3.2, also introduce a handover authentication mechanism. In this protocol, the RA serves as a trust anchor between the GSS and the UAV. When a UAV enters the coverage area of a new GSS, it sends its temporary ID and the temporary ID hashed with the RA shared secret. The new GSS authenticates the UAV by recomputing the hash of the UAV ID with the RA shared secret. Once the GSS authenticates the UAV, it creates a new temporary ID linked to the RA shared secret, computes the ECDH public parameters, and sends both to the UAV. The UAV authenticates the GSS by verifying the new temporary ID. Experimental results show that the handover protocol achieves mutual authentication with only two lightweight messages (1856 bits), significantly outperforming other schemes. However, the system assumes secure RA provisioning and trust in RA secrecy.

A related approach in Kwon *et al.* [36] (also referenced in Section 3.2) proposes a handover protocol for a UAV to pass from one ZSP to another. When a UAV reaches a new ZSP zone, it sends a handover request containing its ID hashed with the session key to the new ZSP. Because the new ZSP cannot yet authenticate the UAV, it retransmits the message to the in-control ZSP using the neighboring-ZSP key. With this, the in-control ZSP serves as a trust anchor. Then the in-control ZSP decrypts and authenticates the UAV and responds positively to the new ZSP. The UAV and the new ZSP then negotiate a new key via an ECDH-based mechanism.The revocation mechanism is granted by CRLs periodic broadcast. Performance evaluation showed total computation costs of 11.001 ms across all entities, significantly lower than existing approaches, while supporting role-based access control through the trusted ZSP infrastructure.

Choe *et al.* [44] propose an ECC-based authentication solution using a CA as trust anchor. The CA generates key pairs, group keys, and trust-anchor secrets from soldier IDs. For Soldier–UAV authentication, both parties compute ephemeral ECC keys and derive session keys using peer ephemeral parameters and trust-anchor secrets. Mutual authentication involves HMAC verification and secured message exchange. UAV–UAV communication uses pre-shared group keys, while CA–UAV authentication adds Ultra Wide Band (UWB) timestamps for location verification. The protocol offers high security

against MITM attacks with low overhead, but the centralized CA presents a single point of failure.

These lightweight trust-anchor systems show better latency and efficiency, but their security is only as strong as the trust anchor and its availability.

Blockchain technology has recently emerged in secure UxV communications due to its decentralized trust. Li et al. and Son et al. [45, 46] demonstrate blockchain-based frameworks that provide decentralized authentication and handover mechanisms for UAV networks and vehicular networks respectively. These approaches leverage distributed ledger technology to eliminate single points of failure and provide tamper-resistant authentication records. However, blockchain solutions typically introduce additional computational overhead and latency compared to traditional cryptographic approaches, making them more suitable for scenarios where decentralization and resilience are prioritized over real-time performance requirements.

In a different approach, Filho *et al.* [47] propose a control handover of UAV between two GCSs to be performed via a preflight list of allowed GCSs and predefined keys for each allowed GCS. In this system, when a new GCS attempts to take control of a UAV, it sends a handover message encrypted with the key negotiated preflight. Upon receiving the message, the UAV authenticates the GCS by decrypting the message with the key linked to that GCS and verifying that the GCS ID is in the allowed list. If so, it then hands over control to the new GCS and disconnects from the previous one. Tests in a SITL setup showed that unauthorized GCSs were blocked, delays were detected, and network overhead remained minimal.

Fern *et al.* [48] studied a Multi-Operator Multi-UAV (MOMU) control concept where two GCSs can share and reassign UAVs in real time through manual operator approval processes. Each operator views all UAV sensor data, but only the current 'owner' can control flight and payload. Tests showed that deliberate UAV control handovers improved performance on high-priority tasks, though efficiency depended on clear communication and teamwork between operators.

As summarized in Table 3.3, authentication/hand-over designs include PKI-centric schemes (strong guarantees but latency/centralization risks), ticket/hop-by-hop models (lightweight yet reliant on continuous authority connectivity or secure pre-flight setup), lightweight trust-anchor variants (efficient but hinge on anchor secrecy/availability), blockchain (decentralized but costly in delay/energy/storage), and preshared or human-in-the-loop approaches (simple yet inflexible at scale). No single option meets all tactical needs, motivating NC2S blended design.

3.4 Summary

In summary, the reviewed literature shows a wide spectrum of approaches for securing UxV communications. At the protocol level, packet protection schemes provide confidentiality and integrity, though at

the cost of computational overhead. Key establishment and management methods ensure secure session keys but often depend on central authorities or preflight configurations. Finally, authentication and handover mechanisms directly address the challenge of control delegation, balancing strong guarantees against latency, reliance on trust anchors, or heavy infrastructures such as blockchain.

Taken together, these works show that while many effective mechanisms exist, none fully satisfy the requirements of tactical UxV operations, where secure, low latency, and flexible control delegation is essential. This gap motivates the development of the system proposed in this thesis.

Table 3.3: Comparison of authentication and control handover solutions.

Reference	**Method**	**Delegation / Handover**	**Strengths**	**Weaknesses**
Ko *et al.* [40]	PKI + ECDSA + ECDH + HMAC	Drone–GCS, Drone–Monitoring Drone	Strong security, formal verification, lighter Drone–Drone protocol	213 ms latency, scalability concerns
Mahmoud *et al.* [41]	PKI + mTLS	Global UAV authentication	Regulatory integration, secure provisioning	Central MCC+FCS = single point of failure
Filho *et al.* [47]	Pre-shared keys + allowed list	UAV–GCS	Efficient, blocks unauthorized GCS	Inflexible (no dynamic GCS addition)
Ayati *et al.* [42]	Kerberos (AS/TGS + Blowfish)	Multi-GCS delegation	Scalable, role-based delegation	Requires continuous AS/TGS connectivity, no UAV testbed validation
Bae *et al.* [34]	Hop key chaining	GCS–UAV + GCS–GCS	Lightweight, rejoin mechanism	Needs secure preflight setup, assumes UAV storage integrity
Wang *et al.* [43]	Root key + counters (AS keys)	LTE GRS handovers	Low cost, ensures key freshness	LTE-specific, less generalizable
Wen *et al.* [35]	RA trust anchor + ECDH	GSS handover	Very efficient (1856 bits), lightweight	Assumes RA secrecy and availability
Kwon *et al.* [36]	ZSP relay + ECDH	ZSP handover	Low cost (11 ms), role-based access control	Requires ZSP infrastructure
Rajasoundaran *et al.* [45]	Blockchain + AES + DAG routing	Swarm UAV handovers	Decentralized trust, resilience, 8–14% better performance	Consensus overhead, energy consumption, scalability issues
Song *et al.* [46]	Blockchain + ECC + hash/XOR	VANET handovers	Very efficient handovers, low computation	Not validated in UAV context, blockchain storage burden
Fern *et al.* [48]	MOMU (human-in-loop)	Operator–operator delegation	Ensures intentional handovers, safe against mistakes	Manual approval delays, teamwork required
Choe *et al.* [44]	ECC + CA trust anchor + UWB	Soldier–UAV, UAV–UAV	Lightweight, resilient, adds location verification	Centralized CA is single trust anchor

4

Solution for Secure and Flexible C3 integrating commercial UxVs - NC2S

Contents

This chapter presents the design, implementation, and testing prototype of the proposed NC2S framework, detailing its architecture, components, and security protocols that together enable a secure and flexible C3 system for the integration of commercial UxVs in military operations. All the NC2S proxy scripts, protocols fluxograms and Graphical User Interfaces (GUIs) can be consulted in [49].

4.1 System Architecture, Trust Model and Design Requirements

This thesis presents a solution for a secure and lightweight communication framework named the NC2S. The system architecture is composed of the following entities:

- **Central Authority (CA)** – Acts as the trust anchor for the initial authentication of all nodes. It is responsible for generating ECDSA certificates and, in the event of node compromise, issuing the corresponding certificate revocation lists (Certificate Revocation List (CertRL)). The CA communicates directly with Tactical Commander (TC)1.
- **Tactical Commander Level 1(TC1)** – Represents the highest-ranking officer of the mission. This entity defines the system policy, requests all certificates and key pairs from the CA, creates and signs credentials, and manages the control handover process. These functions are performed through the Commander Terminal Level 1 graphical user interface (Commander Terminal (CT)1).
- **Commander Terminal Level 1 (CT1)** – A terminal equipped with a graphical user interface that allows the TC1 to manage the entire system. It handles message routing, credential and certificate management, and cryptographic operations. The CT1 can establish connections with multiple CT2 units and several GCS nodes.
- **Tactical Commander Level 2 (TC2)** – A subordinate officer in the hierarchy beneath the TC1. This entity has access to information from multiple GCS and consequently several UxVs. However, the level of access is defined by the TC1 through credential and certificate policies. The TC2 interacts with the system via the CT2 interface.
- **Commander Terminal Level 2 (CT2)** – A terminal similar to the CT1 but with restricted privileges. It cannot request certificates, create credentials, revoke nodes, or initiate control handovers. It maintains communication with the CT1 and multiple GCS nodes.
- **Ground Control Station (GCS)** – The node that connects directly to the UxVs and can assume control of them depending on the permissions defined by the TC1. Vehicle control is executed via the Mission Planner software, which interfaces with a Python script responsible for cryptographic operations, message verification, and information routing. Each GCS can connect to multiple UxVs, CT2 units, and the CT1.

- **Unmanned Vehicles (UxVs)** – The unmanned platforms responsible for executing mission objectives. Each UxV maintains a direct communication link with one or more GCS nodes.

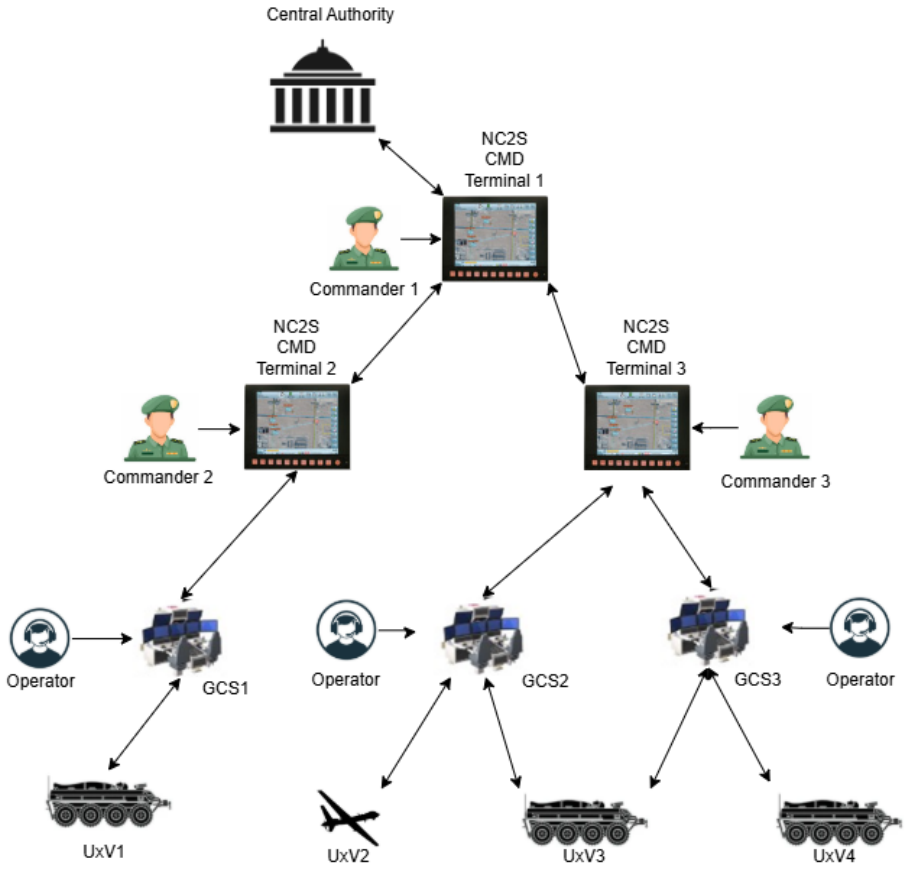

Figure 4.1: NC2S Architecture.

Both the GCS and the UxV comprise two components. A Python-based proxy script is responsible for receiving, verifying, and processing all NC2S messages and forwarding them to the correct nodes, thus integrating routing capabilities. In the case of UxV control, this process is managed on the GCS side by the Mission Planner and simulated on the UxV through the Ardupilot SITL. The Python proxy connects to these applications, unpacking incoming raw MAVLink 2 messages and forwarding them accordingly, while also packing outbound messages from the Mission Planner or SITL before sending them to the respective destination node.

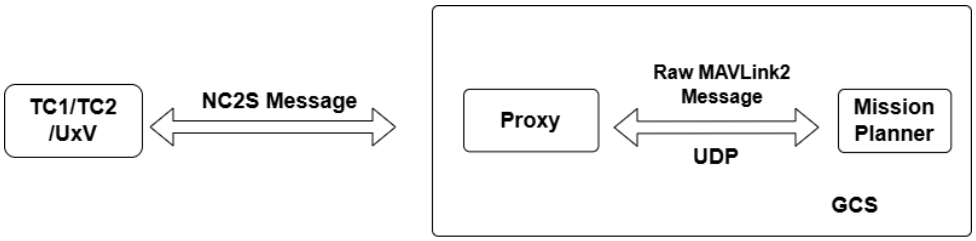

Figure 4.2: GCS Architecture.

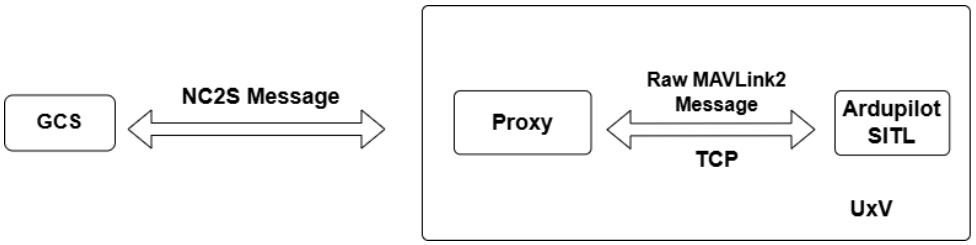

Figure 4.3: UxV Architecture.

4.2 Trust Model and System Requirements

This section formalizes the trust model driving the NC2S, stating its security goals, underlying assumptions, adversary capabilities, and the operational/performance requirements used to guide the design and evaluation.

A – Security Goals (G) The goals below define what NC2S must achieve—independently of specific mechanisms—to be deemed secure in our operational context.

- **G1** *Mutual authentication at join:* only authorized entities may join and participate and each pair forming a connection must authenticate mutually during establishment.
- **G2** *Per-message origin authentication:* every message must be verifiably bound to a legitimate sender participating in the current session.
- **G3** *Credential authenticity and binding:* mission credentials are authentic, unaltered, and cryptographically bound to the entity that presents them.
- **G4** *Authorization before action/forwarding:* acceptance, forwarding, and control delegation are governed by a capacity string present in the credentials consistent with the mission policy document.
- **G5** *Integrity and replay resistance:* messages cannot be modified undetected and replays are rejected within a bounded window.
- **G6** *Revocation responsiveness:* credential/certificate revocations propagate promptly and take effect in active sessions.

- **G7** *Key freshness and separation:* session keys are fresh, context-bound, and separated by direction/purpose. Also keys used for distinct roles are not reused.
- **G8** *Operational confidentiality (policy-driven):* when mission context requires confidentiality, transport links used for control/telemetry are encrypted at an appropriate layer.

B – Assumptions (A) This section states the environmental and organizational trust anchors on which the NC2S guarantees rely.

- **A1 Root of Trust (the CA):**The entire system's security is anchored to a single, offline **CA**. The CA is assumed to be secure, uncompromised, and to perform its functions honestly. While the CA is responsible for creating certificates and the CertRL, it is the TC1 that requests, distributes, and manages these assets.
- **A2 Policy and Authorization Anchor (the TC1):** The **TC1** is the root of *operational* trust and is assumed to be physically secure. While the CA validates a node's *identity*, the TC1 validates its *authority*. The system trusts the TC1 to define and enforce mission policies, issue and manage access credentials, maintain the validity of all security artifacts (certificates, CertRLs, and Credential Revocation Lists (CredRLs)), and ensure the timely distribution of updated CredRLs to regulate operational permissions dynamically.
- **A3** *Time base:* nodes keep bounded clock skew adequate for timestamp-based replay checks (operationally monitored).
- **A4** *Transport-layer confidentiality:* The confidentiality is assumed to be provided by the transport layer, not by the NC2S application layer. For tactical-radio links (e.g., HR-5000H), COMSEC/TRANSEC provide confidentiality and contribute to availability. For Wi-Fi/5G/IP links, an encrypted transport technology (e.g., Wi-Fi Protected Access 2 – Enterprise (WPA2-Enterprise), Virtual Private Network (VPN)/Internet Protocol Security (IPsec), or a Transport Layer Security (TLS) tunnel) should be used. Optionally, an additional encryption key pair may be deployed for payload encryption. Its negotiation/provisioning is out of scope for this thesis.

C – Adversary & Threats (T) This section states the Dolev–Yao–style [50] network adversary model with physical node-capture capability and enumerates the concrete threats considered in our analysis.

- **T1** *Global observation:* It is assumed an attacker can passively observe and record traffic on any link (radio or IP), including long-term collection for later analysis.
- **T2** *Active network interference:* It is assumed an attacker can inject, drop, replay, reorder, and delay packets, and mount MITM attacks.
- **T3** *Field-node capture:* It is assumed an attacker can physically capture any fielded node other than the offline CA and the TC1, extracting stored state (private keys, credentials, logs, session caches).

- **T4** *Credential misuse:* It is assumed an attacker can operate a rogue node with valid but constrained credentials (e.g., insider or stolen device).
- **T5** *Revocation/clock abuse:* It is assumed an attacker can delay CertRLs/CredRLs propagation and induce or exploit clock drift beyond bounds to attempt replay/desynchronization.
- **T7** *Cryptographic limits:* Standard primitives (ECDH, ECDSA, HMAC) are not broken, but it is assumed an attacker can exploit weak configurations or implementation errors if present.

D – Operational and Performance Requirements (OPR) Finally, and prior to the system description, it is important to delineate what are the operational and performance requirements of the designed system.

- **OPR1** Session setup, renewal, and handover complete within operationally acceptable bounds on both Wi-Fi and HR-5000H (validated in Chapter 5).
- **OPR2** Local processing overhead per node remains low (validated in Chapter 5).
- **OPR3** *Latency/overhead optimization via protocol layering:* The system employs mTLS only during the session establishment phase (TCP-based) and switches to UDP for all subsequent communication. Thus, session establishment remains cryptographically protected, while steady-state traffic achieves lower latency and reduced transmission overhead suitable for bandwidth-constrained tactical environments.
- **OPR4** *Commercial interoperability:* To guarantee interoperability with commercial technologies, the communication between the GCS Mission Planner software and the UxVs is based on the MAVLink2 protocol. All other nodes that receive UxV telemetry must therefore be capable of parsing and processing MAVLink data.

4.3 NC2S Basic Elements

This section describes the fundamental elements that compose the NC2S framework, including its node roles, message structure, and credential-based mechanisms that support secure communication and control delegation.

4.3.1 NC2S Message Types

The communication within the NC2S system is composed of 16 distinct message types, all of which share a common base format: **(1-byte Message Type, Payload, 8-byte Timestamp, 16-byte HMAC)**.

- (0x01, MAVLink2 Message, Timestamp, HMAC) – Message type used to send and receive MAVLink2 messages.

- (0x02, Waypoint File, Timestamp, HMAC) – Used to transmit Mission Planner waypoint files.
- (0x04, CredRL, Timestamp, HMAC) – Broadcast message used to distribute the CredRL throughout the network.
- (0x05, UxV Common Name (CN), UxV IP, UxV mTLS PORT, UxV UDP PORT, Credential, Timestamp, HMAC) – Message sent by the TC1 or TC2 to a GCS, instructing it to initiate a connection with a UxV.
- (0x06, list(GCS–UxV), Timestamp, HMAC) – Message sent by a GCS to the TC1 or TC2 for network mapping.
- (0x07, CN, Timestamp, HMAC) – Disconnection message used by a node to inform connected peers that it is shutting down.
- (0x08, GCS CN, UxV CN, UxV IP, UxV mTLS PORT, UxV UDP PORT, Credential, Timestamp, HMAC) – Message sent by the TC1 to a TC2, so it instructs a peer GCS to initiate a connection to a UxV.
- (0x09, GCS CN, GCS IP, GCS mTLS PORT, GCS UDP PORT, Credential, Timestamp, HMAC) – Message sent by the TC1 to a TC2, instructing it to initiate a connection with a GCS.
- (0x10, list(GCS–UxV), Timestamp, HMAC) – Message sent by a TC2 to the TC1 for network mapping purposes.
- (0x11, CN/4-byte Salt, Timestamp, HMAC) – Message used for session key renewal.
- (0x12, CN, Timestamp, HMAC) – HEARTBEAT message periodically sent by TC1 and TC2.
- (0x13, Credential, Timestamp, HMAC) – Message sent to a GCS for credential renewal or update purposes.
- (0x14, Credential, Timestamp, HMAC) – Message sent to a TC2 for credential renewal or update purposes.
- (0x15, Credential, Timestamp, HMAC) – Message sent to a UxV for credential renewal or update purposes.
- (0x16, CertRL, Timestamp, HMAC) – Broadcast message containing the updated CertRL.
- (0x17, Certificate, Timestamp, HMAC) – Message carrying a newly renewed certificate.

All messages include a one-byte type identifier, enabling the Python scripts to determine how each message should be processed. An eight-byte timestamp is incorporated to protect against replay attacks. This is ensured by the receiving nodes that store previously received timestamps and compare incoming ones to detect duplicates. If a timestamp is repeated, the message is discarded.

To prevent attackers from exploiting future timestamps, the system also compares each received timestamp against the current system time. If the difference exceeds 400 ms, the message is rejected.

Message authentication and integrity are ensured through a 16-byte HMAC, computed using the SHA-256 algorithm, which binds the session key to the entire message. This guarantees that only nodes possessing the negotiated session keys can generate a valid HMAC, and that the message was not modified during transmission, since the received HMAC is verified against a locally computed HMAC.

As previously discussed, the MAVLink2 protocol also includes a signature field composed of an eight-byte timestamp and a six-byte HMAC. However, this field was considered insufficiently secure, as the shorter HMAC size introduces potential vulnerabilities. Moreover, even if this signature mechanism were used, it would only secure MAVLink message exchanges, leaving other system messages unprotected. In addition, the Mission Planner software, although capable of signing MAVLink2 messages, lacks an integrated session key establishment and management mechanism, requiring manual key insertion for each communication session.

Despite not using the MAVLink2 signature field, all messages reaching either the UxV controller or the SITL instance, as well as the Mission Planner, are ultimately raw MAVLink2 messages. This is achieved because the developed Python scripts can encapsulate or extract the message type, timestamp, and HMAC fields, making the system fully compatible with existing commercial solutions that rely on the MAVLink2 protocol for communication.

4.3.2 NC2S Digital Certificates and CertRL

The initial mutual authentication of the NC2S nodes is performed using mTLS with digital certificates. As in secure Internet communications, it is necessary to define which nodes act as clients and which act as servers, given that clients initiate the connection and authentication process. In the NC2S architecture, the TC1 always operates as an mTLS client when establishing connections with the TC2 and GCS, which act as mTLS servers. Similarly, the TC2 acts as a client in the TC2–GCS connection, while the GCS functions as a client in the GCS–UxV connection.

The digital certificates used in the system are X.509 ECDSA certificates, issued and signed by the CA using ECC keys. The adopted signature algorithm is ECDSA-with-SHA-256. Each certificate is bound to a unique CN and IP address, ensuring the correct identification of the entity using it. Additionally, each certificate includes the timestamp of its issuance and a validity period, allowing nodes to verify its lifetime. To enable signature verification, the CA issues a self-signed certificate from which its public key

can be obtained.

The private key of each node is also generated by the CA, using the `prime256v1` elliptic curve.

The CertRL is requested by the CT1 from the CA. It contains the serial numbers of all revoked certificates and, similar to the certificates themselves, includes both a creation timestamp and a validity period indicating the expiration date of that list.

Given that the system must operate reliably in high-latency environments and withstand temporary communication interruptions, where packet loss is a realistic concern, packet acknowledgment and retransmission mechanisms were deemed unsuitable. For this reason, the system relies solely on TLS for the mutual authentication and secure session establishment phase. Once the session is authenticated and established, subsequent communication is conducted over UDP.

4.3.3 NC2S Credentials, Capacity Policy and credRL

In the NC2S, credentials serve to associate one node with another and to define the permissions governing their information exchange, that is, what data each node can access, send, or receive.
The format of the NC2S credentials is defined as follows:

- (0x01, PU_{TC1}, PU_{TC2}, Capacity String, Credential Lifetime, Timestamp) Sig[Credential, PR_{TC1}] – Credential issued by the TC1 to its link with the TC2, specifying which information may be exchanged between them.

- (0x02, PU_{TC1}/PU_{TC2}, PU_{GCS}, Capacity String, Lifetime, Timestamp) Sig[Credential, PR_{TC1}] – Credential issued by the TC1 for its own connection with the GCS, or for the TC2–GCS link, defining what data may be transmitted and received within those connections.

- (0x03, PU_{GCS}, PU_{UxV}, Capacity String, Lifetime, Timestamp) Sig[Credential, PR_{TC1}] – Credential issued by the TC1 for the GCS–UxV link, defining the allowed data flow between these two entities.

Each credential begins with a one-byte identification field that allows the Python scripts to determine how to process it. The following two fields contain the public keys of the client and server nodes, respectively, enabling each node to verify whether the credential is intended for it. The *Lifetime* and *Timestamp* fields (both 8 bytes) allow the receiver to verify whether the credential is still valid. All credentials are signed with the TC1's private key using the ECDSA algorithm and then hashed to 256 bits via SHA-256, ensuring their authenticity and integrity.

The *Capacity String* field of the credential is directly associated with the capacity policy of the system. The NC2S follows a hierarchical structure in which the TC1 serves as the central authority responsible for defining each node's access rights to UxV data and other system privileges, such as the ability to transmit waypoint files. This capacity policy is conceptually inspired by the STANAG 4586 [51] Level

of Interoperability (LOI) feature, which defines progressive control and data-exchange permissions between GCSs and UxVs across 5 levels.

The NC2S capacity policy combines hierarchical levels of authority with cumulative permission attributes, allowing flexible and fine-grained access control.

The policy is organized into distinct sections, each corresponding to a functional domain, such as drone telemetry, mission data, or communication privileges. Within each section, multiple hierarchical levels define the degree of access granted to a node. These sections can be combined cumulatively, enabling modular configurations that avoid unnecessary over-privileging or complex group definitions.

This combination of hierarchical and cumulative control enhances security and manageability. For example, one credential might include only the capacity string `Access2`, limiting access to basic UxV data but not permitting the transmission of mission files. Another credential could include `Access3, SendWaypoint`, granting slightly broader access that includes mission file handling while maintaining restrictions on more sensitive information.

To revoke a credential, the TC1, through its CT, issues a CredRL, which functions analogously to the CA's CertRL. The format of the CredRL is:

$$(8\text{-byte Timestamp}, \text{Credential Hash List}, 8\text{-byte Lifetime})\ \text{Sig}[CredRL, PR_{TC1}]$$

The *Timestamp* and *Lifetime* fields indicate the validity period of the list and help identify which CredRL was issued first. To conserve space, revoked credentials are identified by their hash values rather than full records. Furthermore, the CredRL adopts the same optimization as the CertRL, automatically removing entries for credentials that have already expired. This mechanism prevents the CredRL from growing indefinitely as more credentials are revoked, maintaining an efficient and scalable revocation process within the system.

4.4 Mission Setup

This section describes the documentation and configuration that each node must possess prior to deployment and connection to the network.

The mission setup assumes that all nodes are physically co-located in a secure environment during the onboarding process. This procedure is conducted either manually or via an imprinting mechanism to ensure controlled and trusted initialization. By the end of the setup, all nodes must have the necessary information to support network initialization.

Before mission initialization, the TC1 defines the nodes IP addresses and creates their configuration files. These files include, among other details, each node CN and associated IP address, which are sent to the CA for certificate generation. The TC1 also defines the UDP and mTLS ports for each

node. After the configuration is complete, the TC1 requests from the CA the digital certificates and the corresponding ECDSA public and private key pairs for all nodes. Additionally, the TC1 requests an empty CertRL, allowing nodes to accept new mTLS connections. In parallel, the TC1 issues an initial empty CredRL, enabling the nodes to accept new credentials during operation. Moreover, the TC1 defines and enforces the policy, which is stored in a JavaScript Object Notation (JSON) configuration file, thus allowing the commander to update the permissions dynamically by issuing an updated policy file without modifying the underlying node scripts.

Finally, the TC1 securely stores and distributes to each node the documentation listed in Table 4.1.

TC1	TC2	GCS	UxV
• CA Certificate • TC1 Certificate • Each TC2 Certificate • Each GCS Certificate • Each UxV Certificate • Empty CertRL • Empty CredRL • TC1 Private Key • Mission Policy JSON file	• CA Certificate • TC1 Public Key • TC2 Certificate • TC2 Private Key • TC2 Server mTLS Port • TC2 UDP Port • Empty CertRL • Empty CredRL • Mission Policy JSON file	• CA Certificate • TC1 Public Key • GCS Certificate • GCS Private Key • GCS Server mTLS Port • GCS Server UDP Port • Mission Planner UDP Ports • Empty CertRL • Empty CredRL • Mission Policy JSON file	• CA Certificate • TC1 Public Key • UxV Certificate • UxV Private Key • UxV Server mTLS Port • UxV UDP Port • ArduPilot SITL TCP Ports • Empty CertRL • Empty CredRL • Mission Policy JSON file

Table 4.1: NC2S Node Documentation

4.5 NC2S Protocols

The following subsections describe in detail the set of communication and security protocols developed within the NC2S framework, outlining their structure, message flow, and functional role in ensuring secure interactions between nodes.

4.5.1 Common Message Verification Steps

Each node follows a common sequence of steps to verify the integrity and authenticity of a received message before processing it. Upon reception, the node first checks whether a session entry exists for the sender's IP address and UDP port. It then verifies whether the message type is permitted under the session's Capacity String. Finally, to ensure integrity and authentication, the node computes the message HMAC using the current session key and compares it with the received HMAC.

Depending on the communication link, each NC2S node may operate either as a client or as a server. Nodes acting as servers (the GCS in the GCS-TC1 and GCS-TC2 links, the TC2 in the TC2-TC1 link, and the UxV in the GCS-UxV link) implement an additional mechanism where if verification with the current key fails, the node checks whether the session is flagged as being in key renewal. If so, it recomputes the HMAC using the pending key and accepts the message if the verification succeeds.

All nodes also verify message freshness by discarding packets with repeated timestamps or timestamps outside the 400 ms acceptance window.

4.5.2 Registration, Authentication, and Session Key Establishment Between Entities

This section is divided into two parts. The first part describes the process of connection establishment and mutual authentication between two nodes. The second part details the transmission of connection initialization messages and the routing of credentials until they reach their destination nodes. The corresponding flowchart for each protocol is provided to enhance visualization at Appendix A.

The process begins with the TC1, which creates the credential associated with each connection. Using its CT, the TC1 selects the client and server nodes, specifies the server node's IP address, mTLS port, and UDP port, defines the credential lifetime (in seconds), and sets the capacity string defining the permitted operations for that communication. Each server node maintains a dedicated port that remains open to accept new mTLS connection initialization requests.

In this first part, it is assumed that the client node has already received the credential and the connection initialization message. The authentication and connection establishment process proceeds as follows (illustrated in Figures A.1 and A.2):

1. **Connection Initialization:** The client node, knowing the server's IP, mTLS port, and UDP port, initiates an mTLS handshake. If the server is unreachable, the client retries up to three times, with each attempt separated by a 20-second interval (Figures A.1 and A.2, Step 1).

2. **Certificate Exchange and Verification:** After the handshake is concluded, both nodes exchange certificates, perform mutual verification and compute the TLS encryption keys (Figures A.1 and A.2, Step 2):

3. **Credential Transmission:** After successful verification, the client sends its assigned credential to the server (Figures A.1 and A.2, Step 3).
4. **Credential Validation:** Upon receiving the credential, the server verifies its legitimacy as it is described in Figures A.1 and A.2, Step 4.
5. **Clock Synchronization:** To account for potential time desynchronization between nodes, the system calculates the offset using the Network Time Protocol (NTP) algorithm (Figures A.1 and A.2, Step 5).
6. **Session Key Derivation:** The system establishes a pair of direction-oriented session keys for HMAC computation. A shared secret (master key) is negotiated via ECDH, combining the nodes public and private keys. The session keys are derived from this shared secret using a HMAC-based Key Derivation Function (HKDF) based on SHA-256, with the "info" field indicating the communication direction (Figures A.1 and A.2, Step 6).
7. **UDP Channel Setup:** Once the session keys are established, the client node informs the server of the UDP port to be used for subsequent data exchange (Figures A.1 and A.2, Step 7).
8. **Session List Creation:** Both nodes create a session record containing all relevant connection data required for UDP communication (Figures A.1 and A.2, Step 8).

However, the system also includes nodes that are not directly connected to the TC1. In these cases, an intermediate routing mechanism is needed so that nodes know which entity to contact and how to receive the appropriate credentials. This is managed through message types 0x05, 0x08, and 0x09, all transmitted via UDP.

On the TC2 side, when it receives a 0x08 or 0x09 message (Figures A.3 and A.5):

1. verifies the received message as described in Figures A.3 and A.5 Steps 3 to 5.
2. Verifies the credential legitimacy as described Figures A.3 and A.5 Step 6.
3. Depending on the message type, the TC2 either initializes a new connection or forwards the credential:
    - If the message is of type 0x08 and the credential type is 0x03 (valid), the TC2 identifies the appropriate GCS by searching on its session lists for the GCS connection associated with the GCS CN field of the 0x08 message. Then encapsulates the received credential and UxV information into a 0x05 message and forwards it to the target GCS (Figure A.5, Step 7).
    - If the message is of type 0x09 and the credential type is 0x02 (valid), the TC2 directly initiates the connection with the GCS following the authentication and key establishment protocol described earlier (Figure A.3 steps 7 to 14).

On the GCS side, when a 0x05 message is received (Figures A.4 and A.5), the the node:

1. Verifies the received message as described in Figures A.5 Steps 8 to 10 and A.4 Steps 3 to 5.
2. Verifies the credential legitimacy as described Figures A.5 step 11 and A.4 step 6.
3. After successful verification, use the UxV information from the 0x05 message to initiate the connection following the authentication and key establishment procedure described earlier (Figure A.4 steps 7 to 14 and Figure A.5 steps 12-19).

4.5.3 Credential Revocation

This section describes the protocol for revoking a node credential within the NC2S. The credential revocation is deployed when the TC1 understands, due to a tactical scenario, that a node should no longer have access to the other node's information. This process does not imply that the node is compromised, and therefore, its digital certificate does not need to be revoked. The overall procedure is illustrated in the Figure A.6.

The credential revocation process is initiated by the TC1 through its CT and proceeds as follows (Figure A.6, Steps 1–3):

1. The TC1 selects, in its CT1, the credentials to be revoked. The ct1 then generates a new CredRL using the format and characteristics described in Section 4.3.3.
2. The CT1 transmits a message of type 0x04 containing the new CredRL to all connected nodes.
3. The CT1 terminates any active sessions associated with the revoked credentials.

The remaining nodes, when receiving a 0x04 follow similar steps:

1. Verify the received message as described in the Figure A.6 steps 4 to 6, 11 to 13 and 18 to 20.
2. After successful authentication, verify and store the received CredRL as described in the Figure A.6 steps 7 to 8, 14 to 15 and 21 to 22.
3. Encapsulate the received CredRL into a new 0x04 message and broadcast it to all subordinate nodes in the hierarchy as described in the Figure A.6 steps 9 and 16.
4. Check if any of the credential hashes present on the received CredRL are associated with any of its connections. If a match is found, terminate the corresponding connection and delete the revoked credential from memory as described in the Figure A.6 steps 10, 17 and 23.

By broadcasting the 0x04 message across the network, this protocol increases the probability of all nodes receiving the latest credential revocation list, thereby maintaining consistent authorization status and preventing unauthorized access within the system.

4.5.4 Certificate Revocation

This protocol is to be employed as a last-case scenario when a node is captured or defined as being compromised. In such situations, it becomes necessary to remove the affected node from the network and prevent it from rejoining. This is achieved by revoking the node's digital certificate and broadcasting an updated CertRL to all remaining nodes.

The certificate revocation process proceeds as follows (Figure A.7, Steps 1–3):

1. The TC1, through its CT1, requests the CA to revoke the digital certificate of the compromised node. The CA then responds by issuing an updated CertRL containing the serial number of the revoked certificate.
2. The CT1 stores the updated CertRL, constructs a new 0x16 message containing it, and broadcasts this message to the network.
3. The CT1 then checks for any active sessions associated with the revoked certificate serial numbers and terminates the corresponding connections.

The remaining nodes, when receiving a 0x16 follow similar steps:

1. Verify the received message as described in the Figure A.7 steps 4 to 6, 11 to 13 and 18 to 20.
2. Upon successful authentication, verify and store the received CertRL as described in the Figure A.7 steps 7 to 8, 14 to 15 and 21 to 22.
3. Encapsulate the verified CertRL into a new 0x16 message and broadcast it to all subordinate nodes in the hierarchy (Figure A.7 step 9 and step 16).
4. Check whether any stored certificates correspond to serial numbers listed in the received CertRL. If so, terminate the affected connections (Figure A.7 steps 10, 17 and 23).

This broadcast mechanism, combined with immediate connection termination, ensures the prompt isolation of compromised nodes. Furthermore, the presence of the updated CertRL prevents the revoked node from rejoining the network, since the mTLS authentication process verifies whether the peer's certificate serial number appears in the stored CertRL.

4.5.5 Credential Capacities Update Protocol

This protocol addresses situations in which the TC1 needs to modify a node's level of access to another node's information due to tactical or operational factors. The goal is to update the capacity string of an active credential without revoking it and forcing nodes to reauthenticate. The protocol therefore enables a smooth update of access permissions by creating and distributing a new credential with an updated

capacity string, routing it to the correct node, and ensuring that the receiving node verifies and replaces its stored credential with the new one.

The process begins with the TC1, which, through its CT interface, accesses the *Change Capacity* menu. After selecting the credential type and the target nodes, the CT1 generates a new credential for the selected connection. Once the credential is created, it must be forwarded to the destination nodes. This is achieved by encapsulating the credential within 0x13, 0x14 or 0x15 messages, depending on the credential category and destination.

The CT1 determines the correct forwarding path by analyzing the credential type and the destination node CN selected in the *Change Capacity* menu (Figure A.8, Step 1).

On the TC2 side, when a `0x14` message is received, the following operations are performed:

1. Verify the received message as described in the Figure A.8, Steps 2–4.
2. Verifies the credential legitimacy as described in the Figure A.8, Step 5.
3. Process the credential according to its type byte (Figure A.8, Step 6):

On the GCS side, when a `0x13` message is received, the node performs the following operations:

1. Verify the received message as described in Figure A.8 steps 7–9.
2. Verifies the credential legitimacy as described in Figure A.8 step 10.
3. Process the credential according to its type byte (Figure A.8, Step 11):

Finally, **when a UxV receives a** `0x15` message, the following steps are executed:

1. Verify the received message as described in Figure A.8 step 12-14.
2. Verifies the credential legitimacy as described in Figure A.8 step 15.
3. If the credential is of type 0x03, verify that the server public key field matches its own public key and that the client public key corresponds to the GCS public key. If valid and newer, replace the stored credential with the received one (Figure A.8 step 16).

This protocol enables dynamic adjustment of node access levels without disrupting active communications. By using credential replacement rather than revocation, it maintains mission continuity while preserving the security guarantees of mutual authentication and credential freshness.

4.5.6 Control Delegation Protocol

As previously discussed, the main objective of this master's thesis is to design and implement a secure and lightweight system that allows the TC1 to delegate or handover UxV control. After analyzing existing

architectures and solutions for control handover mechanisms, particularly those reviewed in Section 3.3, the NC2S control delegation protocol was developed. This section describes the protocol in detail.

The protocol contemplates two possibilities. One more restrictive involving credentials revocation and another, more flexible, based on modifying the capacity string of active credentials.

4.5.6.A Control Delegation Protocol with Credential Revocation

This approach addresses the scenario in which the control of a UxV must be transferred from one GCS (currently in control) to another GCS that is not yet connected to the UxV. Such situation typically arises from tactical implications that also require the previous GCS to lose access to the UxV.

The protocol begins with the TC1, which, through its CT1 interface, initiates a new connection process for the new GCS to the UxV. Here the TC1 specifies the UxV IP address and port configuration, together with a capacity string that grants the new GCS control privileges over the UxV. The CT1 and the involved nodes then execute the connection establishment and authentication procedure described in Section 4.5.2 until the new GCS successfully establishes a session with the UxV.

Subsequently, the TC1, through its CT1, revokes the credential of the previous GCS that granted it control over the UxV, following the procedure defined in Section 4.5.3. As a result, control is securely transferred to the new GCS, while the previous GCS loses all control privileges.

This method adheres to the zero-trust principle where even if the previous GCS fails to acknowledge the revocation command from the TC1, the UxV will eventually receive the broadcast CredRL from the new GCS and disconnect from the revoked one, ensuring control integrity.

4.5.6.B Control Delegation Protocol with Capacity String Modification

The second approach is designed for cases where two GCS nodes are already connected to the same UxV. Here, one GCS currently holds control, and the TC1 intends to transfer control to the second GCS while allowing the first to retain a lower level of access. This approach eliminates the need for full reauthentication or credential revocation, thereby minimizing handover latency.

The handover process is based on the credential update mechanism described in Section 4.5.5. Through the CT1 network map interface, the TC1 identifies both GCS nodes and generates new credentials for each. The updated credentials contain modified capacity strings that grant the new GCS control privileges while downgrading the previous GCS to a restricted access role.

This approach preserves system security through standard message and credential authentication processes, while significantly reducing the operational time associated with credential revocation and full session reestablishment.

In both approaches, there may be a brief transition period during which both GCS nodes possess control capability over the same UxV. To prevent conflicting commands during this interval, it is expected

that both GCS operators coordinate via voice communication.

4.5.7 Session Key Renewal Protocol

To enhance system security, each established session key pair in the NC2S is assigned a finite lifetime. When a session key lifetime approaches its expiration threshold, the keys must be renewed to maintain secure communication. This section describes the renewal process for a client–server connection.

Each node periodically executes a routine that checks whether any session key pair is nearing its expiration. On the server side, once a session key reaches the end of its lifetime, the server terminates the connection associated with that key. On the other hand, when a client node detects that a session key for a given server connection is nearing a predefined threshold, it initiates the session key renewal protocol. This threshold ensures that renewal occurs before key expiration.

1. **Client Node Behavior:**
   (a) When the client detects that a session key pair has reached the renewal threshold (Figure A.9, Step 2):
      i. It sends a `0x11` message to the server node, signaling the start of the key-renewal process and requesting the derivation of new session keys.
      ii. It adds the server node to a *key-renewal list*, storing the renewal start time and retry counter.
   (b) The client periodically checks the renewal list (Figure A.9, Step 1):
      i. If renewal has been pending for more than 30 seconds:
         A. If the retry count < 3, resend the `0x11` message.
         B. If the retry count > 3, close the connection and restart the full authentication and session-establishment procedure, as described in Section 4.5.2, using the same credential as before.
2. **Server Node Behavior on Receiving a `0x11` Message**:
   (a) Verify the received message as described in Figure A.9, Steps 3–5.
   (b) Derive a new pair of session keys from the original shared secret using HKDF-SHA-256, with a freshly generated 4-byte random nonce as salt. Store these keys alongside the existing (about-to-expire) keys, and mark the session as being in a **key-renewal state** as described in Figure A.9, Step 6.
   (c) Send a `0x11` response message containing the nonce used for key derivation (Figure A.9, Step 7).

3. **Client Node Behavior on Receiving the Server's `0x11` Response**:

   (a) Verify the received message as described in Figure A.9, Steps 8–10.

   (b) Check that the sender is present in the key-renewal list. If so, derive the new session key pair from the shared secret using the received nonce, then update the session list by replacing the old keys with the newly derived pair (Figure A.9, Step 11).

   (c) From this point onward, all client messages use the new key pair for HMAC computation.

4. **Final Confirmation by the Server Node**:

   (a) Verify the received message as described in Figure A.9, Steps 13-14.

   (b) When verifying a message HMAC from the client, and since it is a server node and the session is flagged as being in a key-renewal state, it first computes the HMAC using the current (old) key. If it matches, continue normal operation. If it does not match, recompute the HMAC using the newly derived key pair. If it matches, this confirms that the client has successfully derived the new keys (Figure A.9, Step 15).

   (c) Update the session state by promoting the newly derived keys to become the active key pair and exiting the key-renewal state to resume normal communication as described in Figure A.9, Step 16. The session key-renewal process is then complete.

This mechanism ensures that no sensitive information is transmitted between nodes during key renewal. The periodic key-lifetime verification routines guarantee that, if renewal is not completed in time, the connection is safely terminated. The retry counter also provides resilience in high-latency environments where a `0x11` message might be lost, preventing the client from remaining indefinitely in a key-renewal state.

4.5.8 Credential Renewal Protocol

As previously discussed, each credential in the NC2S includes a lifetime parameter that defines its validity period. Therefore, credentials must be renewed before their expiration to maintain uninterrupted communication and access control integrity.

The TC1, through its CT1, maintains tracking of issued credentials in two distinct ways. First, for its direct connections with TC2 and GCS nodes, it stores in the session list the credential's timestamp, lifetime, capacity string, and the corresponding public keys of the connected nodes. Second, for credentials issued to TC2–GCS and GCS–UxV connections, each active type 0x02 or 0x03 credential is recorded in a separate list containing the credential's data along with the routing path.

The CT1 script periodically executes a routine that inspects both the directly connected nodes' session lists and the credential lists of indirectly connected nodes. This check determines whether any

credential lifetime is approaching its expiration threshold. Once a credential reaches this threshold, the CT initiates the credential renewal protocol, analogous to the session key renewal process described earlier.

For each credential nearing expiration, the CT generates a new credential containing the same parameters as the currently active one, but with an updated timestamp and a newly computed digital signature. The forwarding of these new credentials follows the routing scheme defined in Section 4.5.5, utilizing message types 0x13, 0x14, and 0x15 depending on the destination node hierarchy.

This protocol ensures the continuous validity of credentials without requiring reauthentication or interruption of ongoing sessions, maintaining both operational efficiency and the security guarantees of authenticated credential management.

4.5.9 Certificate Renewal Protocol

Each certificate in the NC2S system has a defined lifetime value. Therefore, it is necessary to renew the nodes' certificates before they expire to prevent authentication failures during mission execution.

The CT1 script includes a periodic routine that checks the *next update* field of the stored certificates. When a certificate approaches its renewal threshold, the CT1 requests from the CA a new certificate for the corresponding node, preserving the same identification information as the previous one (Figure A.10, step 1). After obtaining the new certificate, the CT1 determines the forwarding path by combining the certificate's CN with the entries in the network map (Figure A.10, step 2).

TC2 behavior when receiving a 0x17 message:

1. Verify the received message as described in Section 4.5.1 (Figure A.10, steps 3–5).
2. Verify the certificate authenticity as described in Figure A.10, step 6.
3. Process the certificate according to the destined node as described in Figure A.10, step 7.

GCS behavior when receiving a 0x17 message:

1. Verify the received message as described in Section 4.5.1 (Figure A.10, steps 8–10).
2. Verify the certificate authenticity as described in Figure A.10, step 11.
3. Process the certificate according to the destined node as described in Figure A.10, step 12.

UxV behavior when receiving a 0x17 message:

1. Verify the received message as described in Section 4.5.1 (Figure A.10, steps 13–15).
2. Verify the certificate authenticity as described in Figure A.10, step 16.

3. Compare the node's own CN and public key with those in the received certificate. If they match, and the received certificate was issued after the stored one, replace the old certificate with the new one (Figure A.10, step 17).

This protocol ensures that all nodes in the hierarchical architecture receive updated certificates before their expiration. By relying on authenticated forwarding through 0x17 messages and hierarchical propagation, it maintains continuous trust across the NC2S network without requiring reinitialization of secure sessions.

4.5.10 CertRL Renewal Protocol

Certificate revocation within the NC2S system is managed through the creation and distribution of a CertRL. To ensure that all nodes maintain an up-to-date list containing the serial numbers of revoked certificates, each CertRL includes a defined lifetime parameter.

The CT1 is the entity responsible for requesting CertRL renewal from the CA. Accordingly, its script implements a periodic routine, similar to the other renewal mechanisms in the system, that verifies whether the currently stored CertRL is nearing the end of its lifetime. When the CertRL reaches a predefined threshold, the CT1 requests from the CA the creation of a new CertRL, which includes the serial numbers of all previously revoked certificates. The CA generates the new list and returns it to the CT1.

After receiving the renewed CertRL, the CT1 and the remaining NC2S nodes execute the procedure described in Section 4.5.4, broadcasting the updated CertRL throughout the network and replacing the outdated version on each node. This ensures that all nodes continuously share a consistent and updated view of certificate validity across the entire hierarchical system.

4.5.11 CredRL Renewal Protocol

Similarly to the CertRL, the CredRL also has a defined lifetime. Its renewal process follows the same principle, with the key distinction that it does not require interaction with an external entity. In this case, the CT1 itself is responsible for generating and distributing new credential revocation lists.

The CT1 script includes a routine that periodically verifies the lifetime of the stored CredRL. When the list approaches a predefined threshold, the CT evaluates the entries of the previous revocation list, identifying which credentials remain within their validity period. Any expired credentials are removed from the list. Subsequently, a new CredRL is created containing only the still valid revoked credentials.

The dissemination of the updated CredRL throughout the system follows the same procedure described in Section 4.5.3, ensuring that all nodes maintain an up-to-date view of credential validity and preventing the reauthorization of previously revoked nodes.

4.5.12 Network Mapping Protocol

The CT1 and CT2 include a feature in their GUI that displays the network topology. This functionality is essential for commanders, as it provides a real-time visualization of the nodes currently connected within the system and the potential routing paths between them. The protocol supporting this feature relies on the exchange of message types `0x06` and `0x10`.

Each GCS periodically sends a `0x06` message containing the list of connected UxVs. This message is delivered either to the CT2 nodes or directly to the CT1, depending on the current network configuration.

When a CT2 receives a `0x06` message, it verifies the received message as described in Section 4.5.1 and then merges the received `0x06` GCS–UxV connection list with the existing lists previously received from that same GCS.

In addition, each CT2 periodically sends a `0x10` message to the CT1. This message contains the aggregated network map, constructed from all `0x06` messages collected from the GCS nodes. The CT2 also uses this data to display the current network map on its GUI, showing the nodes to which it is directly or indirectly connected.

Finally, when the CT1 receives a `0x06` or `0x10` message, it verifies the received message as described in Section 4.5.1 and consolidates the node connection data from both `0x06` and `0x10` messages into a unified network map list.

This consolidated list is then used to display the network map on the CT1 GUI. It also supports routing decisions for connection initialization messages (Section 4.5.2), as well as the routing of credentials (4.5.5) and certificates (Section 4.5.9). The protocol flow is illustrated in the flowchart presented in Figure A.11.

4.6 GUI

In order for both TC1 and TC2 to access and visualize the COP, it was necessary to design a dedicated interface that enables secure synchronization and coordinated mission awareness between commanders.

4.6.1 CT1 GUI

The CT1 GUI, illustrated in Figure A.12, provides the commander with an integrated control interface for managing and monitoring the entire NC2S network. It contains the following key features:

- **Network Map** — (Label 1 in Figure A.12) — Displays the network topology, showing all active nodes and their connections. The map is dynamically updated through the protocol described in

Section 4.5.12.

- **Satellite Image** — (Label 2 in Figure A.12) — Shows the real-time geographical positions of the connected UxVs during mission execution.
- **Telemetry Notebooks** — (Label 3 in Figure A.12) — Present the live telemetry data of each UxV, allowing the commander to monitor flight parameters and mission status.
- **Buttons Area** — (Label 4 in Figure A.12) — Contains all command and control buttons. These include the **"New Connection"** button, which initializes the session establishment protocol described in Section 4.5.2. The **"Access Revocation"** button is responsible for revoking the selected credentials through the protocol described in Section 4.5.3. The **"Change Capacity"** button triggers the protocol presented in Section 4.5.5. The **"Send Waypoint Mission"** button is used to send waypoint files in the Mission Planner format. Finally, the **"Revoke Certificates"** button allows the commander to revoke the selected certificates using the procedure described in Section 4.5.4.

4.6.2 CT2 GUI

The CT2 GUI, illustrated in Figure A.13, provides a simplified yet functional interface for mission monitoring and control operations. It includes the following main features:

- **Network Mapping** — (Label 1 in Figure A.13) — Similar to the CT1 interface, this component visualizes the current network topology and node connections. It is dynamically updated according to the protocol described in Section 4.5.12.
- **Satellite Image** — (Label 2 in Figure A.13) — Displays the real-time geographic positions of the connected UxVs during mission execution.
- **Telemetry Notebooks** — (Label 3 in Figure A.13) — Present the live telemetry data of each UxV, enabling the commander to monitor flight and mission parameters.
- **Buttons Area** — (Label 4 in Figure A.13) — Contains the **"Send Waypoint Mission"** button, which functions in the same manner as in the CT1 GUI. When pressed, it allows the user to select a mission file and send it to the desired node.

4.7 Threat Analysis

This section provides an informal evaluation of the security properties of the proposed NC2S system against a Dolev–Yao network adversary [50] and the common cyberattacks identified in Section 2.5. Each subsection describes a specific attack scenario and the corresponding mitigation adopted in NC2S.

4.7.1 Eavesdropping Attack (Confidentiality)

Scenario: An attacker passively records radio/Wi-Fi or tactical links to extract mission data or telemetry.
Mitigation: As stated in Section 4.1, confidentiality is provided by packet encryption at the transport layer. This prevents the adversary from reading intercepted traffic, even under full channel observation.

4.7.2 Message Tampering Attack (Integrity)

Scenario: An active attacker modifies fields or injects forged NC2S packets to alter command semantics or disrupt message flow.

Mitigation: Each message carries an HMAC computed over the complete messages fields using direction-specific session keys. Any packet failing verification is immediately discarded, ensuring message integrity and origin authenticity.

4.7.3 Impersonation Attack (Peer Spoofing)

Scenario: The adversary pretends to be a valid TC, GCS, or UxV during session setup or steady-state communication.
Mitigation: Certificate and credential verification during session establishment, together with per-packet HMAC, binds each message to an authenticated peer. Spoofing attempts without valid cryptographic material are rejected.

4.7.4 Replay Attack

Scenario: The attacker replays or reorders previously valid packets to trigger outdated commands or confuse session state.
Mitigation: Each message includes a timestamp validated within a bounded acceptance window. Out-of-window or duplicate timestamps are ignored, effectively preventing replayed messages.

4.7.5 MITM Attack

Scenario: The attacker interposes between peers to relay or alter handshake and data messages, creating parallel or altered sessions.
Mitigation: Mutual authentication through certificates and credentials, coupled with HMAC-protected traffic, prevents undetected alteration or session splicing. Any message not matching the expected HMAC or sequence is rejected.

4.7.6 Desynchronization Attack / Availability in Constrained Environments

Scenario: In this attack, the adversary intercepts or blocks packets during the key or credential update phase, causing one peer to complete the update while the other fails to do so, resulting in communication loss.

Mitigation: The NC2S implements bounded retries, authenticated renewal timers, and automatic re-initialization procedures that allow both peers to re-establish synchronized state. These mechanisms sustain availability even under intermittent or degraded tactical links such as narrowband radios.

4.7.7 Privilege Escalation Attack (Unauthorized Control or Information Access)

Scenario: The attacker attempts to access data or send commands beyond the privileges granted to its role, or performs an unauthorized control handover.

Mitigation: Each credential embeds a capacity string defining the maximum operational privileges. Every message is verified against these capacity fields and the associated credential validity. Requests or commands exceeding the authorized capacity, or issued under revoked credentials, are denied. This enforces fine-grained access control and prevents unauthorized command or information disclosure.

4.7.8 Key and Node Compromise Attack

Scenario: The adversary compromises a node, either remotely through software intrusion or physically via capture, to extract certificates, credentials, or session keys, seeking to impersonate the node or decrypt past communications.

Mitigation: The NC2S confines the impact of compromise through isolation and renewal. Revocation Lists (CertRL and CredRL) promptly invalidate the compromised node's certificates and credentials across the network, removing its authorization to participate in further exchanges. Additionally, short credential lifetimes, periodic key renewals, and per-link, direction-specific session keys limit the exposure window of any leaked material.

4.8 Usability Analysis

Table 4.2 compares the proposed NC2S with selected systems reviewed in Section 3, considering the attack scenarios analyzed previously and key operational requirements. A ✓ indicates that the paper addresses or protects against the attack (or supports the feature), while "x" means it does not.

Feature	[43]	[38]	[41]	[47]	NC2S
Eavesdropping	✓	✓	✓	✓	✓
Message Tampering	✓	✓	✓	✓	✓
Replay Attack	x	✓	✓	x	✓
Impersonation Attack	✓	✓	✓	✓	✓
MITM	✓	✓	✓	✓	✓
Desynchronization Attack / Availability in Constrained Environments	x	✓	✓	x	✓
Privilege Escalation	x	x	✓	x	✓
Node Capture Attack	✓	✓	✓	x	✓
UxV Control Handover	✓	✓	x	✓	✓
Integration with Commercial Systems	x	x	x	✓	✓
Dynamic Node Join (New Nodes After Network Launch)	x	✓	✓	x	✓
Experimental Validation	x	x	✓	✓	✓

Table 4.2: Comparison between the proposed NC2S system and selected related works.

The NC2S stands out for enabling secure control handover through a hierarchical capacity policy that prevents privilege escalation. It was also experimentally validated with a working prototype, whereas most related systems remain theoretical or simulation-based. Moreover, NC2S is directly compatible with commercial solutions such as ArduPilot Mission Planner and UxVs using the MAVLink protocol.

4.9 Summary

This chapter introduced the design and implementation of the proposed NC2S framework, a secure and flexible C4I solution for heterogeneous UxV networks. It described the system architecture, entities, and design requirements, detailing the main components such as message types, certificates, credentials, and capacity policies. The complete set of communication protocols was presented, including registration, authentication, key establishment, renewal, revocation, delegation, and network mapping, ensuring adaptability and security across mission phases. Lastly, an informal security analysis demonstrated resistance to major cyber threats. Overall, this chapter established the foundation of the NC2S architecture and prepared the ground for its experimental evaluation in the following chapter.

5

Testing Methodology and Results

Contents

After designing and implementing the NC2S system, its performance and applicability were tested under different architectures to evaluate its real-world behavior and validate its operational efficiency. The tests focused on connection establishment, handover latency, key renewal time, communication reliability, and the network's throughput, bandwidth, jitter, packet loss, and latency using `iperf3`.

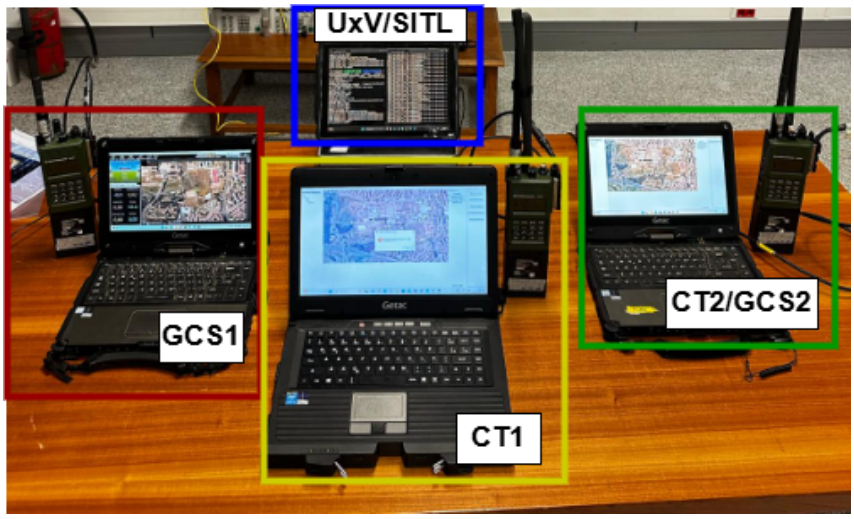

Figure 5.1: System Prototype Testing Environment.

The prototype was composed of four nodes, each representing a system entity in the NC2S architecture:

- **PC1 – TC1:** Getac (Windows), Intel i5-4210M @ 2.6 GHz, 8 GB RAM, 256 GB SSD.
- **PC2 – GCS1 and PC3 – TC2/GCS2:** Getac (Windows), Intel i5-6300 @ 2.5 GHz, 8 GB RAM, 256 GB SSD.
- **PC4 – UxV:** Microsoft Surface 6 Pro (Windows), Intel i7, 16 GB RAM, 512 GB SSD.

For the Wi-Fi architecture, the system operated using an access point from the Army's internal network, the Cisco AIR-CAP1702I-E-K9 (*802.11ac*) model. This dual-band access point supports MIMO 2×2:2 operation, a maximum aggregated data rate of 867 Mb/s in the 5 GHz band, and up to 300 Mb/s in 2.4 GHz, featuring beamforming, WPA2-Enterprise security, and internal omnidirectional antennas. It ensured stable short-range connectivity for the Wi-Fi tests.

For the radio-based configuration, each HR-5000H radio was connected to a dedicated PC through Ethernet. The radios operated as network gateways, creating independent subnets per node. Each PC was configured with IP `10.10.`n`.20`, and its corresponding radio acted as the default gateway with IP `10.10.`n`.51`, where n identifies the node. This design enabled complete logical isolation between links, mirroring field deployment conditions.

The radios used were the Rohde & Schwarz HR-5000H (see Section 2.6.3.B) units operating with the SOVERON® WAVE TNW50 Tactical Network Waveform in Robust Mode. The radio operated inside the 235-238.150 MHz band, divided into 64 channels spaced by 50 kHz, employing both COMSEC and TRANSEC in order to simulate a real battlefield implementation. All radios were configured in single-op mode.

5.1 Iperf3 Tests and Ping Results

To characterize both communication infrastructures, a series of `iperf3` tests were conducted between PC1 and PC2, alternately configured as client and server. The collected metrics included TCP/UDP goodput, packet loss, jitter, and latency. The communication was tested in a Wi-Fi setup and in a HR-5000H setup.

In order to determine the channel goodput in UDP, the server node was instructed to send packets at a certain rate. Then, in both setups, the effective UDP goodput was defined as the highest transmission rate yielding a packet loss of 1% or less. For the Wi-Fi setup, tests were executed at 100, 50, and 40 Mb/s, while for the HR-5000 radios (TNW50 Robust Mode, $\approx$10 kb/s user rate), equivalent tests were performed at 15, 10, 5, and 4 kb/s.

Complementary `ping` tests were used to evaluate Round-Trip Time (RTT) latency. Each experiment was repeated five times, and the mean results were computed to ensure statistical consistency.

Table 5.1 summarizes the obtained results. In the Wi-Fi configuration, the TCP throughput reached between 25.5 Mb/s and 28.8 Mb/s in both link directions, which aligns with typical performance expectations for a short-range IEEE 802.11ac network under controlled conditions. The UDP tests revealed a goodput between 39.6 Mb/s and 73.0 Mb/s depending on the offered bandwidth, with packet losses remaining below 1% for rates up to approximately 40 Mb/s. Jitter values were consistently low (below 1.2 ms), confirming the stability and low latency of the Wi-Fi channel. The mean RTT measured via `ping` was approximately 5.5 ms, further validating the low-latency nature of this link.

In contrast, the HR-5000H radios presented significantly lower performance due to the intrinsic constraints of the TNW50 Robust Mode waveform. TCP throughput tests were inconclusive. One possible justification is that the high link latency caused repeated Retransmission Timeout (RTO), leading to buffer saturation and session termination. This behavior is consistent with the waveform's design for robustness rather than throughput efficiency.

UDP tests on the HR-5000 showed throughput values between 3.8 kb/s and 5.0 kb/s, depending on the offered rate. These figures are in line with the 10 kb/s user data rate specified for Robust Mode. Despite the limited bandwidth, packet loss remained low (below 3%) for all test cases, demonstrating good reliability under constrained channel conditions. However, jitter values were substantially higher,

ranging from 55 ms up to 780 ms, reflecting the long and variable transmission times typical of narrow-band tactical links. The measured RTT values (839–968 ms) further confirm the high-latency nature of this communication medium.

Overall, these results provide an empirical reference for interpreting the performance of the NC2S protocol stack under both infrastructures. While Wi-Fi offers a high-throughput, low-latency channel ideal for testing and short-range operations, the HR-5000 link realistically simulates the constrained and delay-prone environments expected in field deployments. Consequently, the behavior observed in the subsequent tests (e.g., connection establishment and key renewal times) directly reflects these underlying physical-layer characteristics.

	PC1–PC2			**PC2–PC1**		
	Throughput	**Packet Loss**	**Jitter**	**Throughput**	**Packet Loss**	**Jitter**
Wi-Fi						
TCP	25,5 Mb/s	–	–	28,8 Mb/s	–	–
UDP test 100 Mb/s	52,54 Mb/s	13,158 %	0,1746 ms	73,02 Mb/s	7,14 %	0,1489 ms
UDP test 50 Mb/s	47,34 Mb/s	2,582 %	0,6134 ms	47,62 Mb/s	4,156 %	0,783 ms
UDP test 40 Mb/s	39,64 Mb/s	0,554 %	1,2108 ms	38,34 Mb/s	3,168 %	0,4502 ms
Ping	5,5 ms			5,5 ms		
HR-5000H Radio						
TCP	–	–	–	–	–	–
UDP test 15 kb/s	4,924 kb/s	3,08 %	752,57 ms	5,008 kb/s	0 %	783,53 ms
UDP test 10 kb/s	4,792 kb/s	2,20 %	444,37 ms	4,962 kb/s	0 %	423,52 ms
UDP test 5 kb/s	4,578 kb/s	0 %	60,63 ms	4,574 kb/s	0 %	49,94 ms
UDP test 4 kb/s	3,894 kb/s	0 %	55,43 ms	3,878 kb/s	0 %	64,35 ms
Ping	968 ms			839,75 ms		

Table 5.1: Mean *iperf3* results for TCP and UDP tests between the nodes using WiFi and HR-5000H radio links, measured in both directions (PC1–PC2 and PC2–PC1).

5.2 Session Establishment Time

This experiment measured the session establishment time between nodes. Its objective was to determine the time required for a node to join the network after the CT1 initiated or ordered the connection, as described in Section 4.5.2. The overall test architecture is illustrated in Figure 5.2 and was evaluated in two different scenarios. In scenario 1, all nodes communicated over Wi-Fi (links 1–4 in Figure 5.2). In scenario 2, the command links (TC1–TC2, TC1–GCS, and TC1–TC2–GCS) communicated through HR-5000H radios, while the Wi-Fi link was maintained only for the GCS–UxV connection (links 5–8 in Figure 5.2).

Time measurements for all interval-based experiments were obtained through a centralized timestamping method. Each NC2S node was configured to send dedicated event messages through a temporary socket to a Wi-Fi server operating as a reference clock. Upon reception, the server logged both the

message content and its precise arrival timestamp (in μs). The duration of each connection establishment was then calculated by subtracting the timestamps of the *start* and *completion* events recorded by the server. This approach ensured that all nodes were measured against the same synchronized time source, effectively eliminating clock-drift errors between devices. Each connection test was repeated seven times to enable the computation of statistical parameters such as mean, variance, and standard deviation.

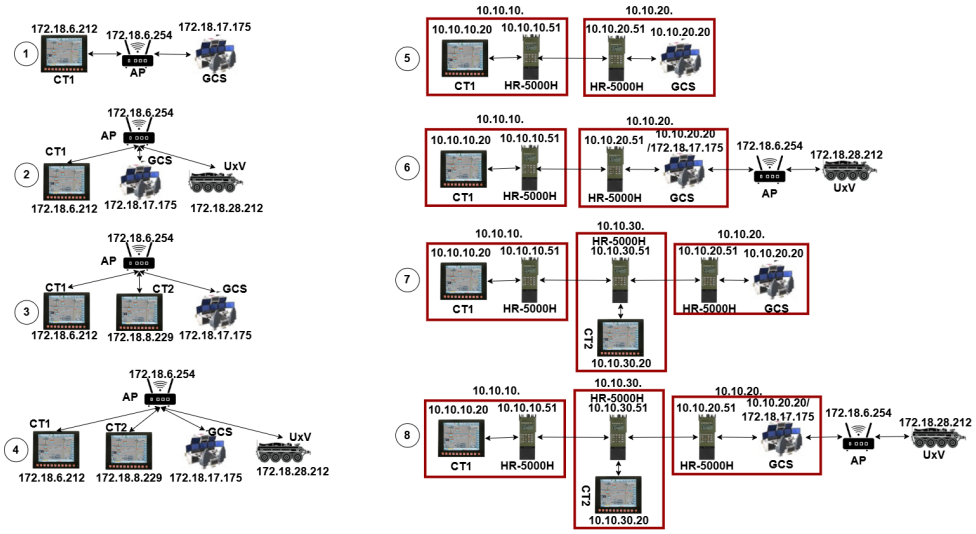

Figure 5.2: Connection Time, Key Renewal, System Reliability, Goodput Estimation and CPU Processing Time Testing Architecture.

5.2.1 TC1-GCS

As presented in Table 5.2, the connection establishment between TC1 and the GCS exhibits a clear difference in performance between the Wi-Fi and HR-5000H communication links. The average connection time over Wi-Fi was approximately 298185 (about 0.30 s), whereas over the HR-5000H it increased to more than 20214282 (approximately 20 s).

Again, the Wi-Fi connection demonstrated low variability, with a standard deviation of about 111593,65 μs, indicating a stable and predictable setup process. In contrast, the HR-5000 results show a much higher dispersion, with a standard deviation of 1158999,94 μs.

	Mean (μs)	Variance $(\mu s)^2$	Standard Deviation (μs)
Wi-Fi	298185	1,25x10^{10}	111593,65
HR-5000H	16137257,5	1,34x10^{12}	1158999,94

Table 5.2: Mean, variance, and standard deviation results for the connection between TC1 and GCS, measured in microseconds (μs), using WiFi and HR-5000H.

5.2.2 TC1-TC2

As shown in Table 5.3, the connection establishment between TC1 and TC2 also presents a clear difference in performance between the Wi-Fi and HR-5000H links. Over Wi-Fi, the mean connection time was 195171 (approximately 0.20 s), whereas the HR-5000H required 1652255 (about 1.65 s).

The variability of the Wi-Fi results remained low, with a standard deviation near 30 ms, reflecting the predictable behaviour of a broadband channel. In contrast, the HR-5000H showed a standard deviation above 1.4 s, indicating considerable fluctuation in the setup process. This increase in both mean and variance follows the same pattern observed in the previous subsection, resulting from the radio's narrowband operation and link-initialization delays inherent to the TNW50 waveform.

	Mean (μs)	Variance $(\mu s)^2$	Standard Deviation (μs)
Wi-Fi	195171	9,25x10^8	30410,90
HR-5000H	1652255	2,02x10^{13}	1420645,43

Table 5.3: Mean, variance, and standard deviation results for the connection between TC1 and TC2, measured in microseconds (μs), using WiFi and HR-5000H.

5.2.3 TC1-TC2-GCS

Table 5.4 presents the connection establishment results for the multi-hop scenario where TC1 communicates with the GCS through an intermediate TC2 node. Three intervals were measured:

1. (1) — Time since the 0x08 message is sent by the TC1 until it is received by the TC2.
2. (2) — Time since TC2 initiates the mTLS connection until the GCS completes the session list creation, marking the end of the connection establishment phase.
3. (3) — Time since TC1 sends the 0x08 message until the GCS completes the connection establishment protocol.

Over Wi-Fi, the transmission of the 0x08 message required only 4.1 ms, while the TC2–GCS connection and the complete TC1–GCS interval averaged 0.55 s and 0.57 s, respectively. These values confirm that the addition of one hop introduces negligible delay in a high-throughput, low-latency network.

In contrast, under the HR-5000H link, the 0x08 message transmission increased to approximately 0.83 s, while the TC2–GCS connection required about 17.9 s. The total TC1–GCS interval reached nearly 18.7 s, demonstrating that the additional hop significantly impacts the overall connection time. The difference between (2) and (3) shows that the 0x08 forwarding delay directly adds time to the total establishment time.

Also, under the HR-5000H links, the standard deviations observed, around 48 ms for the 0x08 transmission and above 2.7 s for the connection phases, are in accord with the jitter measured values mea-

sured in the iperf3 tests. These results confirm that, unlike in Wi-Fi, even a single additional hop under HR-5000H considerably increases both latency and variability in the connection setup process.

	(1)	(2)	(3)
Scenario 1			
Mean (μs)	4148	555571	570310
Variance $(\mu s)^2$	4,45x10^8	5,92x10^9	5,84x10^9
Standard Deviation (μs)	2110,17	76949,01	76439,85
Scenario 2			
Mean (μs)	832987,5	17930313,75	18774698,75
Variance $(\mu s)^2$	2,35x10^9	7,96x10^{12}	7,75x10^{12}
Standard Deviation (μs)	48489,05	2820783,81	2783562,86

Table 5.4: Mean, variance, and standard deviation results for: (1) TC1–TC2 0x08 message transmission, (2) TC2–GCS connection, and (3) TC1–GCS connection.

5.2.4 TC1-GCS-UxV

Table 5.5 presents the results for the connection establishment sequence involving TC1, the GCS, and the UxV. Three intervals were measured:

1. (1) — Time since the 0x05 message is sent by TC1 until it is received by the GCS.
2. (2) — Time since GCS initiates the mTLS connection until the UxV completes the session list creation, marking the end of the connection establishment phase.
3. (3) — Time since TC1 sends the 0x05 message until the UxV completes the connection establishment protocol.

Over Wi-Fi, the 0x05 message transmission required on average 5.3 ms, while the subsequent GCS–UxV connection was established in approximately 0.26 s. The total interval from the 0x05 transmission to the completion of the UxV connection averaged 0.27 s, confirming that the setup process is nearly instantaneous when all links operate under the same high-throughput network.

When the TC1–GCS link was replaced by the HR-5000H, the 0x05 transmission delay increased to nearly 0.9 s, even though the GCS–UxV connection time remained unchanged at around 0.28 s due to its Wi-Fi interface. Consequently, the total end-to-end connection interval rose to 1.19 s. This demonstrates that the latency added by the HR-5000 hop directly propagates through the system, delaying the start of the GCS–UxV connection and extending the overall establishment time.

Also, under the HR-5000H links, the standard deviations were approximately 160 ms for the 0x05 transmission and 190 ms for the total interval. This confirms that the variability originates almost entirely from the tactical radio hop, as the local Wi-Fi segment remains highly stable. These results again highlight that even when only one hop operates under the TNW50 waveform, its low bandwidth and long round-trip time significantly affect the end-to-end responsiveness of the network.

	(1)	(2)	(3)
Scenario 1			
Mean (μs)	5340	258515.71	274555.71
Variance (μs)	2.18×10^{7}	1.25×10^{8}	1.33×10^{8}
Standard Deviation (μs)	4624.63	111982.88	115177.94
Scenario 2			
Mean (μs)	899074.29	282018.57	1198301.43
Variance (μs)	2.53×10^{9}	6.07×10^{8}	3.73×10^{9}
Standard Deviation (μs)	159018.55	77881.57	193073.28

Table 5.5: Mean, variance, and standard deviation results for: (1) TC1–GCS 0x05 message transmission, (2) GCS–UxV connection, and (3) TC1–UxV connection.

5.2.5 TC1-TC2-GCS-UxV

Table 5.6 reports four relevant intervals: (1) the transmission time of the 0x09 message from TC1 to TC2, (2) the time from TC2 sending 0x05 until it is received at the GCS, (3) the GCS–UxV connection establishment, and (4) the end-to-end interval from TC1 sending 0x09 until the UxV is ready to communicate. In this setup the GCS–UxV link is always Wi-Fi and the HR-5000 is only used on the TC1–TC2 and TC2–GCS hops.

	(1)	(2)	(3)	(4)
Scenario 1				
Mean (μs)	15841.43	3202.86	281970	327990
Variance (μs)^2	5.83×10^{9}	6.81×10^{9}	5.77×10^{11}	7.71×10^{11}
Standard Deviation (μs)	24138.64	2609.74	75990.66	87816.05
Scenario 2				
Mean (μs)	841865.71	862287.14	244751.43	1973082.86
Variance (μs)^2	6.23×10^{9}	7.53×10^{9}	7.51×10^{9}	2.22×10^{10}
Standard Deviation (μs)	78940.23	86737.94	86664.53	149131.39

Table 5.6: Mean, variance, and standard deviation results for: (1) TC1–TC2 0x09 message transmission, (2) TC2–GCS 0x05 message transmission, (3) GCS–UxV connection, and (4) TC1–UxV connection.

Over Wi-Fi, both the 0x09 and 0x05 message transmission intervals remain extremely short, while the GCS–UxV connection and the overall end-to-end interval complete within fractions of a second. The low dispersion observed across all stages reflects the deterministic behaviour and high stability of a broadband local network.

When the communication between TC1–TC2 and TC2–GCS operates over HR-5000 radios, both message forwarding intervals (1) and (2) become substantially longer, as each transmission must traverse narrowband tactical links that impose high latency. The GCS–UxV connection time (3) remains largely unaffected, since it relies on the same Wi-Fi link in both configurations. As a result, the total end-to-end connection time (4) increases considerably, with the delays introduced by the two radio hops dominating the overall establishment process.

The greater variability observed in the HR-5000 results for intervals (1), (2), and (3) are in accordance with the iperf3 tests, where the jitter was 2 orders of magnitude bigger in the radio links.

5.2.6 Discussion and Interpretation of Session Establishment Times

Across all tested architectures, the use of HR-5000H tactical radios in the communication path led to a significant increase in both the connection establishment and intermediate message transmission times when compared with Wi-Fi. This behavior was consistent from direct TC1–GCS (Table 5.2), TC1-TC2 (Table 5.3) and TC2-GCS (Table 5.6) connections to multi-hop configurations involving TC2 and GCS nodes. The observed increase and variability are directly related to the physical and link-layer limitations of the TNW50 waveform and its impact on transport-layer mechanisms, as previously quantified by the `iperf3` tests in Section 5.1.

The `iperf3` results showed that the HR-5000H operates with an average RTT close to 900 ms, jitter values reaching several hundred milliseconds, and a throughput of only a few kilobits per second. These conditions contrast sharply with the Wi-Fi network, which presented sub-10 ms RTT and tens of megabits per second of bandwidth. Such differences directly explain the delay propagation observed in the NC2S comunication.

In the case of direct connections, the certificate exchange, credentials transmission and NTP synchronization are carried out inside mTLS. The high values for connetions times can possibily be associated to the high RTT of the HR-5000H link, where each handshake message requires almost one second to traverse the channel. Thus, especially when the messages that contain the files need to be fragmented, each packet acknowledgment or fragment can be delayed beyond the RTO interval. This could be forcing the TCP stack to trigger unnecessary retransmissions, further extending the total connection establishment time. This cumulative effect was visible in the measurements, where each additional retransmission or delayed acknowledgment produced multi-second variations in setup time.

For control messages (0x05, 0x08, and 0x09), which are exchanged over UDP, similar latency and variance patterns were observed, even though UDP does not enforce acknowledgments or retransmissions. One possibility for this latency can possibily be that the radio Maximum Transmission Unit (MTU) to be low and thus the messages needed to be fragmented. Another possible cause can be that the HR-5000H radios function as subnet gateways that encapsulate IP traffic and buffer data before transmission over the TNW50 waveform, thus encreasing the transmission times.

Unfortunatelly critical information, as the radio MTU, is considered by the radio manufacteur to be proprietary information and thus not accessible to the public. Thus it is only possible to observe that the message latency is also a consern in UDP messages and point to possible causes for it.

In contrast, the Wi-Fi network provides a high-throughput, low-latency, and low-jitter medium. The same mTLS handshake and UDP-based control exchanges complete almost instantaneously, as con-

firmed by the iperf3 throughput and RTT measurements. Therefore, the large differences between the two infrastructures are not due to cryptographic or computational overhead, but rather to the physical and link-layer properties of the HR-5000H waveform and its interaction with higher-layer protocols.

In summary, the iperf3 results and the measured connection establishment times are consistent and mutually reinforcing. The high RTT, low throughput and high jitter of the HR-5000H link lead to repeated RTO events and sequential waveform scheduling delays, explaining both the longer mean connection times and the greater variability observed across all radio-based configurations.

5.3 Handover Time

The NC2S framework introduces two methods for UxV control handover, described in Section 4.5.6. This test measured the time required for the handover to occur in both methods.

In each method, the system was tested in two scenarios. In scenario 1, the nodes were connected in a full Wi-Fi infrastructure. In scenario 2, the HR-5000H radios were employed for communication between the CT1 and the GCS nodes, as illustrated in Figure 5.3. Each test was repeated seven times, using the same centralized timestamping method as described above.

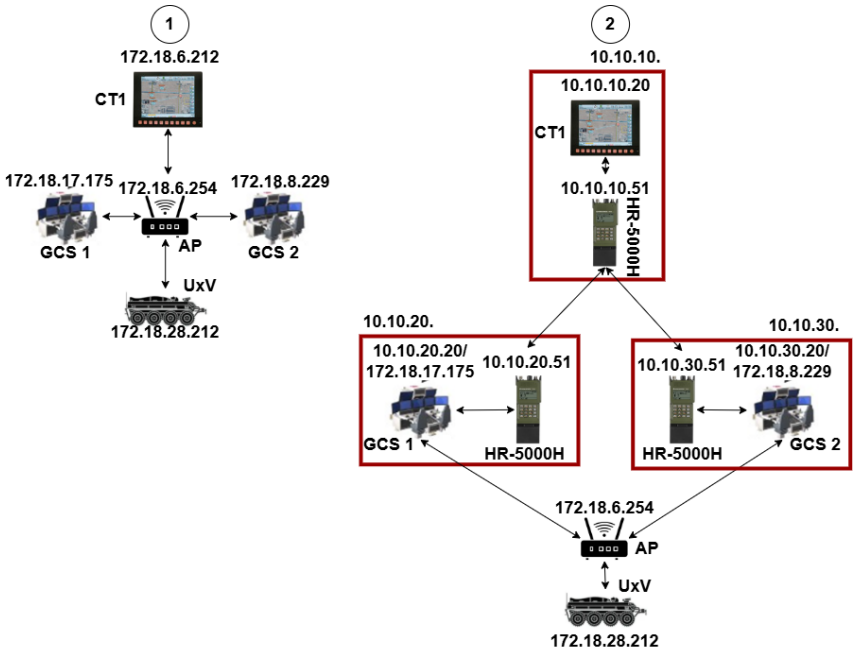

Figure 5.3: Handover Testing Architecture.

5.3.1 Handover with Credential Revocation

The handover time requires a clear definition since the operator (CT1 user) triggers the two handover phases separately, as explained in Section 4.5.6.A. First, it revokes the current GCS–UxV credential, and then establishes a new GCS–UxV session. Therefore, it is important to consider the user latency between the start of the first phase and the start of the second phase. Analyzing handover time should not only rely on system measurements but also account for the human factor, although this was not measured as it could vary from one user to another. The following equations describe how to calculate the handover bounds. The first equation determines the handover interval independent of user latency:

$$T_{\text{handover,system}} \in \left[\max(\overline{T}_{\text{rev}}, \overline{T}_{\text{conn}}),\ \overline{T}_{\text{rev}} + \overline{T}_{\text{conn}} \right],$$

where $\overline{T}_{\text{rev}}$ represents the mean value of the interval from when the CT1 sends the 0x04 message until the UxV receives it and shuts the connection with the GCS and the $\overline{T}_{\text{conn}}$ represents the total connection time from the CT1 to the UxV. The second equation includes the human/action latency Δ_{human}, hence:

$$T_{\text{handover,E2E}} \in \left[\max(\overline{T}_{\text{rev}}, \overline{T}_{\text{conn}}) + \Delta_{\text{human}},\ \overline{T}_{\text{rev}} + \overline{T}_{\text{conn}} + \Delta_{\text{human}} \right],$$

These bounds are appropriate because the two phases are independent, may partially overlap, and their durations are random variables dominated by the underlying link characteristics.

The revocation intervals are present in the Table 5.7 and the connection establishment intervals are the ones measured in Table 5.2.

	(1)	(2)	(3)	(4)	(5)
Scenario 1					
Mean (μs)	6198,57	22804,29	2997,14	9691,43	28557,14
Variance $(\mu s)^2$	1,48x10^7	1,01x10^8	1,79x10^6	4,13x10^9	5,03x10^9
Standard Deviation (μs)	3846,07	10043,62	1339,15	2032,28	7093,23
Scenario 2					
Mean (μs)	604467,14	616130	4758,57	10584,29	623820
Variance $(\mu s)^2$	6,00x10^{11}	6,10x10^{11}	8,30x10^8	1,86x10^{10}	6,23x10^{11}
Standard Deviation (μs)	77464,07	78111,59	9112,34	4308,61	78936,40

Table 5.7: Mean, variance, and standard deviation results for: (1) TC1–GCS 0x04 message, (2) TC1–GCS revocation, (3) GCS–UxV 0x04 message transmission, (4) GCS–UxV revocation, and (5) TC1–UxV revocation.

The 5 intervals in the table represent:

- (1) — Time since the 0x04 message is sent by TC1 until it is received by the GCS.
- (2) — Time since the 0x04 message is sent by TC1 until the GCS process the CredRL and shutdown the connetion to the UxV.
- (3) — Time since the 0x04 message is sent by GCS until it is received by the UxV.

- (4) — Time since the 0x04 message is sent by GCS until the UxV process the CredRL and shutdown the connetion to the GCS.

- (5) — Time since the 0x04 message is sent by TC1 until the UxV process the CredRL and shutdown the connetion to the GCS.

Under Wi-Fi, all intervals (1)–(5) are short with low dispersion, matching the low RTT/jitter profile observed in the `iperf3` characterization. With HR-5000H in the path TC1↔GCS, intervals (1), (2), and (5) become significantly larger and more variable, while (3)–(4) remain small because GCS–UxV uses Wi-Fi in both scenarios. This pattern isolates the tactical radio hop as the dominant contributor to revocation latency and variance.

	$T_{\text{handover,system}}$ (μs)	$T_{\text{handover,E2E}}$ (μs)
Scenario 1	[274555,71; 304012,86]	[274555,71; 304012,86] $+\Delta_{\text{human}}$
Scenario 2	[1198301,43; 1822121,43]	[1198301,43; 1822121,43] $+\Delta_{\text{human}}$

Table 5.8: Handover-with-revocation time bounds derived from measured revocation and connection intervals. The end-to-end bound adds the operator/action latency Δ_{human}.

After computing the bounds it is visible that relative to Wi-Fi, HR-5000H increases the handover time by roughly 4–6 times, with dispersion dominated by UDP propagation over TNW50. The additional end-to-end delay is operator dependent via Δ_{human}.

5.3.2 Handover With Capacity String Modification

The following tables show the handover intervals and messages transmission times for the credential forwarding to both GCSs. The 5 intervals in both tables represent:

- (1) — Time since the 0x13 message is sent by TC1 until it is received by the GCS.

- (2) — Time since the 0x13 message is sent by TC1 until the GCS process and substitutes the credential for the connection to the UxV.

- (3) — Time since the 0x15 message is sent by GCS until it is received by the UxV.

- (4) — Time since the 0x15 message is sent by GCS until the UxV process and substitutes the credential for the connection to theGCS.

- (5) — Time since the 0x13 message is sent by TC1 until the UxV process and substitutes the credential for the connection to theGCS.

	(1)	(2)	(3)	(4)	(5)
Scenario 1					
Mean (μs)	3554,29	30031,43	4270	23760	37495,71
Variance $(\mu s)^2$	7,75x10^9	2,03x10^8	6,89x10^6	3,76x10^9	5,18x10^9
Standard Deviation (μs)	2792,04	14231,26	2587,34	19404,99	22765,21
Scenario 2					
Mean (μs)	851078,57	877842,86	2448,57	13435,71	874175,71
Variance $(\mu s)^2$	6,01x10^{11}	5,89x10^{11}	2,61x10^8	2,97x10^{10}	5,91x10^{11}
Standard Deviation (μs)	78042,74	76798,06	510,54	5474,26	76930,37

Table 5.9: Mean, variance, and standard deviation results for the credential renewal process: (1) TC1–GCS1 0x13 message transmission, (2) TC1–GCS1 credential processing, (3) GCS1–UxV 0x15 message transmission, (4) GCS1–UxV credential processing, and (5) TC1–UxV credential processing.

	(1)	(2)	(3)	(4)	(5)
Scenario 1					
Mean (μs)	9094,29	56894,29	3695,71	16431,43	49438,57
Variance $(\mu s)^2$	2,46x10^7	1,87x10^8	1,10x10^7	2,65x10^8	1,74x10^9
Standard Deviation (μs)	15689,76	43295,25	3317,62	16004,02	41733,79
Scenario 2					
Mean (μs)	918502,86	975558,57	4845,71	15198,57	970468,57
Variance $(\mu s)^2$	2,88x10^{11}	3,19x10^{11}	1,86x10^8	1,64x10^{10}	2,59x10^{11}
Standard Deviation (μs)	168895,20	178370,12	4319,43	12811,41	161086,76

Table 5.10: Mean, variance, and standard deviation results for the credential renewal process: (1) TC1–GCS2 0x13 message transmission, (2) TC1–GCS2 credential processing, (3) GCS2–UxV 0x15 message transmission, (4) GCS2–UxV credential processing, and (5) TC1–UxV credential processing.

Similarly to the handover with the credential revocation the handover with credential update also introduces the human latency factor.

In order to compute the handover bounds also two formulas were also taken into account. The first, where only the system latency is taken into account:

$$T_{\text{handover,cap}} \in \left[\max(\overline{T}_{\text{TC1-GCS1-UxV}}, \overline{T}_{\text{TC1-GCS2-UxV}}) \; ; \; \overline{T}_{\text{TC1-GCS1-UxV}} + \overline{T}_{\text{TC1-GCS2-UxV}} \right],$$

where the $\overline{T}_{\text{TC1-GCS1-UxV}}$ represents the interval from the CT1 sending the 0x13 message and the UxV finishing processing, then new credential with the capacity string that retrieves the GCS1 control, and the $\overline{T}_{\text{TC1-GCS1-UxV}}$ represents the opposite, that is, the interval from the CT1 sending the 0x13 and the UxV updating of the credential for the GCS2 connection given the control to it.

The second equation includes the human/action latency Δ_{human} (in the case that the two rounds are triggered sequentially), hence:

$$T_{\text{handover,E2E}} \in\in \left[\max(\overline{T}_{\text{TC1-GCS1-UxV}}, \overline{T}_{\text{TC1-GCS2-UxV}}) \; + \Delta_{\text{human}}; \; \overline{T}_{\text{TC1-GCS1-UxV}} + \overline{T}_{\text{TC1-GCS2-UxV}} + \Delta_{\text{human}} \right]$$

	$T_{\text{handover,cap}}$ (μs)	$T_{\text{handover,cap}}$ + Δ_{human} (μs)
Wi-Fi	[49438,57; 86934,28]	[**49438,57**; **86934,28**] $+\Delta_{\text{human}}$
HR-5000	[970468,57; 1844644,28]	[970468,57; 1844644,28] $+\Delta_{\text{human}}$

Table 5.11: Capacity-modification handover bounds using $T_{\text{GCS1}} = (5)$ from Table 5.9 and $T_{\text{GCS2}} = (5)$ from Table 5.10.

Compared to Wi-Fi, HR-5000H increases the capacity-modification handover bounds by $\approx 20\times$. The increase is driven almost entirely by the TC1↔GCS radio segments (stages (1), (2)), whereas the GCS–UxV steps (3), (4) remain stable because they run on Wi-Fi. Standard deviation values over radio reflects TNW50 Robust Mode condition on UDP.

5.4 Key Renewal

This test evaluated the time required for two nodes to renew their session key pairs. Again, the system was tested in 2 scenarios illustrated in Figure 5.2. In Scenario 1, the TC1-GCS link was tested under Wi-Fi (labeled as 1) and in scenario 2 this link was tested under HR-5000H (labeled as 8). In both scenarios, the GCS-UxV link was tested only with Wi-Fi. Each experiment was repeated seven times using the same Wi-Fi server timestamping method described previously.

The results of the measurements are present in Table 5.12, TC1–GCS link, and Table 5.13 for the GCS–UxV link. On both tables the 4 intervals represent:

- (1) — Time since the 0x11 message is sent by the client node until it is received by the server node.
- (2) — Time since the 0x11 response message is sent by the server node until it is received by the client node.
- (3) — Time since the initial 0x11 message is sent by the client node until the server node, after receiving the message authenticated with the new keys, substitutes the old keys with the new keys.
- (4) — Time since the initial 0x11 message is sent by the client node until this node, after receiving the 0x11 response message from the server node, derives and substitutes the old keys with the new keys.

As described in Section 4.5.7, the renewal follows a bidirectional exchange of 0x11 messages over UDP. The server node only replaces the old keys once it successfully verifies a message HMAC computed with the new keys, meaning that the measured completion time at the server side depends not only on propagation delay but also on how quickly the client sends a subsequent message containing a valid HMAC generated with the updated keys.

5.4.1 TC1-GCS Key Renewal Performance

	(1)	(2)	(3)	(4)
Scenario 1				
Mean (μs)	8062	8170	64228	37742
Variance $(\mu s)^2$	4,58x10^6	5,39x10^6	4,00x10^7	1,24x10^8
Standard Deviation (μs)	6771,25	7342,59	20000,33	35197,90
Scenario 2				
Mean (μs)	405298	446754	1492390	855576
Variance $(\mu s)^2$	5,98x10^8	2,17x10^9	1,55x10^{11}	3,43x10^9
Standard Deviation (μs)	24459,58	46562,06	39399,86	58539,64

Table 5.12: Mean, variance, and standard deviation results for the key renewal process: (1) TC1–GCS 0x11 message transmission, (2) GCS–TC1 0x11 message transmission, (3) key renewal time measured at the GCS, and (4) key renewal time measured at the TC1.

In Wi-Fi, the 0x11 message transmission times in both directions are similar, being 8.06 ms from TC1 to GCS and 8.17 ms in the reverse direction, showing a highly symmetric and stable link. The total renewal, measured at the GCS (column 3) and at TC1 (column 4), averages 64.2 ms and 37.7 ms, respectively. The difference between these two values reflects the waiting time until the GCS receives a packet authenticated with the new keys. All standard deviations remain below 35 ms, confirming that Wi-Fi latency and jitter have negligible influence.

When the same process occurs over HR-5000H radios, all values increase by roughly two orders of magnitude. The mean 0x11 transmission times rise to 405.3 ms (client–server) and 446.8 ms (server–client), while the renewal measured at the GCS reaches 1.49 s and at the TC1 about 0.86 s. The asymmetry between columns 3 and 4 is justified again by the time the server node (GCS) waits until the next client packet with a valid HMAC computed with the new keys arrives. Standard deviations above 40–60 ms also are in accord with the jitter measurments in the iperf3 tests and with the possible justifications explained in the discussion of the connection times for the UDP transmited messages.

5.4.2 GCS-UxV Key Renewal Performance

Under the full Wi-Fi setup, both message transmission directions are again nearly symmetric—4.79 ms and 3.22 ms—showing consistent performance. The renewal measured at the UxV (column 3) averages was 286.8 ms, significantly higher than the 10.0 ms recorded at the GCS (column 4). This difference, as explained before, is due tue the protocol design, where the GCS performs the key update immediately upon receiving the server's 0x11 message, while the UxV only completes the renewal once it receives the first message with the HMAC computed under the new key set from the GCS, which depends on the client's message rate.

When the GCS operates through HR-5000H to TC1, the 0x11 message delays remain in the same order of magnitude (3.15 ms and 10.46 ms), as expected, since the GCS–UxV link still runs over Wi-

Fi. Similarly, the renewal completion times—319.3 ms at the UxV and 17.35 ms at the GCS—remain comparable to the fully Wi-Fi setup. The higher standard deviation observed at the UxV (column 3) reflects the unpredictable delay until reception of the first packet authenticated with the new keys.

	(1)	(2)	(3)	(4)
Scenario 1				
Mean (μs)	4790	3218	286794	10046
Variance $(\mu s)^2$	2,56x10^7	1,09x10^6	9,63x10^{10}	2,68x10^7
Standard Deviation (μs)	5060,33	1045,60	98151,38	5177,69
Scenario 2				
Mean (μs)	3154	10460	319348	17350
Variance $(\mu s)^2$	4,42x10^6	9,27x10^6	2,12x10^{11}	7,52x10^7
Standard Deviation (μs)	2102,06	9629,94	145686,23	8674,52

Table 5.13: Mean, variance, and standard deviation results for the key renewal process between GCS and UxV: (1) GCS–UxV 0x11 message transmission, (2) UxV–GCS 0x11 message transmission, (3) key renewal time measured at the UxV, and (4) key renewal time measured at the GCS.

In conclusion, across both renewal scenarios, the following trends are evident:

- Transmission delay symmetry - The 0x11 message transmission times ((1) and (2)) are symmetric and low in Wi-Fi, while they increase by roughly two orders of magnitude when transmitted over HR-5000, consistent with the $\sim$900 ms round-trip delay of the TNW50 waveform.
- Asymmetric renewal completion - Renewal times measured on the server side ((3) in both tables) are systematically longer than those on the client side ((4)), as they depend on the arrival of the next valid HMAC authenticated packet.
- Impact of message rate - In the fully Wi-Fi setup, the GCS–UxV renewal (286 ms) takes longer than the TC1–GCS renewal (64 ms), primarily because TC1 transmits data to the GCS at a higher rate than the GCS to the UxV, leading to faster delivery of the post-renewal packet that confirms the update.

5.5 System Reliability, Goodput Estimation, and CPU Processing Time

This test aimed to evaluate communication reliability, estimate the goodput for the NC2S communications, and analyze CPU processing time. The procedure consisted of establishing a connection between TC1 and the GCS, maintaining the communication for 30 seconds, and then ordering the GCS to connect to the UxV. The communication was kept active for two more minutes, after which the scripts were terminated. For each run, the number of messages sent and received, the number of bytes transmitted and received per connection, and the average processing time per message were recorded.

However, this study was subject to several limitations. The HR-5000H manufacturer does not provide the over-the-air datagram structure, making it impossible to determine the complete frame overhead and hence the real physical throughput. Consequently, only the NC2S application-layer goodput could be estimated.

Additionally, given that the TNW50 waveform offers a very limited user data rate (10 kb/s in Robust Mode), it was necessary to reduce the amount of control traffic between command nodes to prevent communication failure.

In operational terms, not all MAVLink 2 telemetry messages sent by the UxV to the GCS are required by TC1, whose role corresponds to a mission commander rather than the direct vehicle operator. Therefore, the GCS script was configured to retransmit only the *HEARTBEAT* and *GPS* messages to TC1 every ten seconds. This way, it was expected that these modifications would effectively reduce the TC1–GCS data rate to a level compatible with the tactical radio bandwidth while operational viability could be maintained.

Each experiment was executed five times to ensure consistency and reproducibility. The testing architectures were divided into 2 scenarios. Scenario 1 represents a full Wi-Fi architecture and is present in the Figure 5.2 labeled as 1. In scenario 2, the HR-5000H radios were introduced in the TC1-GCS link (Figure 5.2 labeled as 8).

5.5.1 System Reliability

Message reliability directly affects the robustness of the NC2S protocol, as all control and key management operations rely on the correct delivery of UDP messages. Table 5.14 summarizes the mean, variance, and standard deviation of messages sent, messages lost, and corresponding message loss percentages for each communication direction under both Wi-Fi and HR-5000H links.

Across all Wi-Fi tests, message delivery was nearly lossless. For the TC1–GCS and GCS–TC1 channels, packet loss remained below 0.2%, with most runs showing no losses at all. The GCS–UxV direction exhibited a similar trend with 0% loss, while the high-rate UxV–GCS traffic, responsible for telemetry and state updates, recorded a small mean packet loss of only 0.37%. Even this value is statistically insignificant for UDP operation in a wireless LAN. The low variances and standard deviations across all links confirm a stable and predictable communication channel. These results are consistent with the `iperf3` findings, where throughput exceeded 50 Mb/s and jitter remained below 1.2 ms.

When using HR-5000H tactical radios, message loss slightly increases but remains within acceptable limits. The TC1–GCS link shows an average packet loss of 1.29%, while all other directions stay at or below 0.31%. This increase is modest given the waveform's narrow bandwidth (5 kb/s) and high latency (900 ms RTT). The fact that the GCS–TC1 link maintain zero losses demonstrates the efficiency of the TNW50's link-layer error correction and the protocol resilience to temporary propagation delays.

Variability remains low (standard deviations under 1%), indicating consistent behaviour across all runs.

	TC1–GCS	GCS–TC1	GCS–UxV	UxV–GCS
Scenario 1 – Messages Sent				
Mean	153,2	56,4	212,6	7792,8
Variance	0,7	1,7	126,3	7666,7
Standard Deviation	0,84	1,30	11,28	87,56
Scenario 1 – Messages Lost				
Mean	0,2	0	0	32
Variance	1,41	0	0	193,9
Standard Deviation	1,19	0	0	14,11
Scenario 1 – Messages Lost (%)				
Mean	0,13	0	0	0,37
Variance	0,08	0	0	0,25
Standard Deviation	0,29	0	0	0,50
Scenario 2 – Messages Sent				
Mean	154,8	56,2	191,4	7457,2
Variance	0,7	1,7	126,3	7666,7
Standard Deviation	0,84	1,30	11,28	87,56
Scenario 2 – Messages Lost				
Mean	2	0	0	23,6
Variance	1,41	0	0	193,9
Standard Deviation	1,19	0	0	14,11
Scenario 2 – Messages Lost (%)				
Mean	1,29	0	0	0,31
Variance	0,82	0	0	0,35
Standard Deviation	0,91	0	0	0,19

Table 5.14: Mean, variance, and standard deviation results for messages sent, messages lost, and packet loss percentage across links TC1–GCS, GCS–TC1, GCS–UxV, and UxV–GCS, comparing WiFi and HR-5000H.

Overall, both infrastructures exhibit excellent reliability. Under Wi-Fi, packet loss is practically negligible, ensuring near-perfect message delivery for all control exchanges. Under HR-5000H, the slight increase in loss percentage does not compromise protocol correctness, as NC2S design tolerates minor UDP losses without impacting the connection or key management sequences. The absence of retransmission mechanisms at the transport layer (UDP) avoids further delay accumulation, while the small absolute loss values confirm that the tactical radio system maintains reliable message delivery despite operating at two orders of magnitude lower throughput than Wi-Fi.

5.5.2 Goodput Estimation

This test aimed to evaluate the effective data rate of the NC2S communications in order to verify whether the HR-5000H radios could, in theory and despite their latency and bandwidth constraints, support the command and control traffic between the GCS and the UxV.

	TC1–GCS	GCS–TC1	GCS–UxV	UxV–GCS
Scenario 1				
Mean (bit/s)	245,45	135,67	671,45	32232,88
Variance	14,04	0,76	17859,48	8918682,13
Standard Deviation	3,75	0,87	133,64	2986,42
Scenario 2				
Mean (bit/s)	247,83	134,63	611,80	30860,57
Variance	1,62	11,53	1164,31	113062,73
Standard Deviation	1,27	3,39	34,12	336,25

Table 5.15: Goodput estimation results (in bit/s) for each communication direction over WiFi and HR-5000H links.

The results, present in Table 5.15, confirm that the NC2S, under the defined message filtering policy, generates extremely low goodput requirements at the command-and-control level. The TC1–GCS and GCS–TC1 links exhibited a maximum mean of only 247,83 bit/s, which is orders of magnitude below the HR-5000H radio throughput measured in the `iperf3` tests (approximately 5 kb/s). This demonstrates that the HR-5000H can reliably support the exchange of command and telemetry messages between command nodes without saturating the channel.

In contrast, the data rate between the GCS and the UxV increased by nearly two orders of magnitude, reaching approximately 32 kb/s in the UxV–GCS direction. This value largely exceeds the 5 kb/s limit observed during the `iperf3` radio benchmark, confirming that the HR-5000H cannot sustain continuous vehicle control or telemetry transmission. Therefore, while suitable for secure coordination and control delegation among commanders, the radio is not appropriate for real-time UxV operation, which requires a higher data rate and lower latency.

The low variance observed in both the Wi-Fi and HR-5000H tests indicates that the NC2S communication behavior remains stable and consistent across repeated transmissions, reflecting a predictable system performance independent of the underlying link technology.

5.5.3 CPU Processing Time

Table 5.16 shows that local processing delays remain low and stable across all nodes. Under Wi-Fi, the mean times were 2.62 ms for TC1, 2.69 ms for GCS, and 1.02 ms for the UxV, with sub-millisecond dispersion. These values confirm that NC2S cryptographic and message-handling operations impose minimal computational load.

With HR-5000H, processing times remain nearly identical—2.48 ms, 2.59 ms, and 1.16 ms respectively—showing that the communication medium has negligible effect on CPU workload. Slightly higher variance on TC1 reflects sporadic packet bursts caused by the radio's higher latency, but all values stay below 3 ms.

Overall, node processing contributes less than 0.1% to total protocol latency. Thus, the observed

differences between Wi-Fi and HR-5000H stem almost entirely from transmission and waveform delays rather than computation.

	TC1	GCS	UxV
Scenario 1			
Mean (μs)	2623	2693.20	1023.89
Variance (μs^2)	1352	83039.76	165379.81
Standard Deviation (μs)	36.77	288.27	407.41
Scenario 2			
Mean (μs)	2480.40	2594	1161.05
Variance (μs^2)	20519.30	2542.50	77597.84
Standard Deviation (μs)	143.25	50.42	278.56

Table 5.16: Mean, variance, and standard deviation of the average processing time (in μs) for each node (TC1, GCS, and UxV) using WiFi and HR-5000H communication.

The variance values shown in Table 5.16 remain small relative to the mean, confirming that NC2S cryptographic and verification routines exhibit stable and predictable computational performance. The slightly higher dispersion observed on TC1 under HR-5000H can be attributed to sporadic packet bursts caused by the radio's higher transmission latency, which temporarily increases the message queue depth and leads to brief CPU utilization peaks.

5.6 Summary

This chapter experimentally evaluated NC2S over two contrasting setups. In one hand, a high-throughput, low-latency Wi-Fi network and on the other hand an HR-5000H with TNW50 waveform in Robust Mode tactical radio setup.

First of all, baseline `iperf3`/`ping` tests (Table 5.1) established the baseline of comparison. Here the Wi-Fi delivered tens of Mb/s and single-digit millisecond RTT, whereas HR-5000H sustained only a few kb/s with ≈0.9 s RTT and high jitter.

Session establishment experiments across all topologies showed that any path that includes a HR-5000H link is dominated by the radio latency, increasing both mean delay and dispersion. For example, in the direct TC1–GCS case (Table 5.2), the mean setup increased from $298185\ \mu s$ (Wi-Fi) to $16137258\ \mu s$ (HR-5000), a $\sim \mathbf{54}\times$ rise. In the TC1–TC2–GCS multi-hop case (Table 5.4), total time grew from 0.57 s to 18.77 s ($\sim \mathbf{33}\times$). In the TC1-GCS-UxV setup the 0x05 transmission grew by $\sim \mathbf{168}\times$ (5.34 ms $\rightarrow$ 0.90 s), while the GCS–UxV step (still on Wi-Fi) remained essentially unchanged. Finally the end-to-end TC1–UxV interval rose by $\sim \mathbf{4.4}\times$ (0.275 s $\rightarrow$ 1.198 s).

Handover experiments quantified system-only bounds and operator-influenced bounds. With credential revocation (Tables 5.7–5.8), HR-5000H increased the system bounds by $\sim \mathbf{4.4}\times$ to $\mathbf{6.0}\times$ relative

to Wi-Fi. With capacity-string modification (Tables 5.9, 5.10, 5.11), the bounds were $\sim \mathbf{20}\times$ larger when the TC1↔GCS path used HR-5000H, while the GCS–UxV steps stayed stable on Wi-Fi.

Regarding the key renewal (Tables 5.12–5.13), in the TC1–GCS link, the 0x11 one-way message times increased by $\sim \mathbf{50}\times$ (8.06 ms $\rightarrow$ 405 ms), and the server-side completion time rose by $\sim \mathbf{23}\times$ (64.2 ms $\rightarrow$ 1.49 s), driven by the wait for the first post-renewal message authenticated under the new keys. In GCS–UxV, where the link stayed on Wi-Fi, timings remained in the same order of magnitude even when the upstream GCS had a radio backhaul, verifying the impact limited scope of this radio backhaul.

Reliability was consistently high (Table 5.14) on both setups. Goodput estimation (Table 5.15) showed that command-level exchanges (TC1↔GCS) require only ≈200–250 bit/s under the applied filtering policy, which is well within HR-5000H capability, whereas UxV→GCS telemetry approaches ∼32 kb/s, exceeding the ∼5 kb/s sustained by the radio benchmarks. CPU processing times (Table 5.16) stayed in the low millisecond range and contributed negligibly to end-to-end latency, confirming that performance is network-bound rather than compute-bound.

Limitations include the lack of HR-5000H over-the-air datagram frame format/MTU details by the manufacturer, thus restricting throughput analysis to the application layer. Overall, the results support a deployment strategy where HR-5000H is used for secure command/coordination and control-delegation traffic, while UxV control/telemetry remains on higher-bandwidth, lower-latency Wi-Fi links. Also, it was proved that message filtering/pacing policies are effective to keep command traffic radio-compatible.

6

Conclusion

Contents

6.1 Conclusions

This thesis was developed to obtain a Master's Degree in Military Electrical Engineering and was integrated into the Portuguese Army's EXE03 UGV-M113 (Robotisation of the M113) project, which aims to modernize and automate the M113 armoured vehicles, enhancing their effectiveness and interoperability in future digitized battlefield scenarios.

The main objective of this research was to design and implement a lightweight and secure command, control, and communication system capable of ensuring CIA while allowing flexible control delegation between GCS and subordinate nodes. The system, named NC2S, was designed to operate efficiently in constrained military environments and support both commercial and tactical communication infrastructures.

Through the analysis of existing approaches proposed by other authors, it became clear that there are multiple strategies to address the challenges of secure communication in unmanned systems. However, the comparative study revealed that the major difficulty lies in reconciling strong security properties with the lightweight operation required for deployment on resource-constrained UxVs. Many existing solutions either impose excessive computational overhead or lack sufficient resilience against credential compromise, highlighting the need for a balanced and modular approach.

The NC2S framework was therefore designed to achieve this balance. It introduced a layered authentication mechanism combining digital certificates and dynamic credentials, supported by a mission policy that precisely defines each node's access privileges, accepted message types, and allowed control levels.

Confidentiality, although not yet natively guaranteed by the protocol itself, is ensured by the military radios to be employed in the command links. In particular, the HR-5000H radio integrate AES-256 encryption (COMSEC) and frequency hopping (TRANSEC) to protect the transmitted data. The only exception lies in the GCS–UxV link, where, due to higher data rates and bandwidth requirements, communication must rely on alternative wideband technologies such as private cellular or Wi-Fi networks. In those cases, the NC2S protocol can be extended to include an additional asymmetric key pair dedicated to end-to-end encryption, thus guaranteeing confidentiality independently of the underlying medium.

To validate the system, a series of controlled laboratory experiments were carried out comparing Wi-Fi and HR-5000H tactical radio setups. The tests included connection establishment time, control handover, key renewal performance, message loss percentage, throughput estimation, and CPU processing time. Each metric was measured over multiple repetitions to ensure statistical reliability.

The obtained results confirmed the correct operation of all implemented protocols and demonstrated the expected trade-off between performance and link robustness. Over Wi-Fi, the connection establishment, credential renewal, and handover protocols were completed in the order of milliseconds to sub-seconds, offering near real-time responsiveness suitable for UxV control. In contrast, HR-5000H

links introduced latencies up to two orders of magnitude higher, with average connection times around 20 s, primarily due to the narrowband TNW50 waveform and internal link establishment mechanisms. Nevertheless, the radio configuration provided stable communication with very low packet loss, validating its suitability for command-level exchanges where reliability and security are prioritized over speed.

However, due to the proprietary nature of the HR-5000H radio, which is designed for exclusive military use, essential technical information such as the over-the-air packet format, the effective MTU, and internal buffering mechanisms are not publicly available. This lack of information prevents a precise analysis of the causes behind the measured delays, making it possible only to infer potential sources such as waveform initialization, link negotiation, or internal encryption and transmission scheduling processes. This limitation represents an inherent inconvenience when conducting performance evaluation with closed military hardware, where complete protocol specifications cannot be accessed for validation.

The tests also confirmed that the cryptographic operations and mTLS-based authentication introduced negligible computational overhead relative to the transport delays, proving that the security mechanisms are compatible with low-power embedded hardware.

Overall, the experimental campaign validated the feasibility of NC2S as a secure, flexible, and lightweight framework for UxV control and command delegation. The system successfully integrated modular security features with measurable performance characteristics, contributing to the advancement of secure C3 architectures for future networked military operations.

6.2 Future Work

Future development of the NC2S framework should focus on improving scalability, interoperability, and security to enable broader deployment in operational environments. One of the main priorities is the evolution of the current centralized trust architecture into a hierarchical and distributed certification model. In this configuration, secondary commanders (TC2) or other trusted entities would be capable of generating and signing credentials within mission-defined boundaries, maintaining accountability through a secure chain of trust. This improvement would enhance system resilience and autonomy in field operations, eliminating the single point of failure associated with the dependence on TC1 for certificate issuance.

In terms of communication infrastructures, a key step forward involves evaluating the system over private 5G mobile networks, where the GCS–UxV link could be extended to a much larger distance than the tested Wi-Fi setup.

At the same time, expanding interoperability with existing Portuguese C4I systems such as the BMS and the Tactical Assault Kit (TAK) would facilitate seamless integration of the NC2S framework into the operational battlefield environment, enhancing information sharing, situational awareness, and joint

mission coordination across command echelons.

From a security perspective, future work should focus on incorporating perfect forward secrecy and thus ensuring that the compromise of long-term keys does not affect previous sessions. Additional efforts should target the implementation of active defense mechanisms against DoS and flooding attacks, supported by lightweight anomaly detection or message-rate control. Moreover, an internal encryption layer for NC2S messages could be introduced to guarantee end-to-end confidentiality over open channels such as Wi-Fi or cellular networks. Finally, large-scale and HITL testing campaigns, involving multiple commanders and real UxV platforms integrated through MAVLink or similar control software, are essential to evaluate scalability, latency, and overall mission reliability under realistic conditions.

Bibliography

[1] A. Koubâa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, "Micro air vehicle link (mavlink) in a nutshell: A survey," *IEEE Access*, vol. 7, pp. 87 658–87 680, 2019.

[2] J. Marques, "Os veículos terrestres não tripulados no campo de batalha moderno," Master's thesis, Academia Militar, Direção de Ensino, Portugal, 2013, accessed: 2025-8-11. [Online]. Available: https://comum.rcaap.pt/entities/publication/59c8d14c-7848-42df-800d-df1997363a71

[3] B. Ferreira, "O comando e controlo no esquadrão de reconhecimento," Master's thesis, Academia Militar, Direção de Ensino, Portugal, July 2012, accessed: 2025-8-11. [Online]. Available: https://comum.rcaap.pt/entities/publication/880e7a35-7ef8-4597-98a1-4ec690572e4b

[4] A. Allouch, O. Cheikhrouhou, A. Koubâa, M. Khalgui, and T. Abbes, "Mavsec: Securing the mavlink protocol for ardupilot/px4 unmanned aerial systems," in *15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 621–628.

[5] N. A. Khan, N. Z. Jhanjhi, S. N. Brohi, and A. A. Almazroi, "A secure communication protocol for unmanned aerial vehicles," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 601–618, 2022.

[6] B. Branco, J. S. Silva, and M. Correia, "D3s: A drone security scoring system," *Information*, vol. 15, no. 12, 2024. [Online]. Available: https://www.mdpi.com/2078-2489/15/12/811

[7] N. U. I. Hossain, M. Lutfi, I. Ahmed, A. Akundi, and D. Cobb, "Modeling and analysis of unmanned aerial vehicle system leveraging systems modeling language (sysml)," *Systems*, vol. 10, no. 6, 2022. [Online]. Available: https://www.mdpi.com/2079-8954/10/6/264

[8] E. Britannica, "Unmanned aerial vehicles (uavs)," 2025. [Online]. Available: https://www.britannica.com/technology/military-aircraft/Unmanned-aerial-vehicles-UAVs

[9] Reuters, "Ukrainian brigade pioneers remote-controlled ground assaults," 2025. [Online]. Available: https://www.reuters.com/world/europe/ukrainian-brigade-pioneers-remote-controlled-ground-assaults-2025-01-16

[10] E. D. A. (EDA), “Interact - interoperability standards for unmanned systems in defence applications,” 2022. [Online]. Available: https://www.interact-padr.eu/wp-content/uploads/2022/10/INTERACT-Factsheet_Final.pdf

[11] ——, “Preparatory action on defence research (padr),” 2022. [Online]. Available: https://eda.europa.eu

[12] IEEE Computer Society, “IEEE Standard for Information Technology – Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” https://doi.org/10.1109/IEEESTD.2021.9363693, IEEE, 2021, iEEE Std 802.11-2021.

[13] CubePilot, “Herelink overview,” 2024, accessed: December 28, 2024. [Online]. Available: https://docs.cubepilot.org/user-guides/herelink/herelink-overview

[14] S. Technology, “Hm30 user manual v1.3,” 2024, accessed: December 28, 2024. [Online]. Available: https://www.siyi.biz/siyi_file/HM30/HM30%20User%20Manual%20v1.3.pdf

[15] Holybro, “Sik telemetry radio v3,” 2024, accessed: December 28, 2024. [Online]. Available: https://docs.holybro.com/radio/sik-telemetry-radio-v3

[16] MAVLink Developers. (2024) Mavlink – micro air vehicle communication protocol. Official protocol documentation. [Online]. Available: https://mavlink.io/en/

[17] MDPI Sensors, “A survey on unmanned underwater vehicles: Challenges, enabling technologies, and future directions,” *Sensors*, vol. 23, no. 17, p. 7321, 2023.

[18] Joint Air Power Competence Centre. (2021, January) A comprehensive approach to countering unmanned aircraft systems. [Online]. Available: https://www.japcc.org/books/a-comprehensive-approach-to-countering-unmanned-aircraft-systems/

[19] H. Fereidouni, O. Fadeitcheva, and M. Zalai. (2023) Iot and man-in-the-middle attacks. [Online]. Available: https://arxiv.org/abs/2308.02479

[20] H. A. Noman and O. M. F. Abu-Sharkh, “Code injection attacks in wireless-based internet of things (iot): A comprehensive review and practical implementations,” *Sensors*, vol. 23, no. 13, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/13/6067

[21] B. J. do Campo Branco, “Drone security scoring system,” Master’s thesis, Academia Militar, Lisbon, Portugal, December 2024, supervisor(s): Prof. José Silvestre Serra Silva, Prof. Miguel Nuno Dias Alves Pupo Correia.

[22] Y.-M. Kwon, J. Yu, B.-M. Cho, Y. Eun, and K.-J. Park, “Empirical analysis of MAVLink protocol vulnerability for attacking unmanned aerial vehicles,” *IEEE Access*, vol. 6, pp. 43 203–43 212, August 2018.

[23] T. Hussain, R. M. Mehmood, A. ul Haq, K. A. Alnafjan, and A. S. Alghamdi, “Designing framework for the interoperability of c4i systems,” *IEEE*, pp. 586–590, 2014.

[24] V. Sequeira, “Sistema de comando e controlo bms - battlefield management system,” *Revista Atoleiros*, no. 34, pp. 54–57, April 2020, accessed: December 16, 2024. [Online]. Available: https://www.researchgate.net/publication/341576980_Sistema_de_Comando_e_Controlo_BMS_-_Battlefield_Management_System

[25] C. F. V. Gomes, “Realidade aumentada aplicada ao sistema de combate do soldado,” Dissertação para obtenção do Grau de Mestre em Engenharia Eletrotécnica e de Computadores, Academia Militar, Lisboa, Novembro 2022.

[26] Nuno, “Joint dismounted soldier system,” pp. 2–4, 2024.

[27] R. F. C. Rodrigues, “Caraterização eletromagnética da antena laminar do rádio p/prc-525,” MSc Thesis, Universidade de Lisboa, Lisboa, Portugal, 2012.

[28] Rohde and Schwarz, “Soveron® hr handheld tactical radio - product flyer,” 2020, accessed: 2025-08-18. [Online]. Available: https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_common_library/dl_brochures_and_datasheets/pdf_1/SOVERON-HR_fly_en_5215-8170-32_v0300.pdf

[29] A. Allouch, O. Cheikhrouhou, A. Koubâa, M. Khalgui, and T. Abbes, “Mavsec: Securing the mavlink protocol for ardupilot/px4 unmanned aerial systems,” in *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, 2019, pp. 621–628.

[30] B. Tufekci, A. Arslan, C. Tunc, and K. Morozov, “Enhancing the security of the mavlink with symmetric authenticated encryption for drones,” in *2024 11th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*. IEEE, 2024, pp. 58–65. [Online]. Available: https://doi.org/10.1109/IOTSMS62296.2024.10710297

[31] Y. Li and C. Pu, “Lightweight digital signature solution to defend micro aerial vehicles against man-in-the-middle attack,” in *2020 IEEE 23rd International Conference on Computational Science and Engineering (CSE)*, 2020, pp. 92–97.

[32] H. Ismael and Z. Al-Ta’i, “Authentication and encryption drone communication by using hight lightweight algorithm,” *Turkish Journal of Computer and Mathematics Education*, vol. 12, pp. 5891–5908, 01 2021. [Online].

Available: https://www.researchgate.net/publication/392193717_Authentication_and_Encryption_Drone_Communication_by_Using_HIGHT_Lightweight_Algorithm

[33] I. F. Hashmi and E. Munir, "Securing the skies: Enhancing communication security for unmanned aerial vehicles," in *2023 17th International Conference on Open Source Systems and Technologies (ICOSST)*. IEEE, 2023, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICOSST60641.2023.10414239

[34] M. Bae and H. Kim, "Authentication and delegation for operating a multi-drone system," *Sensors*, vol. 19, no. 9, p. 2066, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/9/2066

[35] K. Wen, S. Wang, Y. Wu, J. Wang, L. Han, and Q. Xie, "A secure authentication protocol supporting efficient handover for uav," *Mathematics*, vol. 12, no. 5, p. 716, 2024, open Access. [Online]. Available: https://doi.org/10.3390/math12050716

[36] D. Kwon, S. Son, Y. Park, H. Kim, Y. Park, S. Lee, and Y. Jeon, "Design of secure handover authentication scheme for urban air mobility environments," *IEEE Access*, vol. 10, pp. 42 529–42 541, 2022.

[37] L. Li, X. Lian, Y. Wang, and L. Tan, "Csecmas: An efficient and secure certificate signing based elliptic curve multiple authentication scheme for drone communication networks," *Applied Sciences*, vol. 12, p. 9203, 2022. [Online]. Available: https://doi.org/10.3390/app12189203

[38] H. Khalid, S. J. Hashim, F. Hashim, S. M. S. Ahamed, M. A. Chaudhary, H. H. M. Altarturi, and M. Saadoon, "Hoopoe: High performance and efficient anonymous handover authentication protocol for flying out of zone uavs," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 8, pp. 10 906–10 920, 2023.

[39] B. Semal, K. Markantonakis, and R. N. Akram, "A certificateless group authenticated key agreement protocol for secure communication in untrusted uav networks," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, 2018, pp. 1–8.

[40] Y. Ko, J. Kim, D. G. Duguma, P. V. Astillo, I. You, and G. Pau, "Drone secure communication protocol for future sensitive applications in military zone," *Sensors*, vol. 21, no. 6, p. 2057, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/6/2057

[41] D. Pirker, T. Fischer, H. Witschnig, and C. Steger, "Trust-provisioning infrastructure for a global and secured uav authentication system," in *2020 International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom)*, 2020, pp. 1–6.

[42] S. A. Ayati and H. R. Naji, "A secure mechanism to protect uav communications," in *2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, 2022, pp. 1–6.

[43] G. Wang, K. Lim, B.-S. Lee, and J. Y. Ahn, “Handover key management in an lte-based unmanned aerial vehicle control network,” in *2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, 2017, pp. 200–205.

[44] H. Choe and D. Kang, “Ecc-based authentication protocol for military internet of drone (iod): A holistic security framework,” *IEEE Access*, vol. 13, pp. 21 503–21 519, 2025.

[45] S. Rajasoundaran, V. N. Santhosh Kumar, S. M. Selvi, and A. Kannan, “Reactive handover coordination system with regenerative blockchain principles for swarm unmanned aerial vehicles,” Research Square, Preprint (Version 1) rs.3.rs-3132087/v1, Jul. 2023. [Online]. Available: https://doi.org/10.21203/rs.3.rs-3132087/v1

[46] S. Son, J. Lee, Y. Park, Y. Park, and A. K. Das, “Design of blockchain-based lightweight v2i handover authentication protocol for vanet,” *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1346–1358, 2022.

[47] E. V. Filho, F. Gomes, S. Monteiro, R. Severino, S. Penna, A. Koubaa, and E. Tovar, “A drone secure handover architecture validated in a software in the loop environment,” *Journal of Physics: Conference Series*, vol. 2526, no. 1, p. 012083, jun 2023. [Online]. Available: https://doi.org/10.1088/1742-6596/2526/1/012083

[48] L. Fern, M. Draper, T. Oron-Gilad, R. J. Shively, T. Porat, M. Rottem-Hovev, and J. Silbiger, “Multi-operator multi-uav (momu) control: Exploring the influence of sensor tools and playbook task delegation,” NASA Ames Research Center, Moffett Field, California, NASA Technical Publication NASA/TP–2018–219875, Mar. 2018. [Online]. Available: https://ntrs.nasa.gov/citations/20180003961

[49] T. N. Rebolo, “Nc2s repository: Source code and prototypes,” 2025, gitHub repository. [Online]. Available: https://github.com/TomasRebolo/NC2S-Repository

[50] D. Dolev and A. Yao, “On the security of public key protocols,” *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[51] N. S. A. (NSA), *STANAG 4586 Ed.3 Nov 2012, Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability*, NATO Standardization Agency, 2012.

NC2S Flowcharts

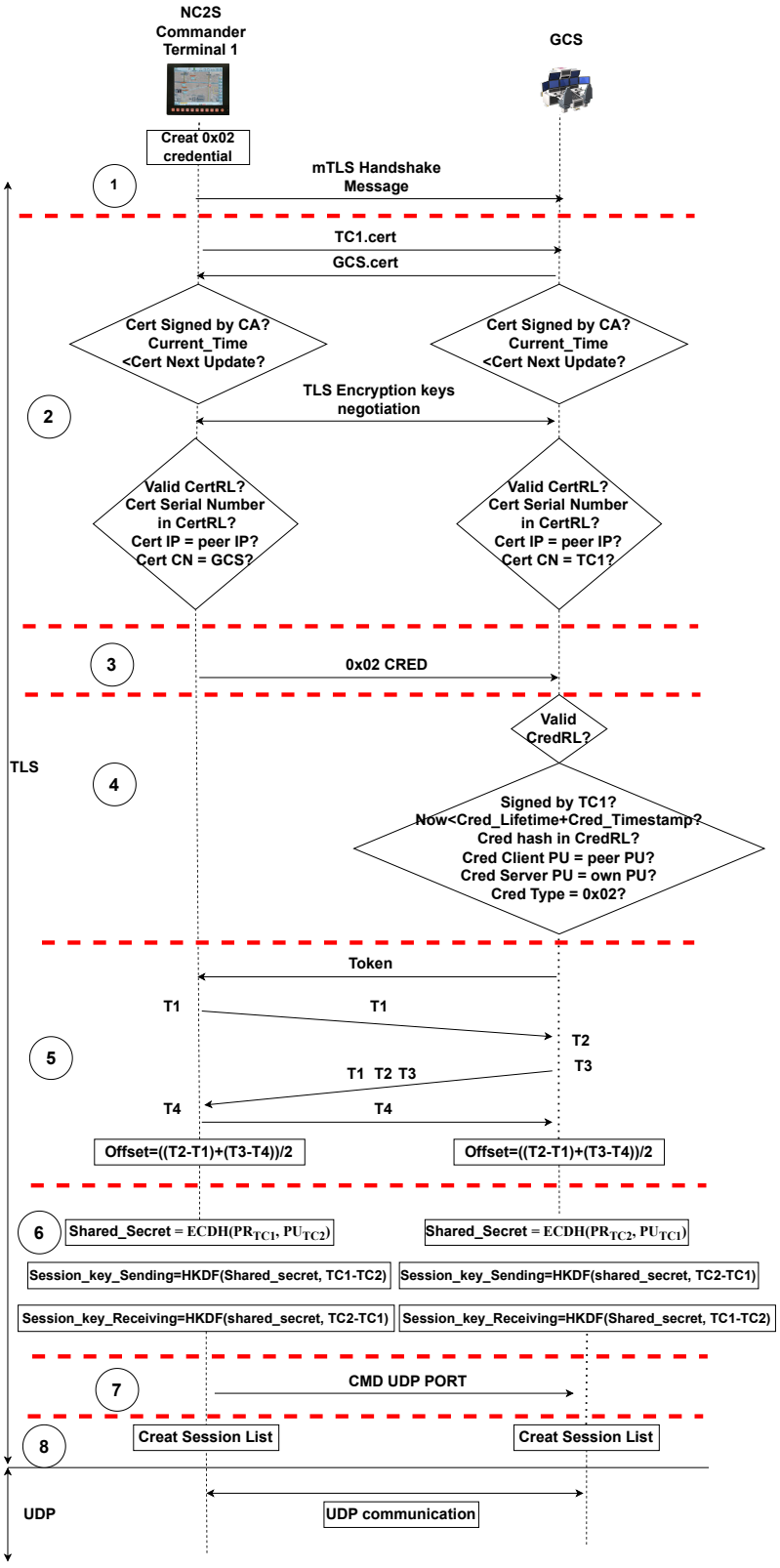

Figure A.1: TC1–GCS Connection Establishment

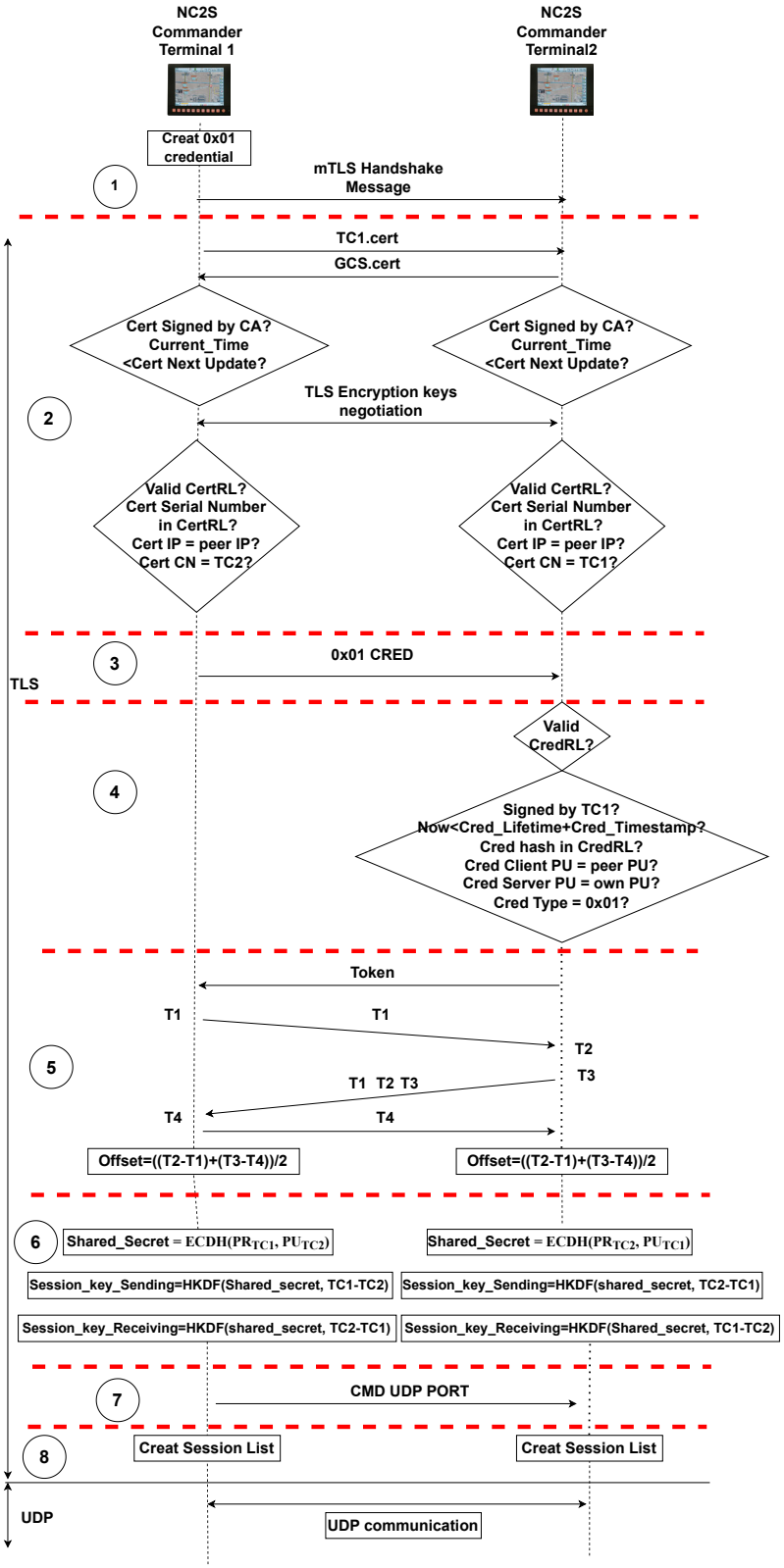

Figure A.2: TC1–TC2 Connection Establishment

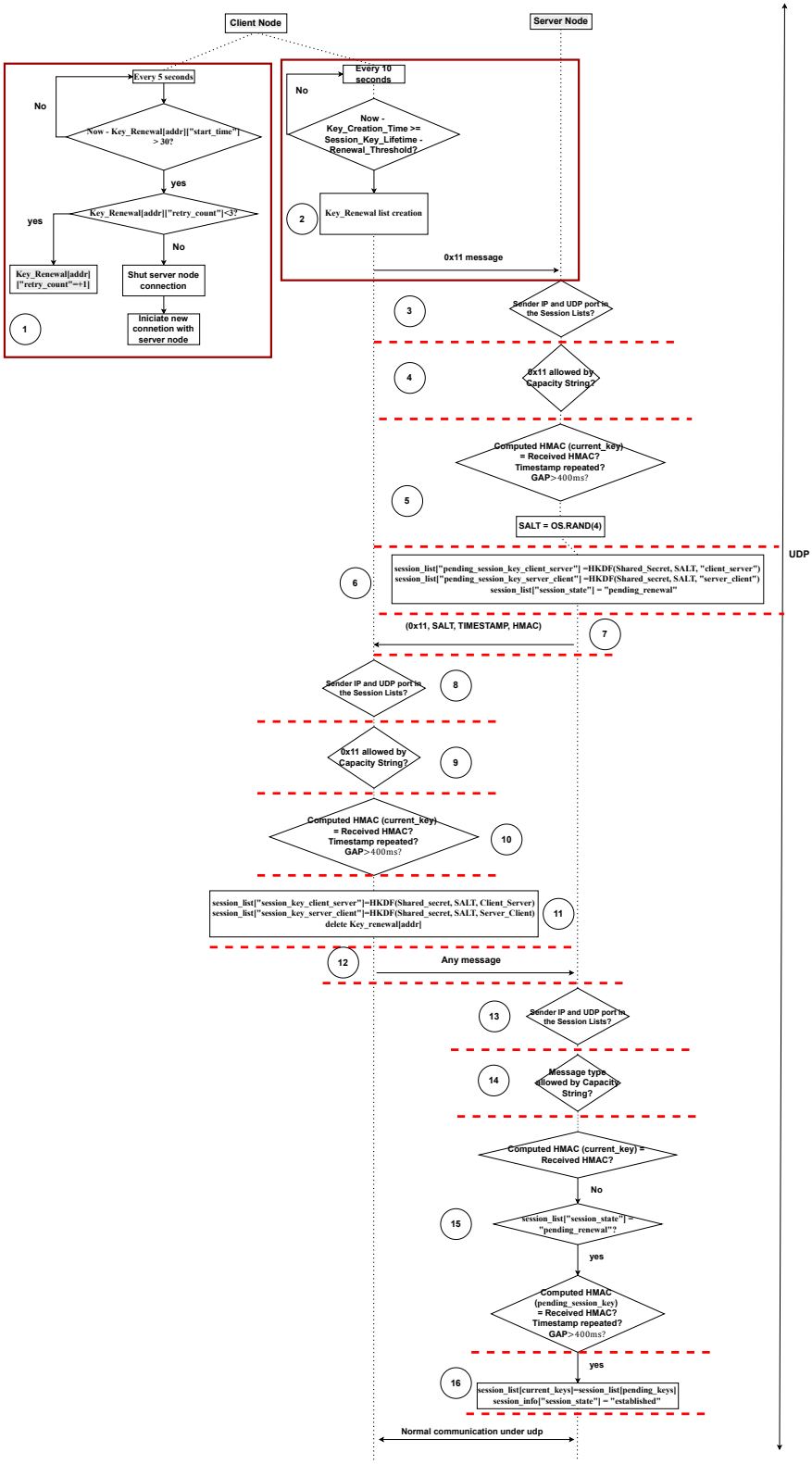

Figure A.9: Key Renewal Protocol

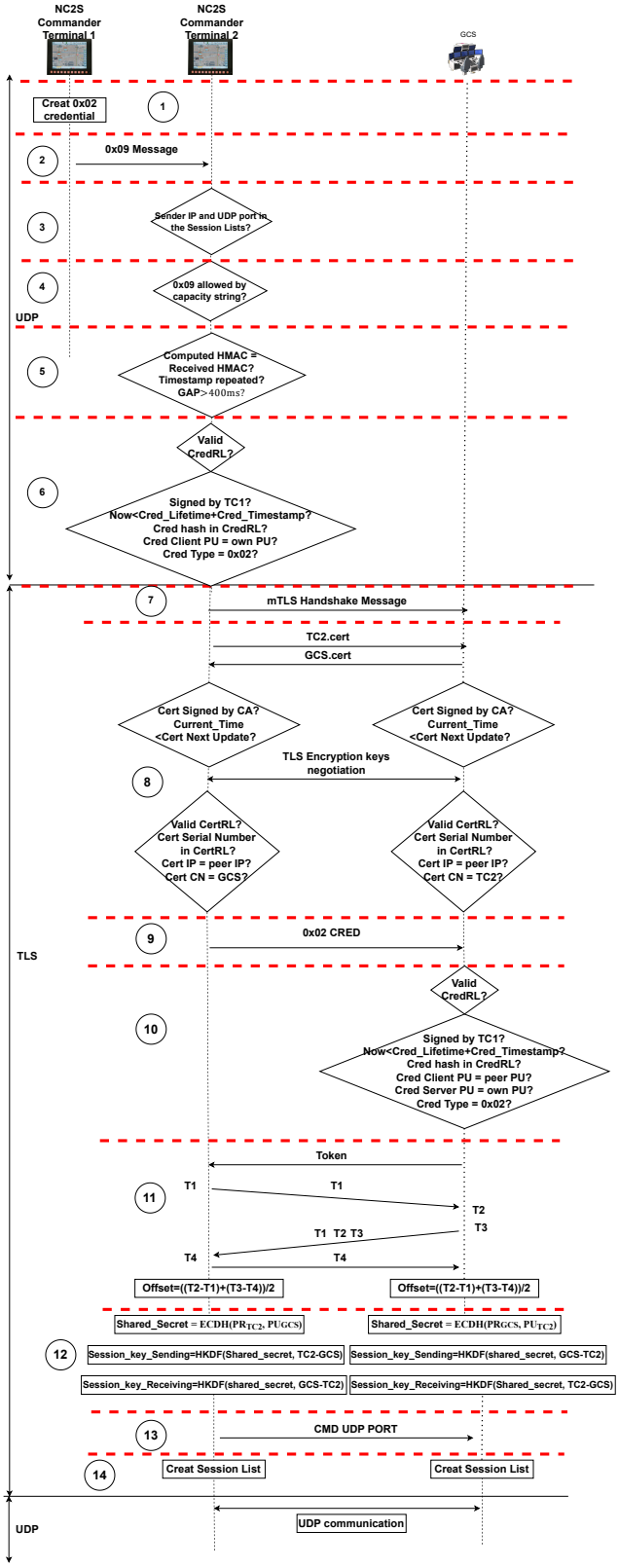

Figure A.3: TC1–TC2–GCS

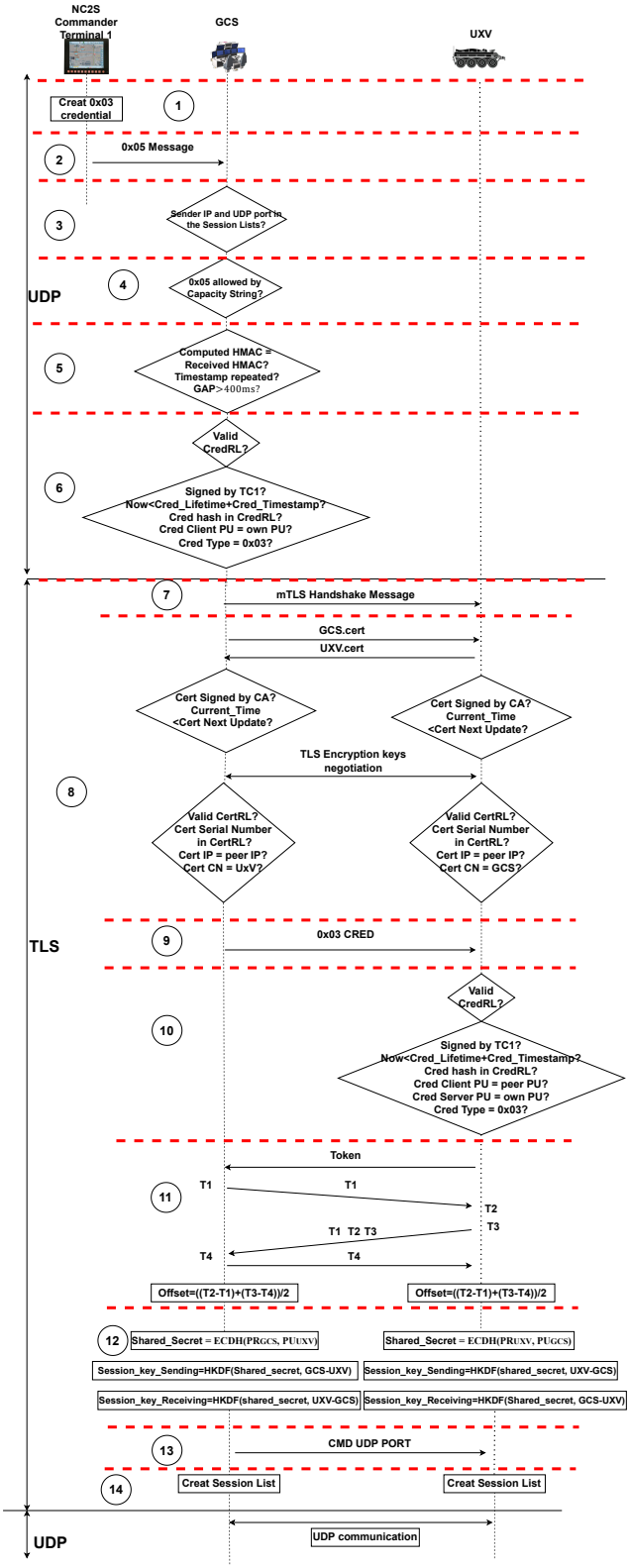

Figure A.4: TC1–GCS–UxV

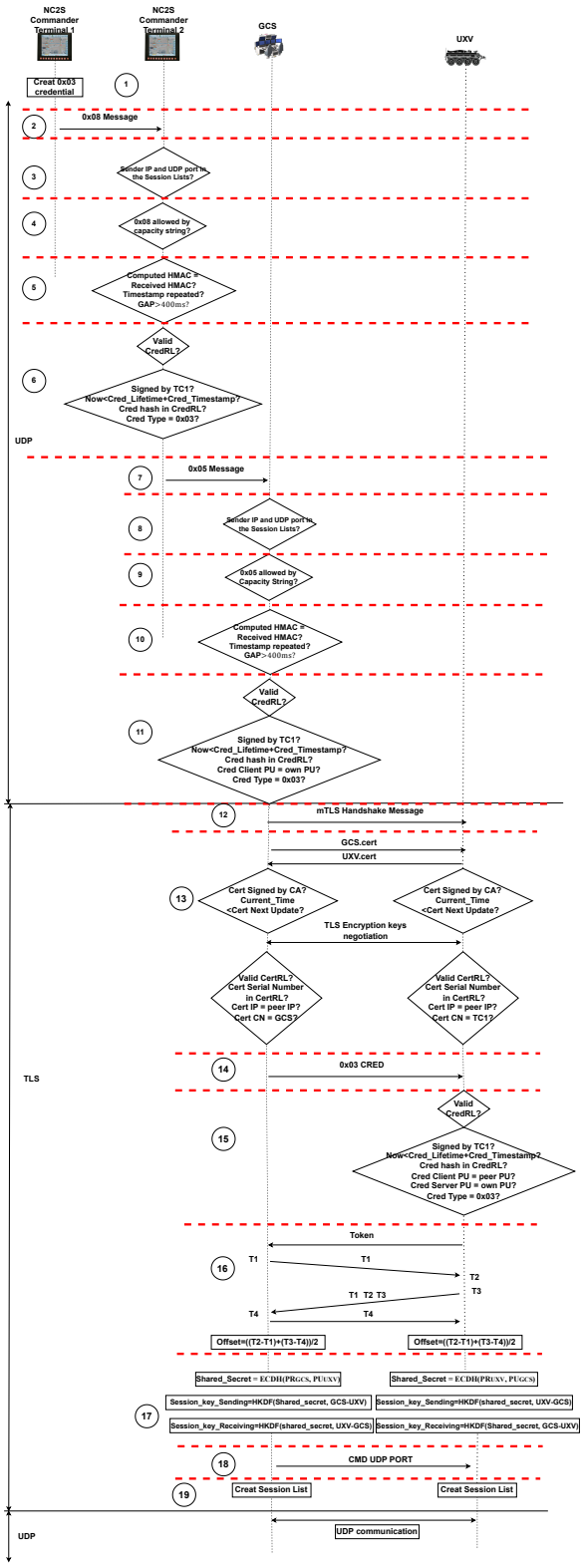

Figure A.5: TC1–TC2–GCS–UxV

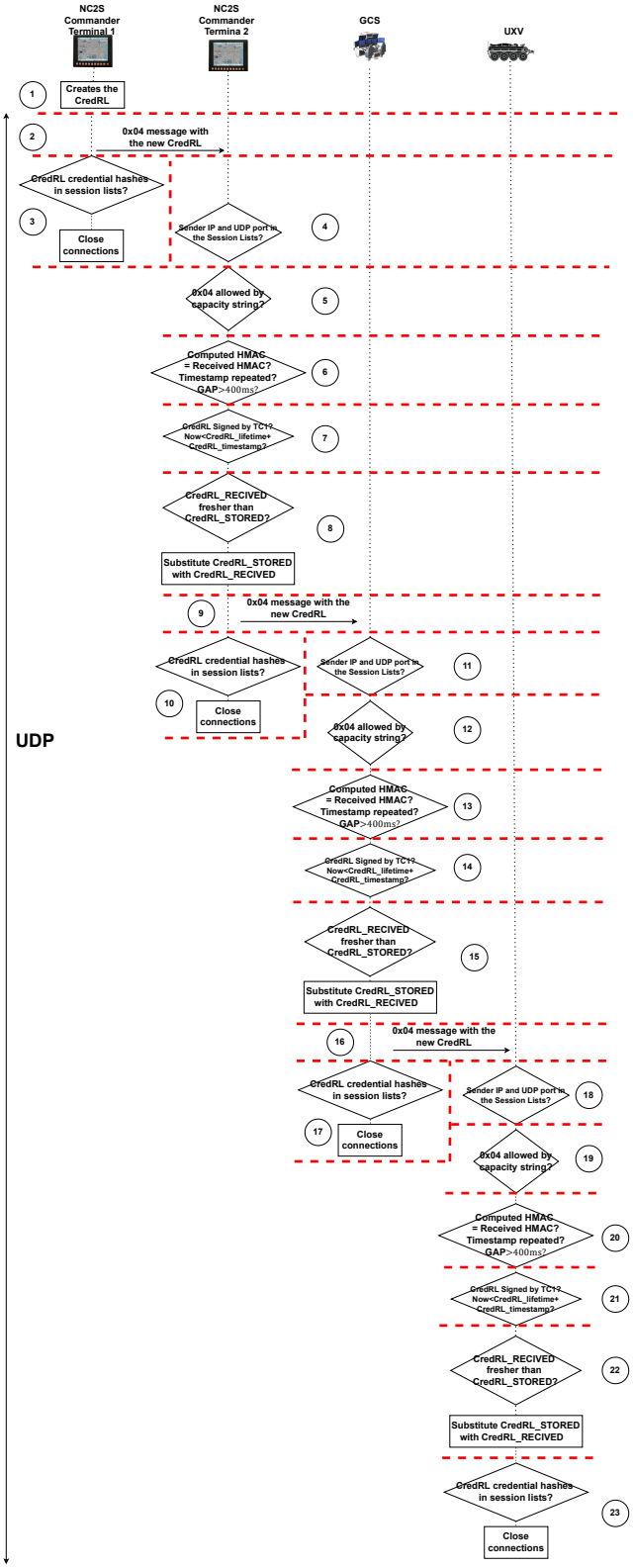

Figure A.6: Credential Revocation Protocol

Figure A.7: Certificate Revocation Protocol

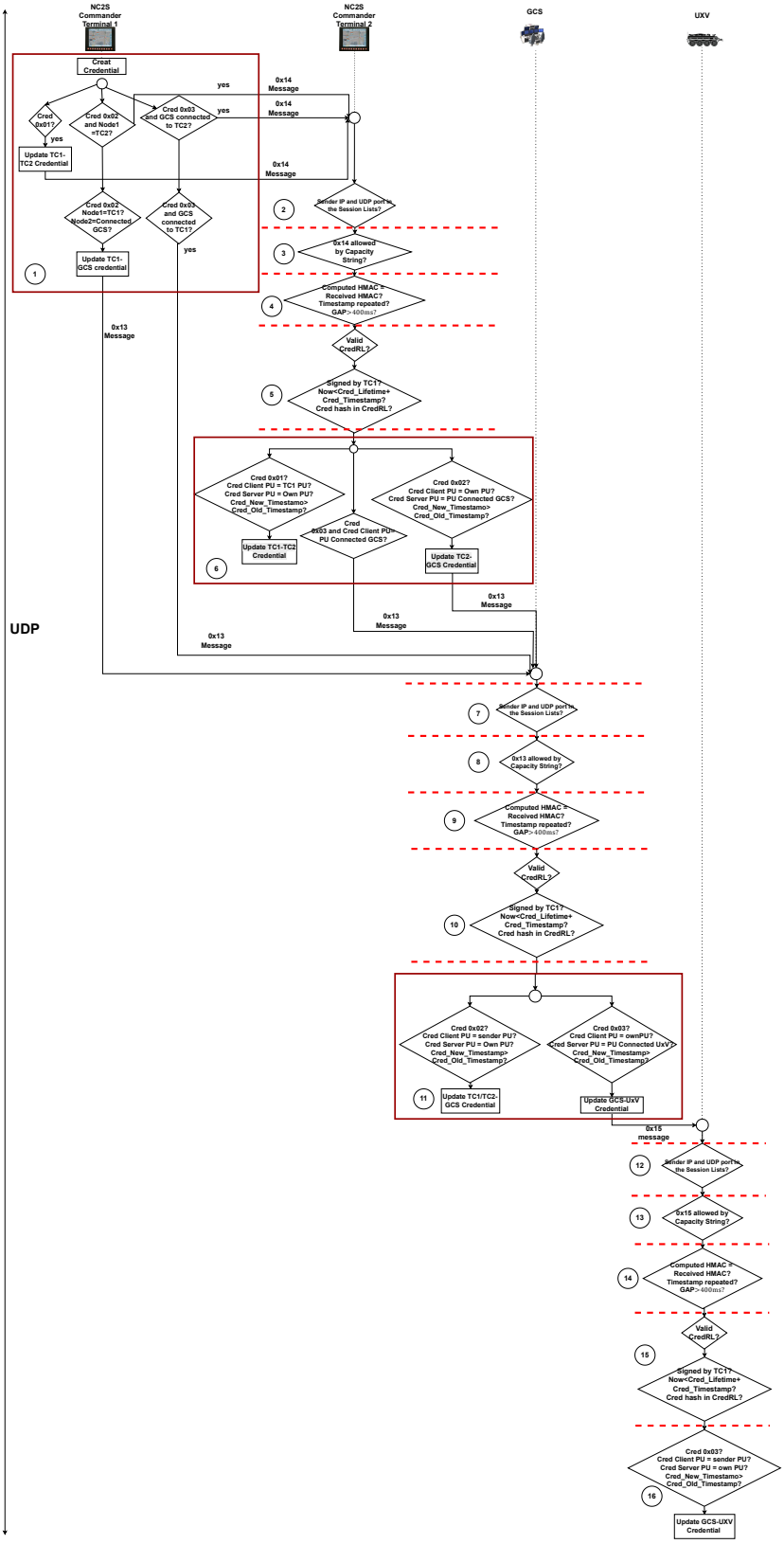

Figure A.8: Credential Update Protocol

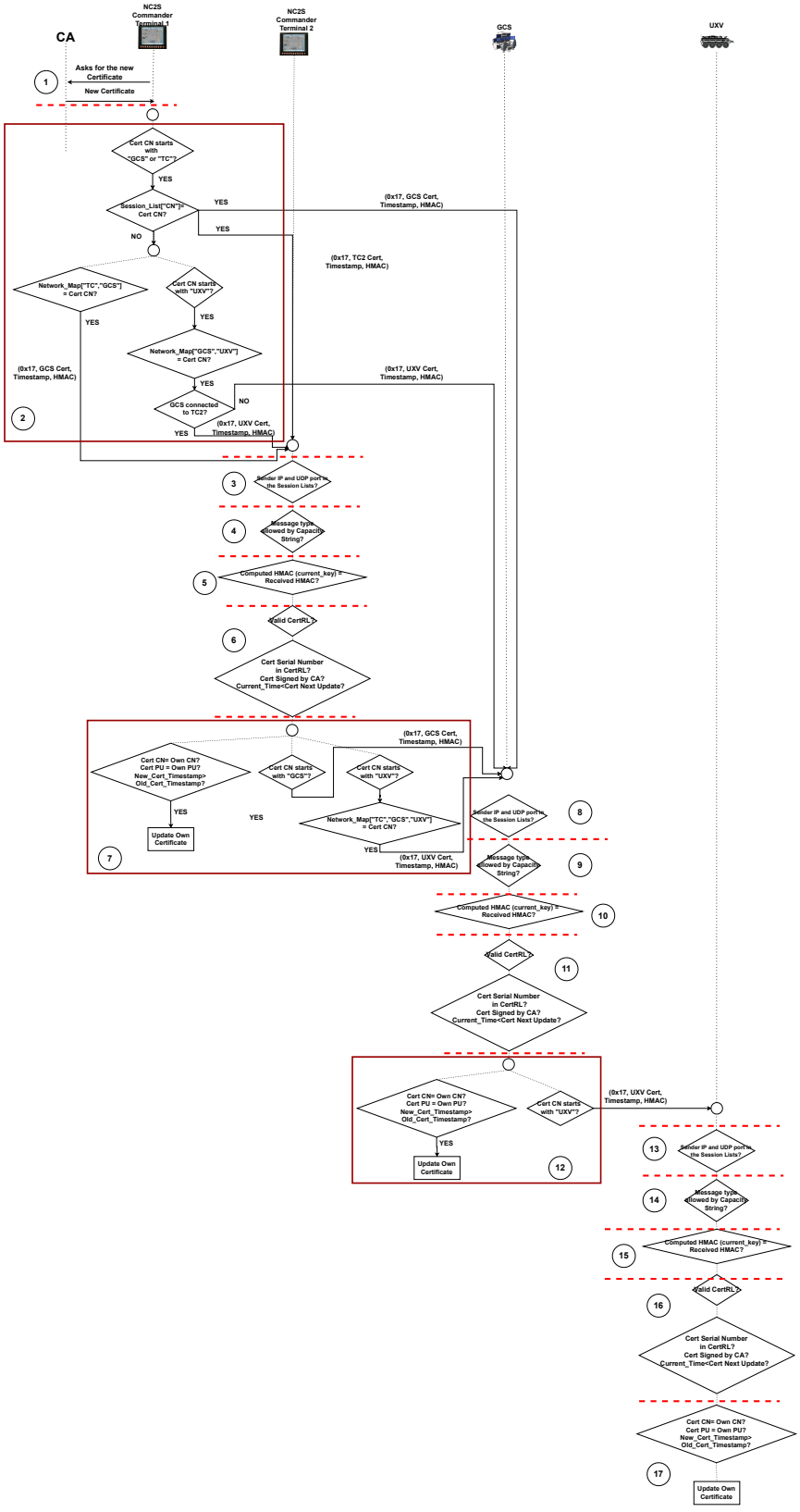

Figure A.10: Certificate Renewal Protocol

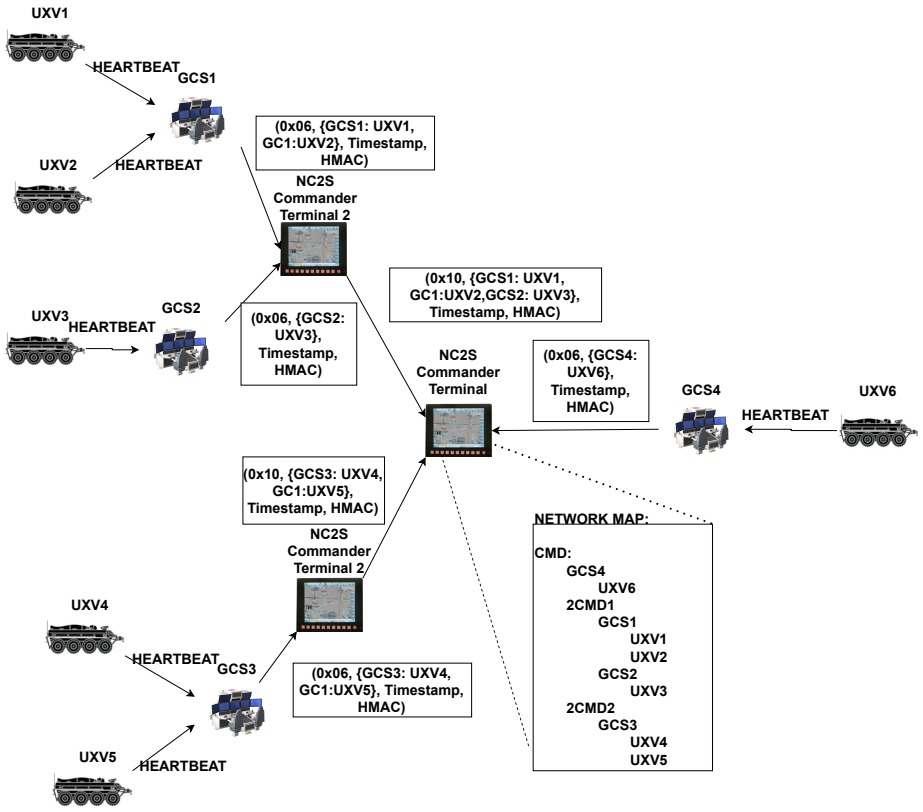

Figure A.11: Network Map Update Protocol.

Figure A.12: CT1 GUI.

Figure A.13: CT2 GUI