57º SBPO
SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
Gramado • RS
DE 5 A 9 DE OUTUBRO DE 2025

Modeling Secure MAVLink Protocol in the Tamarin Prover

Chandler Klüser Chantre[1],Anderson Fernandes Pereira dos Santos [1,2], Paulo Fernando Ferreira Rosa [1]
[1]Instituto Militar de Engenharia
LIARC - Laboratório de Inteligência Artificial, Robótica e Cibernética
Praça General Tibúrcio, 80 - Rio de Janeiro, RJ, 22290270, Brasil
[2]Venturus Centro de Inovação Tecnológica
Centro de Excelência em Tecnologias Emergentes
Estrada Giuseppina Vianelli di Napolli, nº 1.185 - Condomínio GlobalTech Campinas - Polo II de Alta Tecnologia - Campinas, SP, 13086530, Brazil
chandler@ime.eb.br

ABSTRACT

We present an intermediate formalization of the MAVLink protocol in the Tamarin Prover as the initial phase of a broader Software-In-The-Loop (SITL) simulation pipeline. In the first step, we define four core multiset-rewriting rules: InitKey, GCS_Send, Drone_Recv, and Replay_Attack to model key establishment, encrypted command transmission with monotonically increasing sequence counters, command reception with replay protection, and adversarial replay capabilities. The Tamarin model will be integrated into an under-development QEMU-based ArduPilot SITL environment. Ongoing work focuses on embedding authenticated telemetry into the formal model, automating proof strategies within Tamarin, and forging a seamless link between symbolic verification and the protocol's runtime behavior.

KEYWORDS. MAVLink, Tamarin, Replay Attack Model.

1. Introduction

Unmanned Aerial Vehicles (UAVs) currently rely on the MAVLink protocol to exchange control commands and telemetry data in real time. Although widely adopted in both civilian and military applications, MAVLink provides no native mechanisms to guarantee the core pillars of information security, neither any kind of specific malicious traffic, like the one generated through Replay Attack. In standard setting, messages are sent in clear text and protected only by a simple checksum. Using Tamarin Prover, a specific modeling can be developed, under the Dolev–Yao attacker approach, an adversary on the same network can intercept sensitive communications, inject malicious payloads, or replay previously captured packets, potentially seizing control of the drone mid-flight or corrupting mission data without detection.

To ensure the secure adoption of the protocol in scenarios where is required high levels of information protection (like confidentiality, privacy, availability, integraty and authenticity), we proposed a two-stage research project. In the first stage, we developed a symbolic multiset-rewriting model with four core rules: InitKey, GCS_Send, Drone_Recv, and Replay_Attack. Each one responsible, respectively, for shared-key generation, encrypted command transmission, reception with replay protection, and modeling of replay attacks. We then formalized two central security goals—secrecy of command parameters

and uniqueness of sequence as lemmas. Under the idealized abstractions of perfect cryptography and reliable delivery, these properties have already been validated: Tamarin automatically proves secrecy, while a brief interactive induction confirms replay resistance The Tamarin Team [2024]. In this regard, Merabet et al. [2025] and Mai and Haque [2024] enhance MAVLink frame sizes to incorporate content encryption, thereby strengthening data security. Similarly, Sabuwala and Daruwala [2022] demonstrates that lightweight algorithms, such as ChaCha20, can be effectively implemented for MAVLink security while maintaining minimal performance overhead.

In the second stage, we will extend and validate the model within a controlled Software-In-The-Loop (SITL) environment. To this end, we will run ArduPilot in QEMU (emulating a Raspberry Pi 3B), integrated with a Python-based ground-station client and a configurable attacker node capable of intercepting, modifying, or replaying MAVLink UDP packets. The testbed is currently in an advanced phase of implementation and validation, this to become ready to the planned experiments. At the same time, we will improve the symbolic model to include MAC-based authentication, simulate more realistic packet-loss and latency effects, and develop automated proof strategies in Tamarin, thereby minimizing manual intervention. Thahsin et al. [2025] robustly reinforces the validity of employing digital testbeds for such evaluations, and also demonstrates the feasibility of implementing AES encryption in counter mode.

The article is organized as follows: section 2 presents the fundamental concepts of the MAVLink protocol, and section 3 introduces the main features and functionalities of the Tamarin Prover. Next, in section 4 we detail the formal modeling of the protocol developed to date. Section 5 shows the results of the Tamarin executions and includes a critical analysis with guidelines for the model's future extensions. Finally, the last section discusses the lessons learned and the prospects for adopting the Tamarin Prover in similar research projects.

2. The MAVLink Protocol

MAVLink (Micro Air Vehicle Communication Protocol) is a lightweight protocol widely adopted in civil and military UAVs Koubâa et al. [2019]. It was designed to enable real-time exchange of telemetry, navigation commands, and detailed sensor data between the vehicle and a ground station—whether operated by a human pilot or by automated systems.

The communication between an UAV and its Ground Control Station (GCS) follows a structured process involving both periodic status messages and command exchanges. Initially, both endpoints periodically transmit HEARTBEAT messages containing status information to announce their presence and verify network integrity, as exhibit in Figure 1.

type	**autopilot**	**base_mode**	**custom_mode**	**system_status**	**mavlink_version**
8 bits	8 bits	8 bits	32 bits	8 bits	8 bits

Figure 1: MAVLink Heartbeat Message

When the GCS issues a command (such as COMMAND_LONG) it constructs a MAVLink frame composed of a header (including the Magic Value, STX, payload length, sequence counter SEQ, system and component identifiers, MSG ID, and other flags as described in Figure 2), a payload with the command parameters, and a two-byte CRC calculated according to the X.25/ITU standard.

STX	LEN	INC FLAGS	CMP FLAGS	SEQ	SYS ID	COMP ID	MSG ID (3 bytes)

Figure 2: MAVLink 2.0 Header adapted from Koubâa et al. [2019]

This frame is then transmitted over various network protocols such as UDP, TCP, or serial communication links like UART, RS-232, LoRa, or generic radio frequency (RF) channels, depending on the system configuration and operational requirements. Furthermore, Hamza et al. [2024] point that MAVLink operate across a wide range of hardware platforms, which make it a popular choice for both industrial and academic settings.

The MAVLink protocol presents several vulnerabilities documented by researchers such as Ficco et al. [2022]; Mekdad et al. [2024]. A structured synthesis of these weaknesses was presented by Koubâa et al. [2019], who classified them based on the pillars of information security, characterizing each pillar according to the type of cyber action performed, as illustrated in Figure 3.

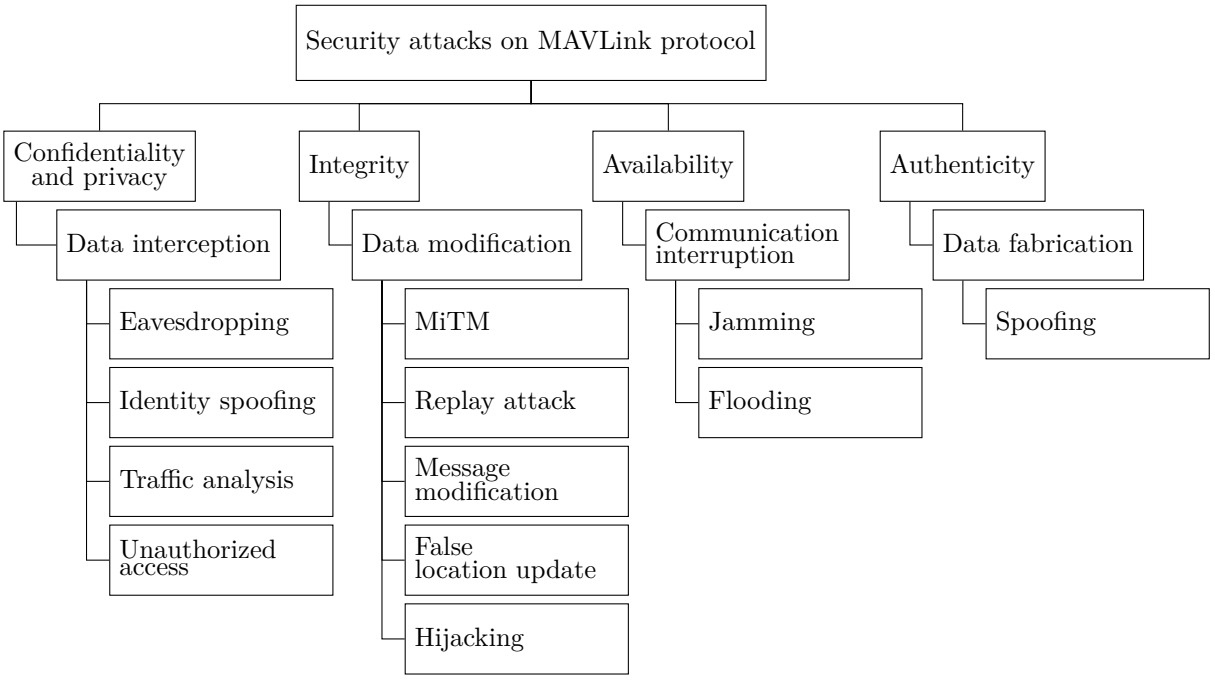

Figure 3: Security threats and attacks against MAVLink Protocol

If data is intercepted, confidentiality and/or privacy may be compromised. In that situation, classified information can be displayed to a non-authorized people. There are a many attacks that have this goals, but in our scenario, this can occur through the attacks listed in Table 1.

Table 1: Data Interception

Attack Type	Description
Eavesdropping	The attacker passively listens to MAVLink communications to capture sensitive information such as telemetry data, commands, or system identifiers, compromising confidentiality.
Identity Spoofing	The attacker impersonates a legitimate UAV or GCS by forging MAVLink identifiers, enabling unauthorized participation in the communication.
Traffic Analysis	Even without reading the content of the messages, the attacker observes the communication patterns and frequency to infer operational behaviors and mission structure.
Unauthorized Access	The attacker gains access to the MAVLink network or devices without proper authorization, either physically or remotely, enabling further exploitation.

If the data is modified, integrity is compromised. The altered data may be retransmitted, but it still falls under the taxonomy characterized by the breach of data integrity during its interception and modification. This can occur through the attacks listed in Table 2.

Table 2: Data Modification

Attack Type	Description
Man-in-the-Middle (MitM)	The attacker intercepts MAVLink messages between the UAV and GCS, allowing the alteration, suppression, or injection of packets in real time, compromising the message flow integrity.
Replay Attack	Previously captured valid messages are retransmitted to deceive the receiver, potentially causing repeated commands or outdated state updates that threaten operational consistency.
Message Modification	The attacker captures and alters the contents of a MAVLink message (e.g., command parameters or telemetry data), misleading the receiver and violating data integrity.
False Local Update	The attacker injects forged status or sensor data into the communication, leading the UAV or GCS to update internal states with incorrect information.
Hijacking	By altering identification fields or command content, the attacker takes control of the UAV's behavior or communication channel, redirecting operations maliciously. For this content, we do not distinguish skyjacking and radio jacking, as Koubâa et al. [2019] has done.

If communication is interrupted, serious damage may occur in UAV management, leading to unpredictable effects and thus characterizing a likely loss of availability. The attacks presented in Table 3 are common examples that can lead to this situation.

Table 3: Communication Interruption

Attack Type	**Description**
Jamming	The attacker deliberately emits radio frequency signals on the same frequency band used by MAVLink to disrupt or completely block communication between the UAV and the GCS.
Flooding	The attacker overloads the communication channel or processing capacity by sending a high volume of messages. This category includes Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks.

It is also possible for data be artificially generated to simulate a different situation, thereby characterizing the class of spoofing attacks. In this condition, the data is no longer authentic. This situation is described in Table 4.

Table 4: Authenticity

Attack Type	**Description**
Spoofing	The attacker generates forged messages or impersonates legitimate entities (such as a UAV or GCS) in order to deceive the system into accepting false information as authentic, without necessarily modifying existing data. This violates the authenticity of the communication.

2.0.1. Replay Attack

This type of attack consists of intercepting legitimate messages exchanged between the UAV and its Ground Control Station (GCS), followed by their retransmission at later moments. Since the MAVLink protocol, in its standard version, lacks native mechanisms for authentication and replay protection, these reused messages are treated as valid by the receiving system.

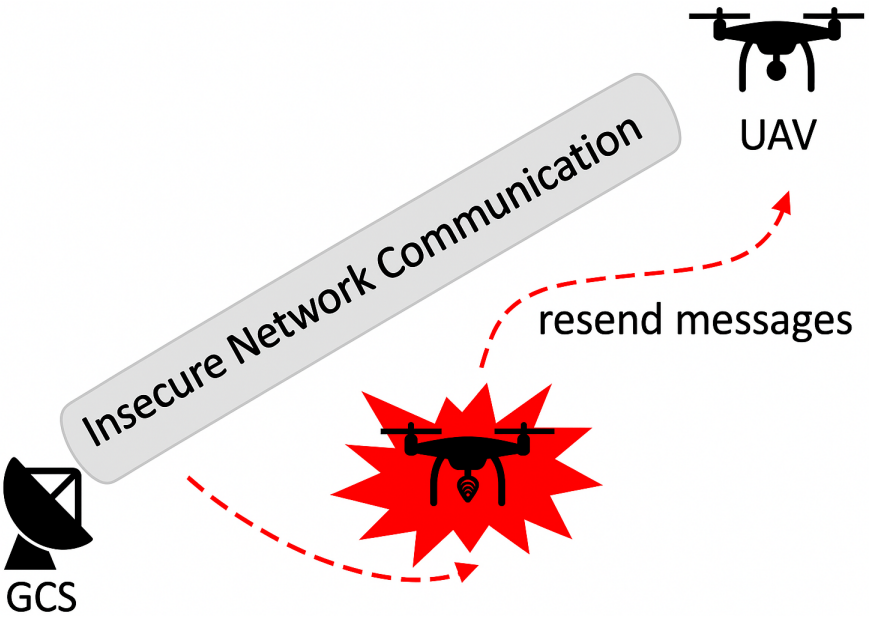

Figure 4: Replay Attack in UAV

The consequences of this kind of attack are varied and concerning. One of the most immediate impacts is the possibility of unauthorized command execution. For example, a takeoff or mission change command that was correctly issued at an earlier time may be captured and later replayed, causing the UAV to execute the action again outside its intended context, thus compromising mission safety. Furthermore, the repetition of messages can lead to inconsistencies in the operational states between the GCS and the UAV, resulting in loss of control over the vehicle.

Another critical aspect is the difficulty in detecting this type of attack. Since the message is not tampered with, basic integrity verification mechanism, like CRC, are unable to identify the replay, allowing the attack to go unnoticed. When combined with other techniques, such as spoofing or session hijacking, replay attacks can be used to trigger more complex malicious behaviors, including route deviations, forced landings, or deliberate mission interruption.

We prioritized the absence of replay-attack protection for our initial formal model because it represents an immediate threat to command integrity and is readily captured by sequence-counter semantics. The other identified weaknesses—lack of confidentiality, integrity guarantees, and DoS resilience—will be systematically addressed in the project's next phase.

2.1. The Tamarin Prover

Tamarin is a symbolic model checker for security protocols that combines a multiset, rewriting engine with first-order logic. In its core, each protocol action is encoded as a rewrite rule that consumes and produces facts in a global multiset.

A Tamarin Prover document consists of several sections that formally define a symbolic security model for communication protocols. These sections are used to describe the available cryptographic operators, the behavior of protocol participants, the messages exchanged, and the security properties to be verified.

The structure of a Tamarin document typically begins with the *builtins section*, which specifies the predefined cryptographic primitives used in the model, such as symmetric encryption, asymmetric encryption, and hash functions. Next, custom operators can be declared in the *functions section* if the model requires constructs not provided natively.

If algebraic relationships between operators are needed, then *equations section* is used to represent that. Although not always used, the *types section* allows for the explicit definition of data types, which can be useful for modularity and stricter validation.

The core of the specification lies in the rules, defined by *rule blocks*. Each rule describes a system transition by indicating which facts must be present, which actions are logged during protocol execution, and which new facts become true after the transition. Rules may include equations al theories, like encryption, hashing, or modular arithmetic functions, and explicitly model the adversary under the Dolev–Yao assumption, enabling precise reasoning about message flows, key compromise, and algebraic properties The Tamarin Team [2024].

The Dolev–Yao model (Dolev and Yao [1983]) is a fundamental assumption in protocol security that defines the capabilities of an attacker within a communication system. This model is widely used in formal verification of cryptographic protocols where it is supposed that the attacker is considered extremely powerful from a network perspective. The attacker is able to intercept, block, modify, resend, or create messages based on their knowledge, but is limited in their ability to use cryptography, so the attacker cannot break cryptographic primitives.

This approach allows the abstraction of the computational details of cryptography, treating cryptographic primitives as ideal functions (or “black boxes”). This makes the model particularly effective for analyzing design flaws in protocols, such as authentication failures, incorrect use of encryption, or lack of protection against replay attacks. Thus, even assuming that the cryptographic algorithms used are perfect, the Dolev–Yao model helps identify vulnerabilities related to message logic and flow within the protocol.

Furthermore, the security properties to be verified are expressed using *lemmas*. These lemmas are written in temporal logic and describe, for example, that certain information remains secret, that two entities agree on a value, or that message replay does not occur.

Facts are classified as either linear or persistent. The first one, linear facts, represents ephemeral events or resources and are removed from the multiset once a consuming rule fires, whereas persistent facts model enduring state and remain available across multiple rule applications. This distinction allows modeling phases temporary or permanent.

Security goals are expressed as *Lemmas*, which quantify universally or existentially over execution traces. A secrecy lemma, for instance, may assert that "no trace exists in which the adversary derives a particular key," while an authentication lemma ensures correspondence between send and receive events. Tamarin’s proof engine can discharge many lemmas fully automatically using SMT-based constraint solving; for more complex properties, it provides an interactive mode in which the user can apply custom proof strategies, induction hypotheses, or lemma strengthening.

```
theory Dummy_Protocol_Model
begin

/* ===== 1. Built-ins ===== */
builtins: symmetric-encryption, asymmetric-encryption, hashing

/* ===== 2. Customized functions ===== */
functions: f/1, g/2, pair/2

/* ===== 3. Equations ===== */
equations:
  g(f(x), y) = pair(x, y)

/* ===== 4. Types ===== */
types: agent, key, nonce, data

/* ===== 5. Rules ===== */
rule AttackerPlays:
  [ In(x) ]
  --[ AttackerReplays(x) ]->
  [ Out(x) ]

/* ===== 6. Global restrictions ===== */
restriction UniqueReception:
  "All N #i #j. DummyReceived(N)@i &  DummyReceived(N)@j ==> #i = #j"

/* ===== 7. Reusable Macros ===== */
macro WasReceived(n) :=

/* ===== 8. Security Lemmas ===== */
lemma key_secrecy: "not (Ex k #i. InitKey(k) @ i & K(k) @ i)"

end theory
```

3. Secure MAVLink Tamarin Model

In this initial modeling phase using Tamarin approach, it was decised to work with Replay Attack, because the reasons already described. The mitigation proposed is formed by four rules and two lemmas. Each one of these to share a symmetric-encription key and avoid replay attack. For each rule and lemma used in our Tamarin model will be presented and further discussed. The complete code will be available in project research repository.

3.1. InitKey: Shared Key Initialization

```
rule InitKey:
    [ Fr(~K) ]
    --[ SharedKey(~K) ]->
    [ !Key(~K) ]
```

- **Purpose**: Generate and publish a fresh symmetric key for subsequent use.

- **Precondition**: None.
- **Action**: A new constant `K` is created (`Fr( K)`).
- **Event**: Emits `SharedKey( K)` to record that `K` has been shared.
- **Postcondition**: The persistent fact `!Key( K)` makes `K` available to all future rules.

3.2. GCS_Send: Encrypted Command Transmission with Sequence Counter

```
rule GCS_Send:
    [ !Key(K), Fr(Seq), Fr(N), Fr(cmd) ]
    --[ SendCommand(Seq,N) ]->
    [ Out(senc(pair(Seq, pair(cmd, N)), K)) ]
```

- **Purpose**: Package and encrypt a command along with a sequence counter.
- **Preconditions**:
  1. Persistent fact `!Key(K)` exists (shared symmetric key available).
  2. Fresh nonces `Seq`, `N`, and command payload `cmd` are generated.
- **Action & Event**: Emits `SendCommand(Seq,N)` to log the send event.
- **Postcondition**: Produces `Out(senc(pair(Seq,pair(cmd,N)),K))`, i.e., the pair `(Seq,(cmd,N))` encrypted under `K`.

3.3. Drone_Recv: Replay-Protected Command Reception

```
rule Drone_Recv:
    [ !Key(K), In(senc(pair(Seq, pair(cmd, N)), K)),
    !Received(Seq) ]
    --[ CommandReceived(Seq,N) ]->
    [ !Received(Seq) ]
```

- **Purpose**: Decrypt incoming commands and block replay attacks.
- **Preconditions**:
  1. Persistent fact `!Key(K)` (shared key available).
  2. Input of the encrypted message `In(senc(pair(Seq,pair(cmd,N)),K))`.
  3. Persistent fact `!Received(Seq)`, ensuring `Seq` has not been seen before.
- **Action & Event**: Emits `CommandReceived(Seq,N)` upon accepting the command.
- **Postcondition**: Updates `!Received(Seq)` to prevent future replays of the same sequence.

3.4. Replay_Attack: Adversarial Replay Modeling

```
rule Replay_Attack:
    [ In(X) ]
    --[ Replay(X) ]->
    [ Out(X) ]
```

- **Purpose**: Simulate the adversary's ability to replay any observed message.
- **Precondition**: The attacker has intercepted some message `X` (`In(X)`).
- **Action & Event**: Emits `Replay(X)` to indicate a replay attempt.
- **Postcondition**: Outputs `Out(X)`, retransmitting the same message.

3.5. Security Secrecy Lemma

- **Statement**:

$$\neg(\exists K, i.\ SharedKey(K)@i \ \wedge \ K(K)@i).$$

- **Meaning**: There is no execution trace in which the adversary both observes the `SharedKey(K)` event and derives `K`.

3.6. Security No_Replay Lemma

- **Statement**:

$$\forall\, Seq, N, i, j.\ CommandReceived(Seq, N)@i \wedge CommandReceived(Seq, N)@j \implies i = j.$$

- **Meaning**: Any given sequence number `Seq` can trigger `CommandReceived` at only one observation point, thus preventing replay attacks.

4. Proof Trace Analysis

Below is an analysis of the proof trace, including excerpts from the Tamarin frontend output.

4.1. Key Persistence:

confirms that `InitKey` generates and stores the key `t.1` as a persistent fact.

4.2. Replay-Protected Reception:

for each new sequence number `Seq`, the rule `Drone_Recv` fires exactly once, blocking replays.

4.3. Attacker Knowledge Facts:

All these `!KU(...)` obligations are solved (marked "currently deducible" or "probably constructible"), indicating that under the Dolev–Yao model the adversary can derive exactly these terms.

4.4. No Remaining Obligations:

are justified by at least one rule or by Tamarin's built-in SMT solving. This demonstrates that our model fully covers key establishment, replay protection, and the adversary's term-construction capabilities.

5. Discussion

This work presents the initial phase for modeling security protection to UAVs communication witg MAVLink. To reach this goal, a two-stage framework that combines symbolic modeling in the Tamarin Prover with an eventual Software-In-The-Loop (SITL) validation Ficco et al. [2022], Koubâa et al. [2019] is under-development. The idealized assumptions of perfect cryptography and reliable delivery, our Tamarin model demonstrates that simple multiset-rewriting rules can enforce both command secrecy and strict sequence-number uniqueness, effectively neutralizing replay attacks. In MAVLink protocol, there was identified a great number of vulnerabilities, as also described, and Replay Attack was priorized becasue the impact in real scenarios.

The four core rewrite rules—`InitKey`, `GCS_Send`, `Drone_Recv`, and `Replay_Attack` capture the essential phases of MAVLink communication: key establishment, encrypted command transmission with monotonically increasing counters, replay-protected reception, and adversarial replay capabilities. The two security lemmas (`secrecy` and `no_replay`) are discharged automatically (for secrecy) or with minimal interactive effort (for no-replay), and our proof trace confirms that each persistent fact and attacker-knowledge obligation is generated and resolved exactly as intended, leaving no unresolved constraints.

Looking ahead, integrating this formal model into a QEMU-based ArduPilot SITL testbed will allow us to observe the correspondence between symbolic replay detections and real-world misbehaviors. Future extensions will incorporate MAC-based authentication, realistic packet-loss and latency effects, and fully automated proof strategies in Tamarin. By bridging symbolic verification with run-time protocol behavior, we aim to deliver a comprehensive, end-to-end assurance pipeline for secure MAVLink deployments.

References

Dolev, D. and Yao, A. C. (1983). On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208.

Ficco, M., Palmiero, R., Rak, M., and Granata, D. (2022). MAVLink Protocol for Unmanned Aerial Vehicle: Vulnerabilities Analysis. In *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing (DASC/PiCom/CBDCom/CyberSciTech)*, p. 1–6.

Hamza, M. A., Mohsin, M., Khalil, M., and Kazam Abbas Kazmi, S. M. (2024). MAVLink Protocol: A Survey of Security Threats and Countermeasures. In *2024 4th International Conference on Digital Futures and Transformative Technologies (ICoDT2)*, p. 1–8.

Koubâa, A., Allouch, A., Alajlan, M., Javed, Y., Belghith, A., and Khalgui, M. (2019). Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey. *IEEE Access*, 7:87658–87680. Conference Name: IEEE Access.

Mai, C. and Haque, A. (2024). MavSec: A safer version of MavLink. In *2024 International Wireless Communications and Mobile Computing (IWCMC)*, p. 768–773.

Mekdad, Y., Acar, A., Aris, A., El Fergougui, A., Conti, M., Lazzeretti, R., and Uluagac, S. (2024). Exploring Jamming and Hijacking Attacks for Micro Aerial Drones. In *ICC 2024 - IEEE International Conference on Communications*, p. 1939–1944.

Merabet, A., Lakas, A., Belkacem, A. N., Benamarouche, A., and Silva, P. (2025). eMAVLink: Enhancing MAVLink for Secure and Robust UAV Communication. In *2025 International Wireless Communications and Mobile Computing (IWCMC)*, p. 692–697.

Sabuwala, N. and Daruwala, R. D. (2022). Securing Unmanned Aerial Vehicles by Encrypting MAVLink Protocol. In *2022 IEEE Bombay Section Signature Conference (IBSSC)*, p. 1–6.

Thahsin, A., Ananthapadmanabhan, A., Pathak, S., Maity, A., and Kasbekar, G. S. (2025). Enhancing MAVLink Security: Implementation and Performance Evaluation of Encryption on a Drone Testbed. In *2025 International Conference on Information Networking (ICOIN)*, p. 529–534.

The Tamarin Team (2024). Tamarin prover manual: Security protocol analysis in the symbolic model. URL `https://tamarin-prover.com`. Licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International.