Formal Verification of Secure Protocols for Aviation Communication using Tamarin Prover:

Background & Motivation

The Future Communication Infrastructure (FCI) aims to ensure safe and reliable communication between aircraft and ground systems, even during handovers when an aircraft changes its network connection or service provider. Since FCI will handle safety-critical data, it is essential to confirm that attackers cannot intercept, replay, or impersonate messages. While traditional testing and simulation can help, they only cover a limited range of situations. Formal verification is an extra step that allows us to check that the protocol's most important promises (like secrecy and authenticity) are consistently maintained in every possible scenario, not just in a few test runs.

Tamarin Prover is a practical tool for checking the security of communication protocols. With Tamarin, you describe the steps of your protocol (who sends which messages and when), highlight important events (like starting or finishing a secure session), and write down the main guarantees you expect (for example, "only these two parties should know the session key" or "an attacker should not be able to reuse messages to break security"). Tamarin will then automatically check if these guarantees hold up in every possible situation covered by your model. If a guarantee doesn't hold, Tamarin will show a clear, step-by-step explanation of how things could go wrong.

Project Aim

Your project is to evaluate the security of a proposed FCI protocol (your supervisor/Mentor will provide a complete design) using the Tamarin Prover tool. You will model this protocol, check that it fulfills its main security goals, and help identify any potential weaknesses before the protocol is adopted for real-world aviation.

Scope & Key Features

- Use the provided FCI security protocol design as your main case for analysis.
- Model all protocol steps and cryptographic functions using Tamarin Prover.
- Focus on important security questions, such as:
    - Mutual authentication: Are both parties certain of each other's identity?
    - Session key secrecy: Is the shared key protected from others, including attackers or relays?

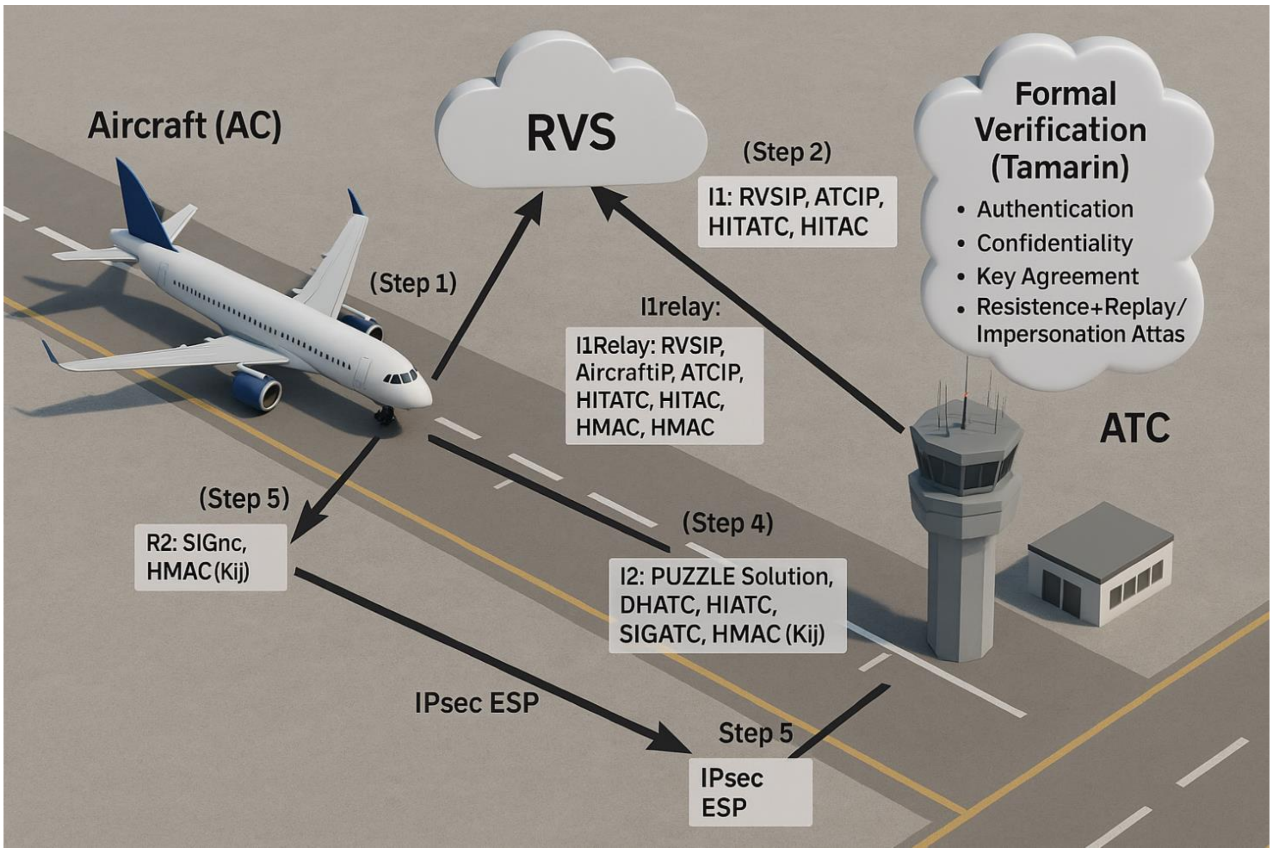
- Replay protection: Can the protocol prevent an attacker from successfully reusing old messages?

- Clearly label events and document any simplifications or assumptions you make (such as treating certificates as always valid).

Learning Goals

By completing this project, you will:

- Learn to read and interpret the structure of a modern aviation security protocol.
- Express protocol logic and key security expectations using Tamarin's language.
- Use formal verification tools to assess protocol security.
- Summarize and communicate your results clearly, highlighting both strengths and any areas that need improvement.


Aircraft (AC)
RVS
(Step 2)
I1: RVSIP, ATCIP, HITATC, HITAC
Formal Verification (Tamarin)
• Authentication
• Confidentiality
• Key Agreement
• Resistence+Replay/ Impersonation Attas
(Step 1)
I1relay:
I1Relay: RVSIP, AircraftiP, ATCIP, HITATC, HITAC, HMAC, HMAC
ATC
(Step 5)
R2: SIGnc, HMAC(Kij)
(Step 4)
I2: PUZZLE Solution, DHATC, HIATC, SIGATC, HMAC (Kij)
IPsec ESP
Step 5
IPsec ESP

Deliverables

- One Tamarin model file (.spthy) representing the provided FCI protocol, with checks for 3–5 core security guarantees.
- A concise report (6–10 pages):
    - Your assumptions and simplifications.

- Each security property is described in plain English.
- The results of your checks explained clearly and (if needed) illustrated with screenshots.
- Figures: At least one protocol message-flow diagram, and an attack-trace diagram if you find any vulnerability.

Prerequisite Knowledge (required/recommended)

Required (practical):

- Basic programming ability (working with small code files, understanding logic).
- The ability to read and understand short technical descriptions/specifications.
- Basic git knowledge for collaborating in a group.

Recommended (nice to have, not mandatory):

- Introductory security concepts (such as authentication and replay attacks).
- Courses like TDDE34 (Software Verification) is helpful but not required.
- Interest in applied security and practical protocol design.

Supporting Resources for Students

Since most students won’t have seen Tamarin before, here are some accessible references/tutorials you can share:

1. Official Tamarin Prover Manual (Beginner-friendly sections):
2. https://tamarin-prover.github.io/manual/
    - Covers installation, first example, and basic syntax.
    - A step-by-step, hands-on introduction to Tamarin with examples.
3. https://github.com/tamarin-prover/tamarin-prover/tree/develop/examples
    - Small working models you can run and adapt.

Note Please: (A mini-tutorial, starter template, and cheat-sheet will also be provided by Suleman Khan, a PhD student at LiU, and he will be your Mentor during the project and work closely with the students.)