Instructor: Nadim Kobeissi **Website:** https://appliedcryptography.page

Lab Assignment: Designing and Verifying a TLS-like Protocol using ProVerif

Overview

In this lab, you will design and formally verify a Transport Layer Security (TLS)-like protocol using ProVerif, a formal verification tool for cryptographic protocols. This represents your first opportunity to apply formal methods to verify the security properties of a complete cryptographic protocol—one that provides confidentiality, integrity, and authentication for network communications. By designing and verifying this protocol, you'll gain practical experience with cryptographic protocol design, formal verification, and security property specification. Future lab assignments will build upon these skills by incorporating more complex cryptographic protocols and verification scenarios.

Learning Objectives

After completing this lab, you should be able to:

- Apply formal verification to analyze security properties of cryptographic protocols.
- Understand and use the ProVerif verification tool.
- Design and specify a cryptographic protocol with proper threat modeling.
- Evaluate protocol security properties such as secrecy, authentication, and forward secrecy.

Background

Formal verification tools like ProVerif allow protocol designers to mathematically verify security properties. In the context of a TLS-like protocol:

- ProVerif models attackers who have complete control over the communication network.
- Security properties can be precisely defined and verified, such as confidentiality of session keys.
- Verification is performed automatically by exploring all possible protocol executions.
- ProVerif uses symbolic cryptography to reason about cryptographic primitives.

Requirements

Your TLS-like protocol must implement the following core functionality:

1. **Protocol Initialization:**
    - Design a secure handshake procedure between client and server.
    - Incorporate key exchange mechanisms (e.g., Diffie-Hellman).
    - Implement proper authentication through digital signatures or certificates.
2. **Key Exchange:**
    - Establish secure session keys between client and server.
    - Ensure forward secrecy for session communications.
    - Protect against Man-in-the-Middle attacks.
3. **Secure Communication:**

- Design mechanisms for encrypting and authenticating messages.
- Implement protection against replay and reordering attacks.
- Ensure secure session termination.

4. **Formal Specification:**
    - Model the protocol in ProVerif's applied pi calculus.
    - Define security properties to be verified.
    - Design appropriate queries to check security properties.
    - Include proper protocol termination and error handling.

Implementation Guidelines

Step 1: Design

Begin by creating a threat model for your TLS-like protocol. Consider:

- Who are the attackers? (Network adversaries, malicious endpoints)
- What assets are you protecting? (Session keys, message confidentiality, authentication)
- What are the attack vectors? (Man-in-the-Middle, replay, downgrade attacks)
- What cryptographic protections will you employ?

Document your design decisions and security assumptions.

Step 2: Protocol Specification

Design your protocol using formal notation:

- Define message formats and cryptographic operations.
- Specify the exact sequence of messages exchanged.
- Define the security properties you expect your protocol to satisfy.

Step 3: ProVerif Modeling

Implement your protocol in ProVerif:

- Model cryptographic primitives using ProVerif's type system.
- Define processes for client and server roles.
- Formalize security properties as queries.
- Set up the attacker model in ProVerif.

Step 4: Verification

Verify your protocol's security properties:

- Run ProVerif to check for secrecy violations.
- Verify authentication properties.
- Test for resistance against replay attacks.
- Verify forward secrecy.

Step 5: Protocol Refinement

Improve your protocol based on verification results:

- Address any vulnerabilities discovered.
- Optimize the protocol if possible.
- Document changes and their justifications.

Deliverables

Submit the following:

1. Protocol specification including:
    - Formal description of your TLS-like protocol.
    - Message sequence diagrams.
    - Cryptographic primitives used and their roles.
2. ProVerif code implementing your protocol.
3. Design document including:
    - Threat model and security assumptions.
    - Formal security properties being verified.
    - Design decisions and their rationale.
4. Security analysis discussing:
    - Verification results from ProVerif.
    - Strengths of your protocol design.
    - Limitations and potential vulnerabilities.
    - Suggested improvements for a production version.

Evaluation Criteria

Your project will be evaluated based on:

- Correctness of ProVerif specifications and queries.
- Security of the overall protocol design.
- Completeness of required functionality.
- Quality of code and documentation.
- Thoughtfulness of security analysis.

Resources

- The course textbook and materials on cryptographic protocols.
- ProVerif documentation and examples.
- TLS 1.3 specification for reference.
- Research papers on formal verification of security protocols (see course website).

Submission Guidelines

- Submit your ProVerif code as a ZIP archive or through a Git repository.
- Include all documentation in PDF or Markdown format.
- Presentations: Prepare a 10-minute presentation demonstrating your protocol design and verification results.