kb-crypto
Cryptography & Networking Project

Berat Kocak

May 2026

Contents

1. Project Overview

This project implements a client-server communication tool using raw sockets over insecure HTTP, with the goal of rebuilding safe cryptography from first principles.

1.1. Motivation & Learning Objectives

The primary objective of this project is to teach myself the programming language C++ and learn networking in detail. I aim to address the following questions:

1. Cryptography
   - Secure Design: I will design a secure protocol on top of cleartext HTTP, including key exchange, symmetric encryption, and message authentication.
2. Networking
   - Protocol: I want to fully understand the HTTP protocol at the byte level, including request/response formatting, headers, and status codes.
   - Raw Sockets: I am using raw sockets instead of libraries like httplib/cpp-httplib so I can fully understand every part of the protocol.
3. Critical Evaluation
   - Attack Vectors: I will identify potential attack vectors (MITM, replay, timing analysis, traffic analysis, etc.), and harden the protocol based on my findings.
   - Cryptographic strength: I want to use modern cryptographic algorithms.

1.2. Scope

The following areas are in the scope:

1. Implementing a raw socket HTTP server and client in C++ using Win32 API
2. Designing a custom cryptographic protocol layered over HTTP
3. Implementing fundamental crypto primitives (no external libraries like OpenSSL)
4. Documenting attack surfaces and mitigation strategies

The following areas are explicitly out of scope:

1. Optimizations for performance or scalability
2. Formal cryptographic verification

2. The problem

A worker must leave headquarters to investigate factory issues in a country where the government controls all network traffic. HTTPS is blocked, data must flow in cleartext, and authorities inspect incoming travelers for credentials or keys. We need a solution for secure data transmission under these constraints.

2.1. Phase 1: Initial Setup

This phase occurs in a trusted network, like the company headquarters. Data can flow securely even when unencrypted. The client registers by uploading credentials, which the server stores securely. The worker then departs for the other country.

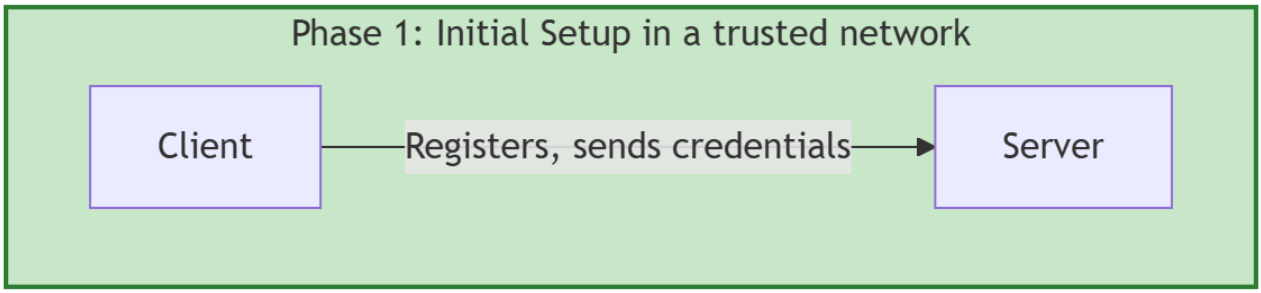

Figure 1: Simplified flowchart of the first phase

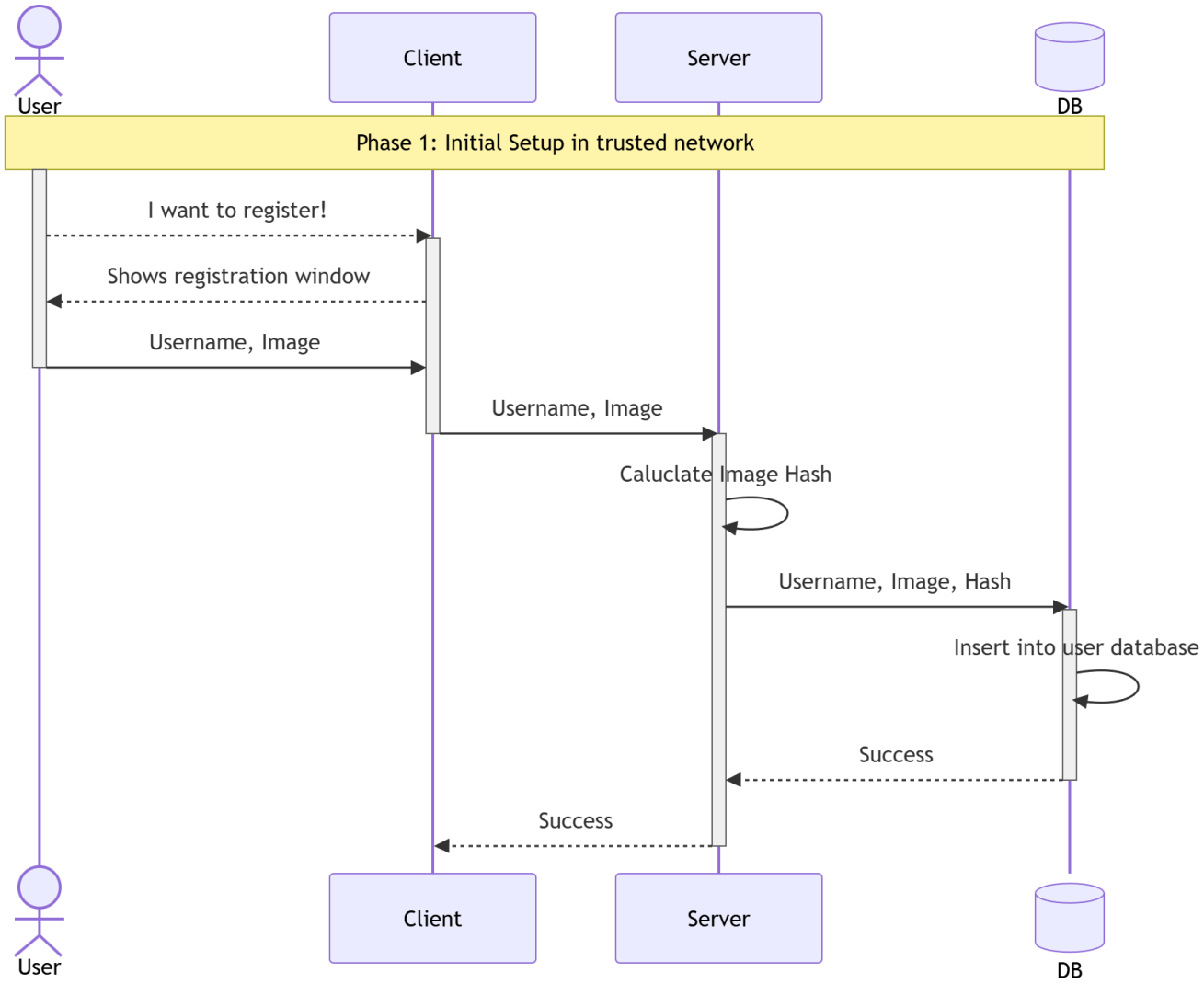

Figure 2: Non-technical sequence diagram of phase one

2.2. Phase 2: Continuing Usage

At the airport, authorities screen the worker for credentials like keys or passwords. The worker carries nothing suspicious and is cleared. They proceed to the factory, investigate issues, and notify headquarters using the communication tool.

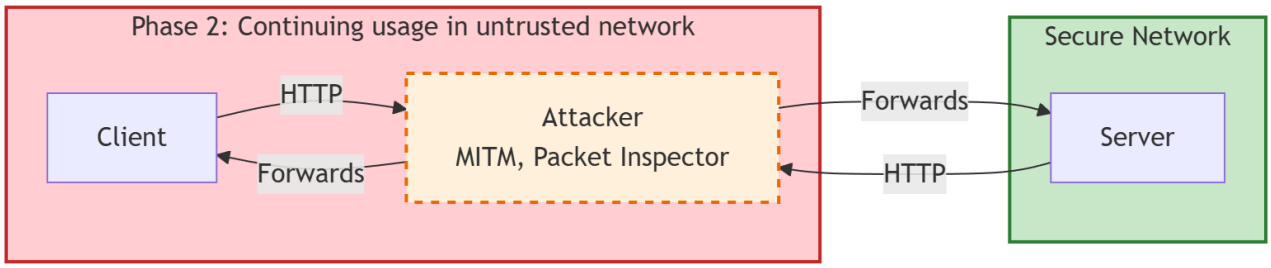

Figure 3: Simplified flowchart of the second phase

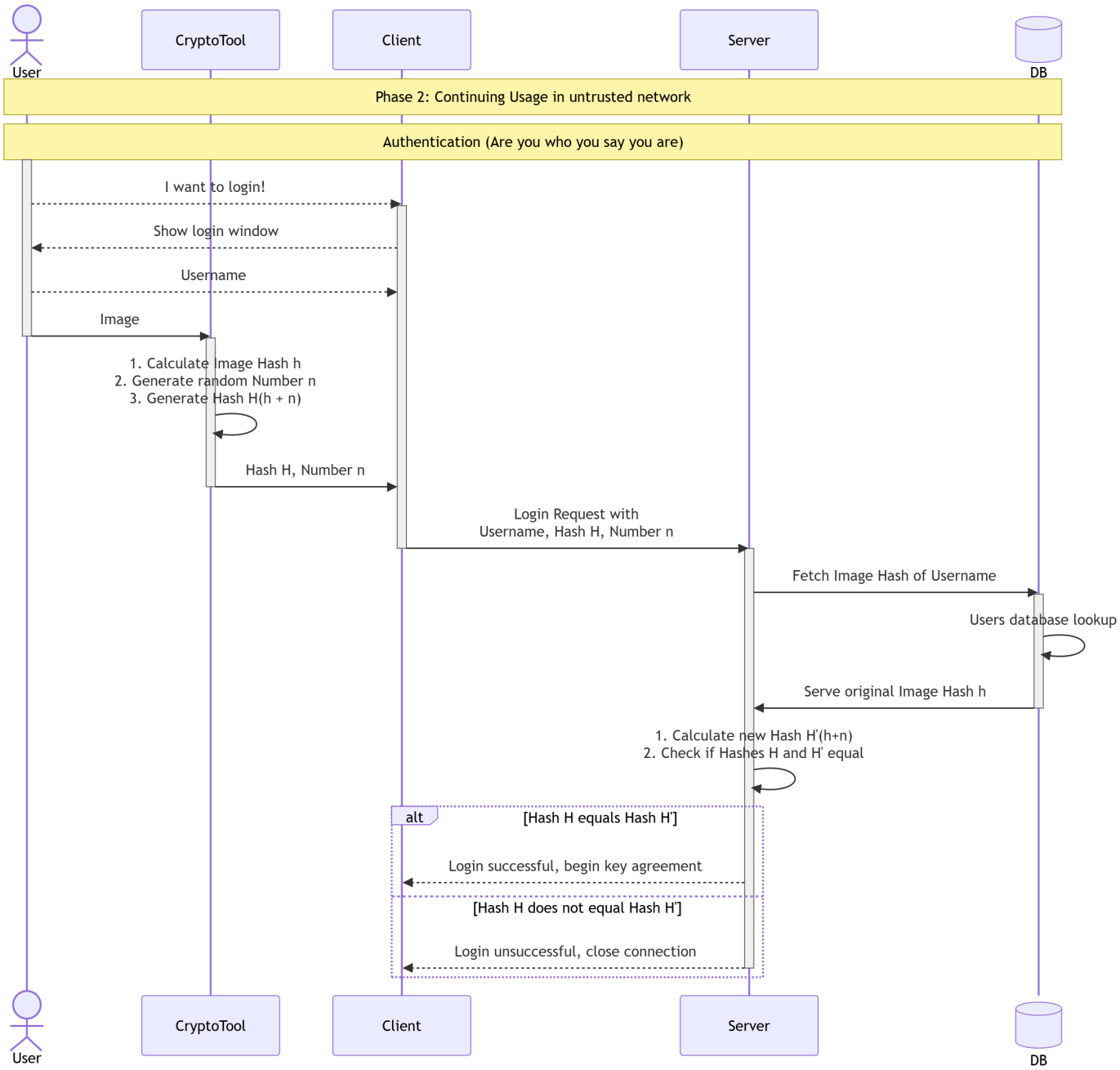

Figure 4: Non-technical sequence diagram of the authentication in phase two

2.3. Implementation Details

2.3.1. Registration (Phase 1)

2.3.2. Credential Storage (Phase 1)

2.3.3. Authentification (Phase 2)

2.3.4. Key agreement (Phase 2)

2.3.5. Transmission (Phase 2)

2.3.6. Logging (Phase 2)

3. Evaluation and Next Steps

3.1. Reflection on Initial Goals

3.2. Future Improvements