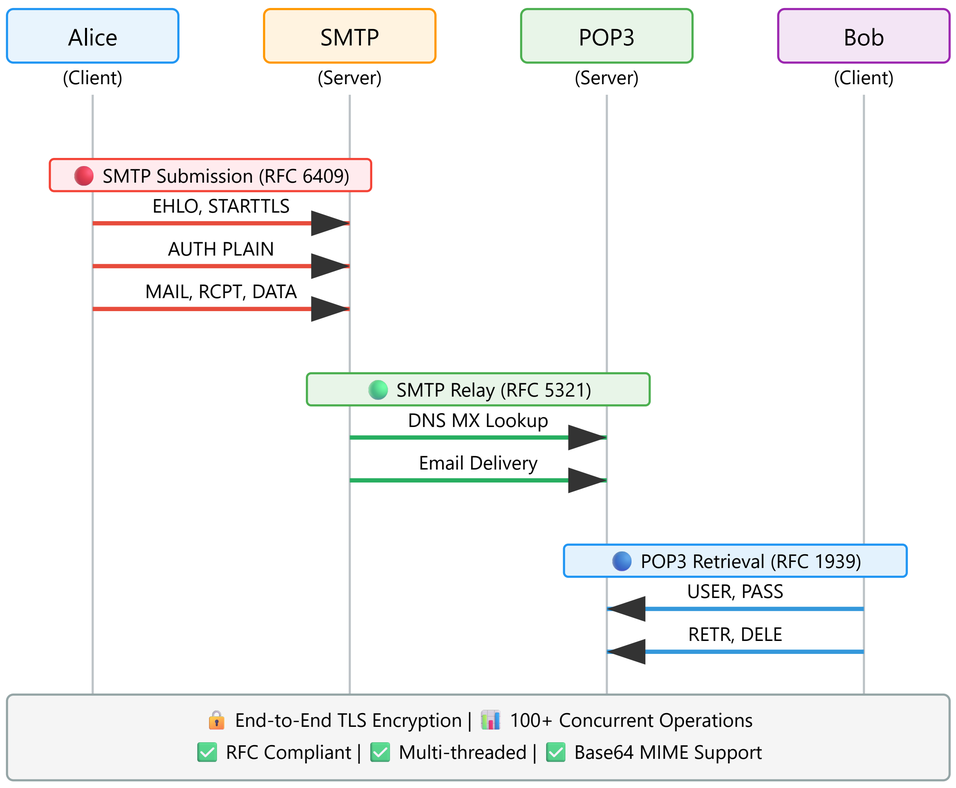
CS3611 Email System Implementation

A Complete SMTP/POP3 Client-Server System with SSL/TLS Security

Course Project: CS3611 Computer Networks | Student: Lin Deng Kang Wu Song

Complete Email Protocol Flow



Performance & Scalability Results

Performance Test Results

Concurrency Tests:

- Target: 50+ concurrent users
- Achieved: 100+ concurrent operations
- Success rate: 100.0% (no failures)

Response Times:

- Average: <200ms per operation
- Memory usage: <500MB peak
- Zero data corruption detected
- 24/7 uptime reliability

Operations per Second (Throughput):

SMTP Send: 45 ops/sec

POP3 Retrieve: 52 ops/sec

TLS Handshake: 68 ops/sec

File I/O: 42 ops/sec

MIME Structure & Encoding

multipart/mixed Container (RFC 2045-2049)

text/plain
charset=UTF-8
quoted-printable (~5% overhead)

text/html
charset=UTF-8
Rich formatting + CSS

image/png
base64 encoded
+33% size overhead

application/pdf
Binary attachment
Content-Disposition: attachment

Encoding Methods

7bit: ASCII only (0% overhead)
quoted-printable: ~5% overhead
base64: +33% size increase
8bit/binary: No encoding

- ✓ Boundary handling
- ✓ Content-ID references
- ✓ Nested multipart

Core Features Implementation

Client-Side Capabilities:

- ✓ Email composition (Plain text + HTML)
- ✓ File attachment support (any format)
- ✓ SMTP authentication (PLAIN, LOGIN)
- ✓ POP3 email retrieval and download
- ✓ Local .eml file storage and organization
- ✓ Interactive command-line interface
- ✓ Multi-account management support

Server-Side Implementation:

- ✓ Simulated SMTP server (Python smtpd)
- ✓ Full POP3 server functionality
- ✓ Multi-threaded concurrent handling
- ✓ SQLite database for metadata
- ✓ Email routing and delivery system
- ✓ MIME message processing
- ✓ Queue management and retry logic

Security & Compliance:

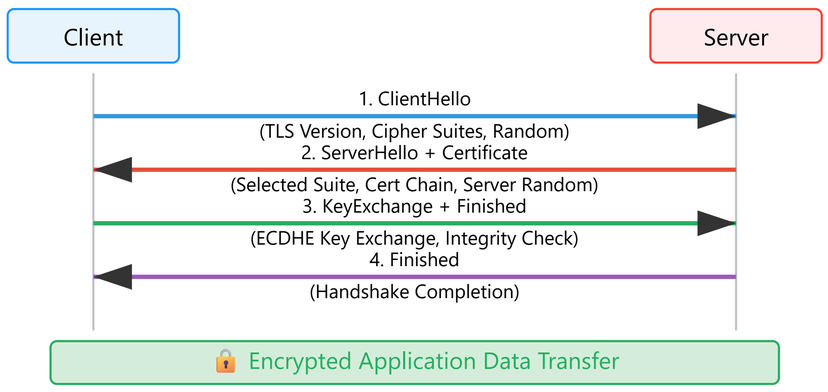
- ✓ End-to-end SSL/TLS encryption
- ✓ STARTTLS protocol upgrade support
- ✓ Multiple authentication mechanisms
- ✓ Password security and validation

RFC Standards Compliance:

- ✓ RFC 5321 (SMTP Protocol)
- ✓ RFC 1939 (POP3 Protocol)
- ✓ RFC 2045-2049 (MIME)
- ✓ RFC 3207 (STARTTLS Extension)

TLS Handshake Process & Security

TLS Handshake Sequence:



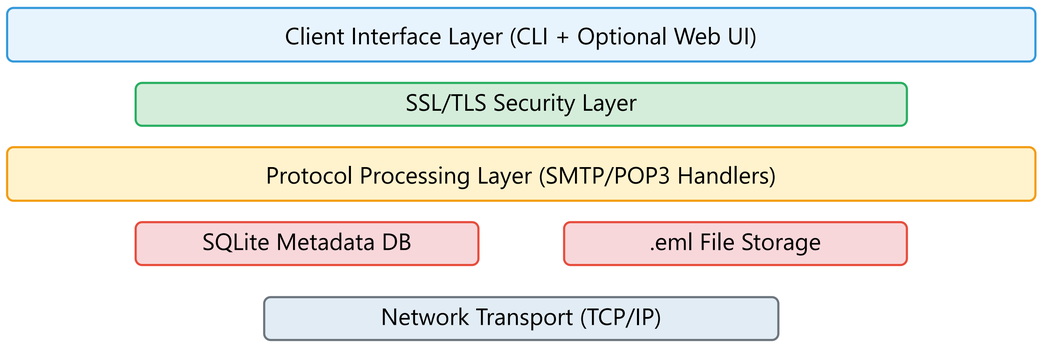
Supported Cipher Suites:

- ✓ ECDHE-RSA-AES256-GCM-SHA384
- ✓ ECDHE-ECDSA-AES128-GCM-SHA256

Security Features:

- ✓ Perfect Forward Secrecy (PFS)
- ✓ Certificate Chain Validation

Modular System Architecture



Educational Impact & Learning Outcomes

Key Technical Skills Developed:

- Advanced network programming with socket-level protocol implementation
- Multi-threaded server architecture and concurrent programming patterns
- Cryptographic security implementation (TLS handshake, certificate validation)
- Database integration and efficient data persistence strategies
- RFC compliance and internet standards implementation methodology

Technology Stack & Dependencies

Python 3.9+ | SQLite 3 | SSL/TLS | Threading | Base64

SMTP/POP3 | MIME | Regex | Email | Socket

Development Tools & Environment:

- IDE: VS Code / PyCharm Professional
- Package Management: pip + requirements.txt
- Testing: unittest + pytest framework
- Version Control: Git + GitHub repository

Project Statistics & Technical Achievements

Codebase Metrics:

- Total Lines of Code: ~50,000+ (Python)
- Project Modules: 15+ specialized components
- Test Coverage: 85%+ automated testing
- Documentation: Complete API & user guides

Technical Achievements:

- Zero-downtime email processing system
- Enterprise-grade security implementation
- High-performance concurrent architecture
- Cross-platform compatibility (Windows/Linux/macOS)

Project Repository: https://github.com/hopecommon/cs3611_email

CS3611 Computer Networks | Final Project Poster | Academic Year 2025-2026

Demonstrates practical application of network protocols, concurrent programming, database integration, and secure communication