🔐 SSL/TLS: Secure Communication Protocols - Technical Writeup

Overview

SSL (Secure Sockets Layer) and **TLS (Transport Layer Security)** are cryptographic protocols designed to provide **secure communication** over a network, typically the internet. TLS is the modern, secure version; SSL is deprecated.

🧱 Architecture Stack

```
Application Layer: HTTPS, FTPS, SMTPS
⬇
TLS Layer: Encryption/Decryption, Authentication
⬇
Transport Layer: TCP
⬇
Network Layer: IP
```

🤝 TLS Handshake Protocol

The **TLS handshake** establishes a secure session between the client and server. It negotiates cryptographic parameters and securely shares symmetric keys.

🔍 TLS 1.2 Handshake (Detailed)

1. **ClientHello**
2. Client sends:
    - TLS version
    - Random value (client_random)
    - List of supported cipher suites

    - Session ID (if resuming)

3. **ServerHello**
4. Server responds with:
    - Chosen TLS version and cipher suite
    - Random value (server_random)
    - Server certificate
    - Optional: ServerKeyExchange (if using DHE/ECDHE)
5. **Certificate Verification**
6. Client verifies the server’s certificate using the CA chain.
7. **Key Exchange**
8. **RSA**: Client encrypts a premaster secret using the server's public key.
9. **ECDHE**: Both parties exchange ephemeral keys and derive a shared secret.
10. **Session Key Derivation**
11. Both sides generate symmetric keys from premaster secret + randoms.
12. **ChangeCipherSpec & Finished**
13. Both sides send a "ChangeCipherSpec" message and switch to encrypted communication.
14. "Finished" messages are encrypted and confirm handshake success.

🔒 Cryptographic Components

Component	**Algorithms**
Key Exchange	ECDHE, DHE, (RSA in TLS 1.2 only)

Component	Algorithms
Authentication	RSA, ECDSA
Encryption	AES-GCM, ChaCha20-Poly1305
Integrity	HMAC-SHA256, AEAD (TLS 1.3)

Best Practices

1. **Disable SSL, TLS 1.0, and 1.1**
2. Prefer **TLS 1.3**
3. Use strong ciphers (AES-GCM, ChaCha20)
4. Enable **forward secrecy** (ECDHE)
5. Use certificates from trusted CAs (e.g., Let's Encrypt)
6. Rotate certificates regularly
7. Implement **HSTS** for HTTPS-only communication

References

- RFC 8446 - TLS 1.3 Specification
- SSL Labs Test Tool
- Mozilla TLS Config Generator
- OpenSSL Docs