Secure File Transfer Protocol - Security Flow

1. System Initialization

1.1 Generate RSA key pair (2048-bit)

1.2 Generate HMAC key

1.3 Create data and log directories

1.4 Configure logging security handlers

2. User Authentication

2.1 Connect to server via TCP socket

2.2 Send username to server

2.3 Transmit public key to server

2.4 Receive online user list with public keys

3. Initiating File Transfer

3.1 Select recipient and file

3.2 Generate AES symmetric key

3.3 Encrypt symmetric key with recipient public key (RSA)

3.4 Encrypt HMAC key with recipient's public key (RSA)

3.5 Create FileTransferRequest

3.6 Send request to server (UUID assigned

4. Server Processing

4.1 Validate sender and recipient

4.2 Generate unique transfer ID

4.3 Forward request to recipient

5. File Preparation

5.1 Divide file into 4KB chunks

5.2 Generate random IV for each chunk

5.3 Encrypt chunks with AES-CBC

5.4 Update transfer history

6. Chunk Security Processing

6.1 Generate timestamp for each chunk

6.2 Generate unique nonce (UUID)

6.3 Calculate HMAC over: (encrypted data + IV + timestamp + nonce)

6.4 Create SecureMessage with: (encrypted data, MAC, IV, timestamp, nonce)

6.5 Optional: Create SignedSecureMessage

7. Chunk Transmission

7.1 Send each chunk with metadata

7.2 Server forwards chunks to recipient

7.3 Update transfer progress

7.4 Support pause/resume functionality

8. Chunk Reception

8.1 Decrypt symmetric key with private key

8.2 Decrypt HMAC key with private key

8.3 Verify HMAC of each chunk

8.4 Verify timestamp is recent

8.5 Track nonces to prevent replay attacks

8.6 Reject chunks with failed verification

9. File Decryption and Assembly

9.1 Decrypt each chunk with AES-CBC using IV + symmetric key

9.2 Assemble chunks in correct order

9.3 Write to file in downloads directory

9.4 Update transfer history

10. Completion and Notification

10.1 Sender notifies completion

10.2 Server relays completion status

10.3 Update sender's transfer history

10.4 Update recipient's transfer history

11. Logging and Auditing

11.1 Log security operations

11.2 Log transfer events

11.3 Log cryptographic operations

11.4 Store logs in dedicated directory

Security Protocol Elements

Confidentiality: AES-256-CBC encryption

Integrity: HMAC-SHA256 verification

Authentication: RSA digital signatures

Anti-replay: Timestamps and nonces

Key Security Protocol Highlights

Confidentiality	End-to-end encryption with AES-CBC using unique symmetric key per transfer
Integrity	HMAC-SHA256 verification of all data chunks to detect tampering
Secure Logging	Comprehensive security event logging in dedicated logs directory
Availability	DoS Protection Mechanisms
Authentication	RSA key pairs for asymmetric encryption and signatures
Anti-replay	Timestamps and unique nonces to prevent message replay attacks
Key Management	Public/private key pairs for secure key exchange

DoS Protection Mechanisms

DoS Protection Implementation Flow

Connection Limiting
D1.1 Check connection count per IP
D1.2 Enforce MAX_CONNECTIONS_PER_IP
D1.3 Log excessive connection attempts
D1.4 Implement connection timeouts

Request Rate Limiting
D2.1 Track requests per minute per IP
D2.2 Enforce MAX_REQUESTS_PER_MINUTE (60)
D2.3 Exempt file chunk transfers from limits
D2.4 Return "Rate limit exceeded" error

IP Blacklisting System
D3.1 Record suspicious IP activity
D3.2 Blacklist IPs after threshold violatio
D3.3 Apply temporary blacklist with expiratio
D3.4 Auto-cleanup expired blacklist entries

Login Attempt Limiting
D4.1 Track login attempts per hour per IP
D4.2 Enforce MAX_LOGIN_ATTEMPTS_PER_HOUR
D4.3 Auto-blacklist after exceeding threshold

Bandwidth Control
D5.1 Track bandwidth usage per connection
D5.2 Enforce BANDWIDTH_LIMIT_BYTES_PER_SEC
D5.3 Sliding window for bandwidth calculation

Active DoS Monitoring
D6.1 Regular security metric evaluation
D6.2 Threshold-based threat level detect
D6.3 Automated defense activation
D6.4 Administrative alerting