Secure Key Exchange and Tamper-Resistant Communication Framework

Presentation: **Shaba Altaf Shaon, Hasan Md Imran, and Ratun Rahman**
Date: 10 November, 2025

Objective

- Develop and demonstrate a secure communication framework that enables two peers to authenticate each other, exchange encryption keys securely, and detect any data tampering or replay attacks.

Key Features:

- A trusted registry helps devices securely find and verify each other on the network.
- Both peers prove their identities to one another using digital signatures
- Temporary session keys are created for every connection, ensuring past data stays safe even if a key is later compromised.
- All messages are encrypted and checked for integrity, keeping data confidential and unaltered.
- The system can instantly detect and block any tampering, protecting communication in real time.

System Architecture

Registry Server:

- Acts as a trusted introducer maintaining verified peer information.
- Prevents directory poisoning through signed registry replies.

Client Nodes (Initiator & Responder):

- Generate long-term and temporary key pairs.
- Exchange signed handshakes to verify identities.
- Establish shared session keys for encrypted communication.

Secure Channel Design

Encryption Layer:

- Session keys derived securely from the handshake.
- Each message encrypted and authenticated for integrity.
- Directional counters ensure message ordering and freshness.

Security Enhancements:

- Replay and tamper detection mechanisms built into message flow.
- Automatic termination on integrity failure to maintain safety.

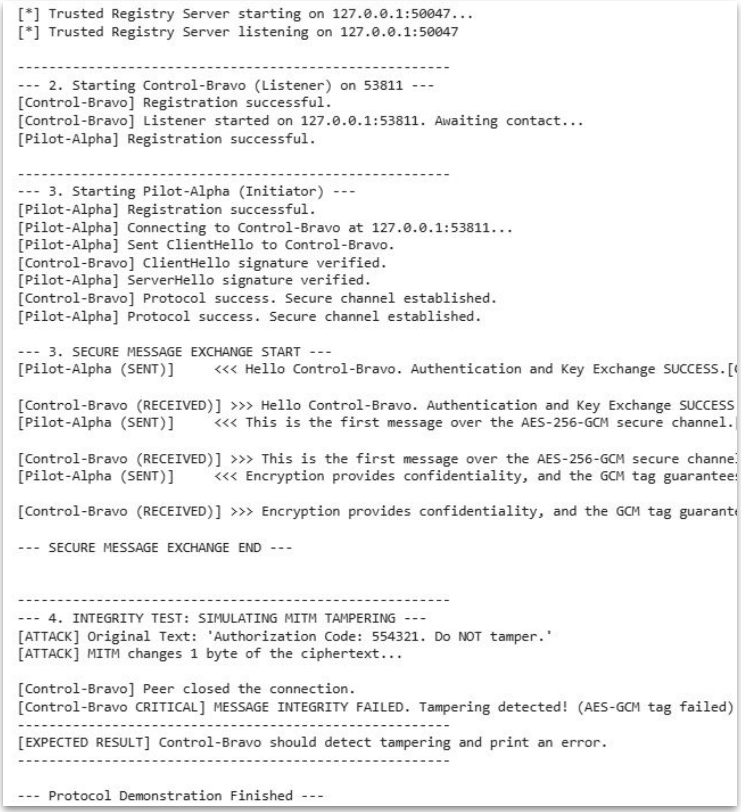
Demonstration & Results

Steps Demonstrated:

- Registry setup and cliSteps Demonstrated:
- Registry setup and client registration.
- Mutual authentication handshake.
- Secure message exchange between peers.
- Simulated tampering attempt (MITM test) detected in real time.

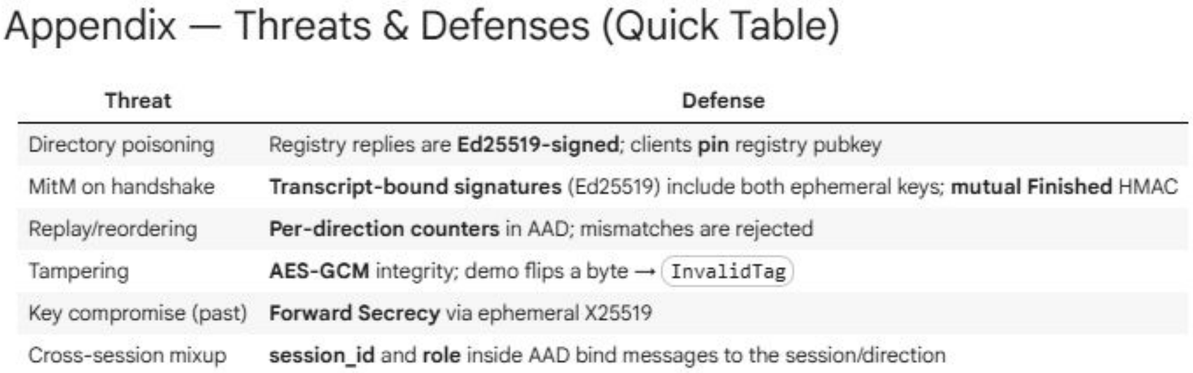
Outcome:

- Successful setup of a secure, authenticated channel.
- Tampering correctly identified and blocked.ent registration.



```
[*] Trusted Registry Server starting on 127.0.0.1:50047...
[*] Trusted Registry Server listening on 127.0.0.1:50047

-------------------------------------------------------
--- 2. Starting Control-Bravo (Listener) on 53811 ---
[Control-Bravo] Registration successful.
[Control-Bravo] Listener started on 127.0.0.1:53811. Awaiting contact...
[Pilot-Alpha] Registration successful.

-------------------------------------------------------
--- 3. Starting Pilot-Alpha (Initiator) ---
[Pilot-Alpha] Registration successful.
[Pilot-Alpha] Connecting to Control-Bravo at 127.0.0.1:53811...
[Pilot-Alpha] Sent ClientHello to Control-Bravo.
[Control-Bravo] ClientHello signature verified.
[Pilot-Alpha] ServerHello signature verified.
[Control-Bravo] Protocol success. Secure channel established.
[Pilot-Alpha] Protocol success. Secure channel established.

--- 3. SECURE MESSAGE EXCHANGE START ---
[Pilot-Alpha (SENT)]     <<< Hello Control-Bravo. Authentication and Key Exchange SUCCESS.[

[Control-Bravo (RECEIVED)] >>> Hello Control-Bravo. Authentication and Key Exchange SUCCESS
[Pilot-Alpha (SENT)]     <<< This is the first message over the AES-256-GCM secure channel.

[Control-Bravo (RECEIVED)] >>> This is the first message over the AES-256-GCM secure channe
[Pilot-Alpha (SENT)]     <<< Encryption provides confidentiality, and the GCM tag guarantee

[Control-Bravo (RECEIVED)] >>> Encryption provides confidentiality, and the GCM tag guarant

--- SECURE MESSAGE EXCHANGE END ---


-------------------------------------------------------
--- 4. INTEGRITY TEST: SIMULATING MITM TAMPERING ---
[ATTACK] Original Text: 'Authorization Code: 554321. Do NOT tamper.'
[ATTACK] MITM changes 1 byte of the ciphertext...

[Control-Bravo] Peer closed the connection.
[Control-Bravo CRITICAL] MESSAGE INTEGRITY FAILED. Tampering detected! (AES-GCM tag failed)
-------------------------------------------------------
[EXPECTED RESULT] Control-Bravo should detect tampering and print an error.
-------------------------------------------------------

--- Protocol Demonstration Finished ---
```

Conclusions



Appendix — Threats & Defenses (Quick Table)

Threat	Defense
Directory poisoning	Registry replies are **Ed25519-signed**; clients **pin** registry pubkey
MitM on handshake	**Transcript-bound signatures** (Ed25519) include both ephemeral keys; **mutual Finished** HMAC
Replay/reordering	**Per-direction counters** in AAD; mismatches are rejected
Tampering	**AES-GCM** integrity; demo flips a byte → `InvalidTag`
Key compromise (past)	**Forward Secrecy** via ephemeral X25519
Cross-session mixup	**session_id** and **role** inside AAD bind messages to the session/direction

Security Goal	Achieved?	Evidence
Confidentiality	✅	Messages encrypted with AES-256-GCM
Authentication	✅	Ed25519 signatures verified both sides
Message Integrity	✅	MITM tampering detected via AEAD tag