NETWORK AND COMPUTER SECURITY COURSE (SIRS) - 2024/2025

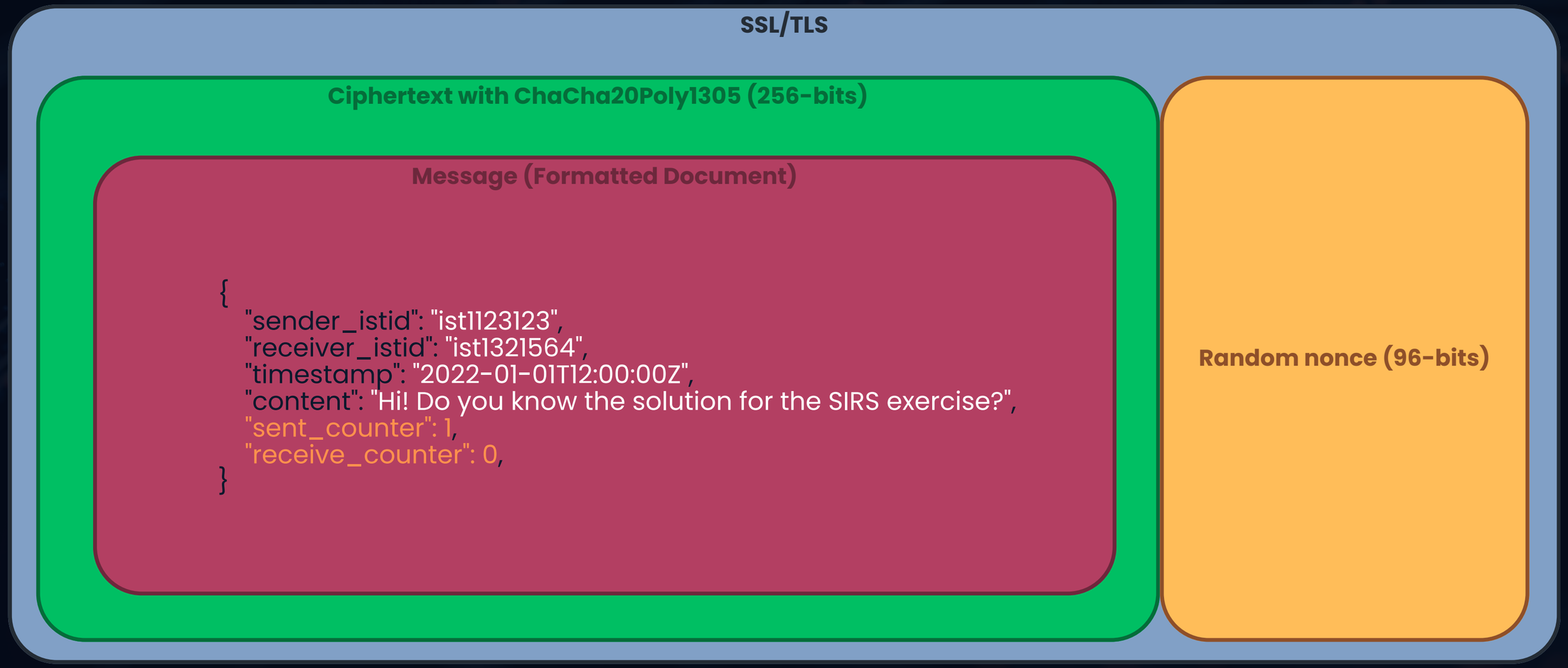
MESSAGIST

Group A07
ist1112191 - Xiting Wang
ist1112269 - Laura Cunha
ist1112270 - Rodrigo Correia

github

SECURE DOCUMENT FORMAT

Design



SECURE DOCUMENT FORMAT

RustCrypto library

CRIPTOLIB

Funcionalities

```
$_
  › help
  › protect
  › check
  › unprotect
  › hash-password
  › verify-password
  › gen-secret-key
  › gen-rsa-keypair
  › encrypt-key-with-pub-key
  › decrypt-key-with-priv-key
```

Freshness

- Communications occur over SSL/TLS
- Nonce sent with the formatted document

256-bit key

- Secret key is generated for each message sent
- Stored for long term on database

Random nonce

- Unique cipher

No Signed

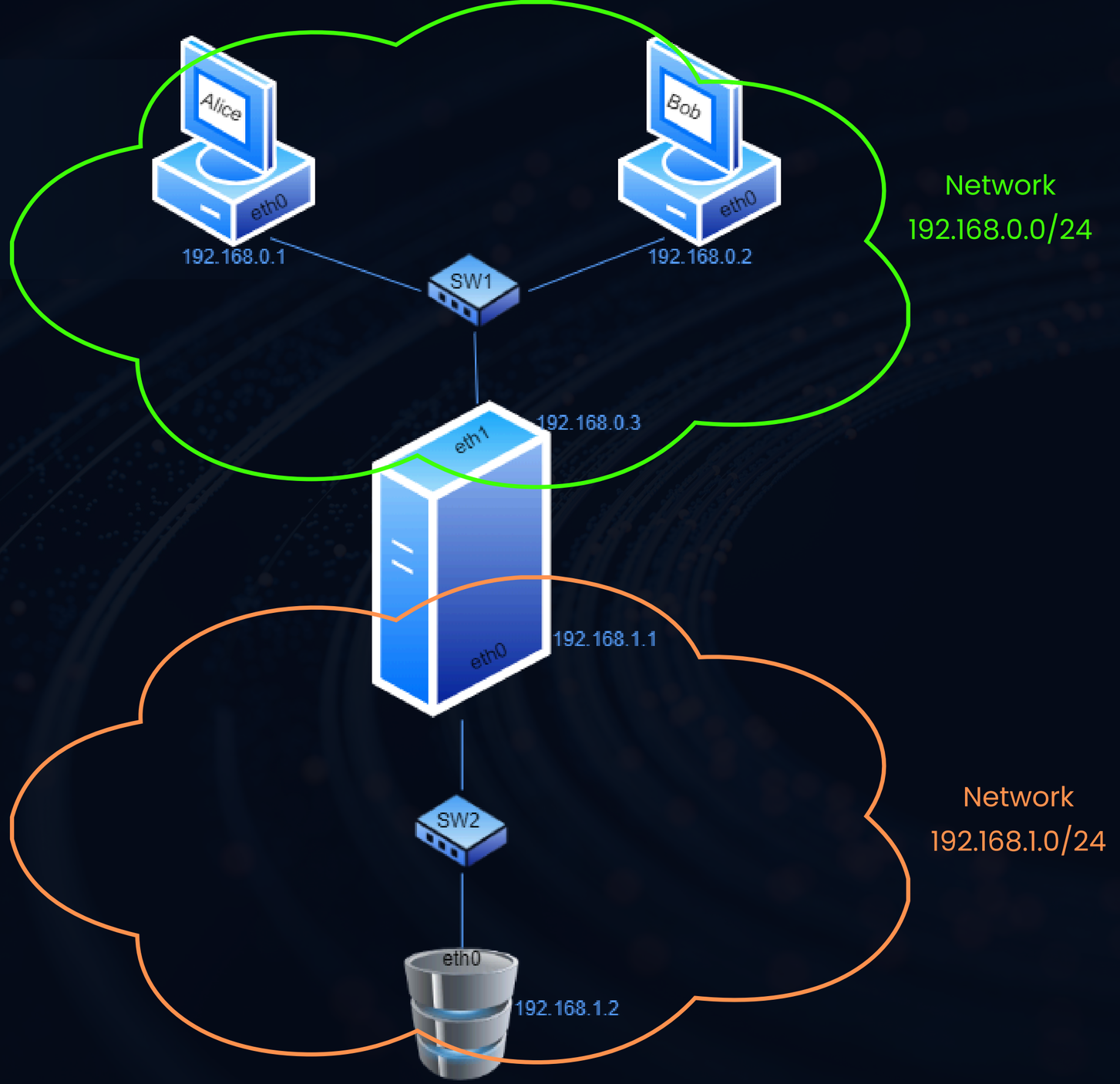
- Complexity and computational cost
- Unnecessary additional properties (non-repudiation, non-forgery, non-reusable)

Algorithms

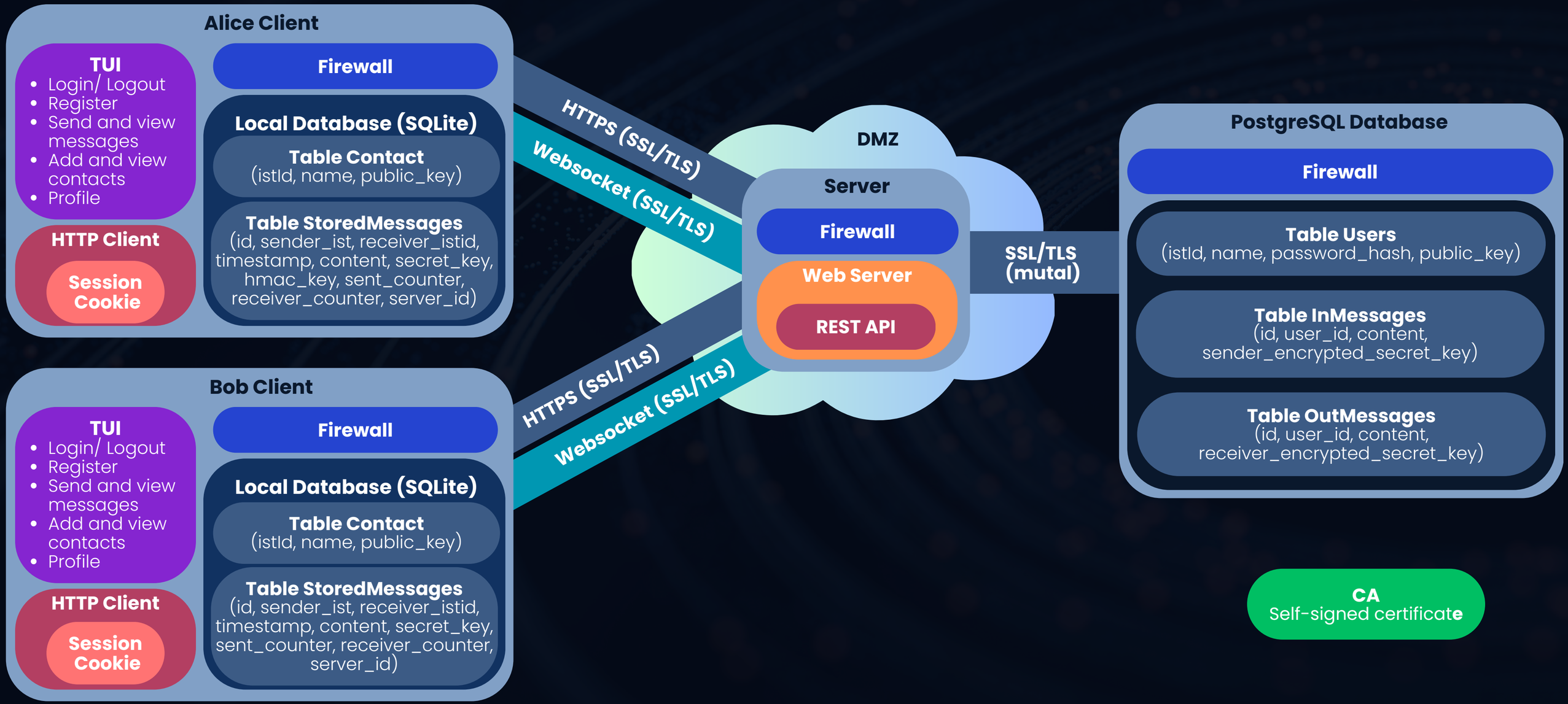
ChaCha20-Poly1305

- Reliable Rust library
- Stream cipher with a hash function
- Confidentiality, integrity, and authenticity in a single operation
- No padding
- Lower latency compared to block ciphers
- High-performance parallelism

NETWORKS AND MACHINES



BUILT INFRAESTRUCTURE



SECURIRY MECHANISMS

SSL/TLS

- **Confidentiality, integrity, authenticity**
- Robust face **man-in-the-middle, replay** (freshness), interception of passwords, downgrade attacks, etc
- Connection **authentication**:
  - Server <-> Database
  - Client <- Server

DBs

- All databases have **sanitization** inputs

User Authentication

- **Password**-based login
- Cipher **session cookie**
- **Local SQL database's file encrypted** with KDF from the login's password

Firewall

- All VMS have **firewalls** that allow only essential traffic

KEY DISTRIBUTION

CA

Label:

Private Key

Sign

Certificate

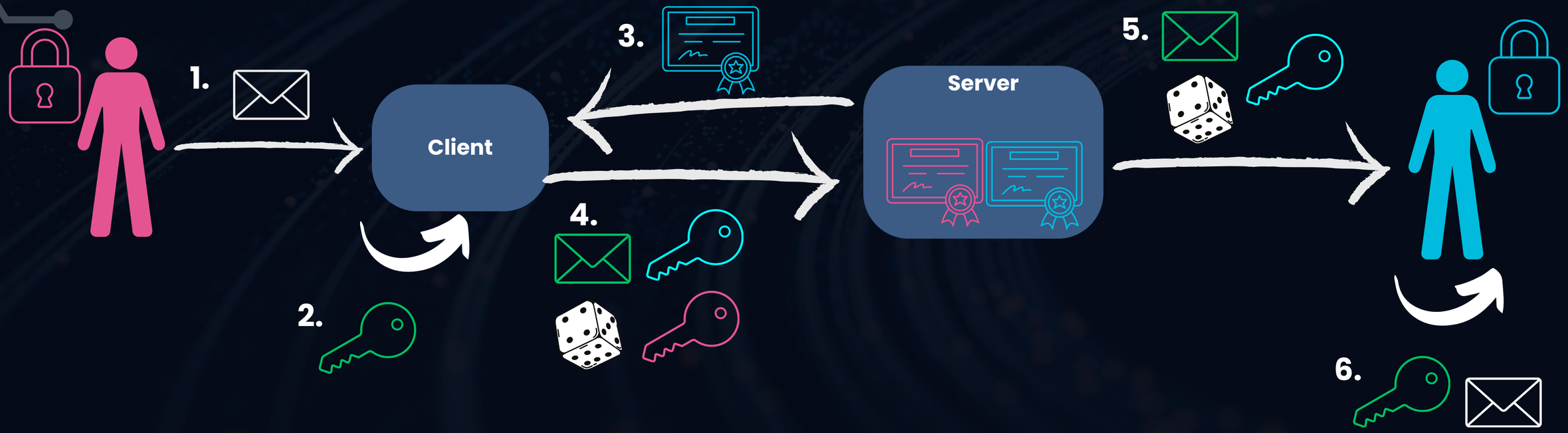
KEY DISTRIBUTION

Register

Label:

KEY DISTRIBUTION

Send Message



Label:

Private Key Public Key Message Secret Key Nonce

CHALLENGES

PROTECTION NEEDS

SR1: CONFIDENTIALITY

- **ChaCha20**
- Shared **symmetric keys**

SR2: INTEGRITY 1

- **Poly1305 hash function**
- **Integrity** and **authenticity**

SR3: INTEGRITY 2

- Message **counters**
- Prevent **replay** attacks, and **out-of-order** messages
- Checked and maintained by **clients**

SR4: AUTHENTICATION

- **Password**-based login
- **Session cookies** (with expiration time)

CHALLENGES

CHALLENGE A

SRA1: CONFIDENTIALITY

- **End-to-end encryption**
- **Server cannot read** message content nor keys
- **Encrypt the secret key** with the public key of both clients

SRA2: CONFIDENTIALITY

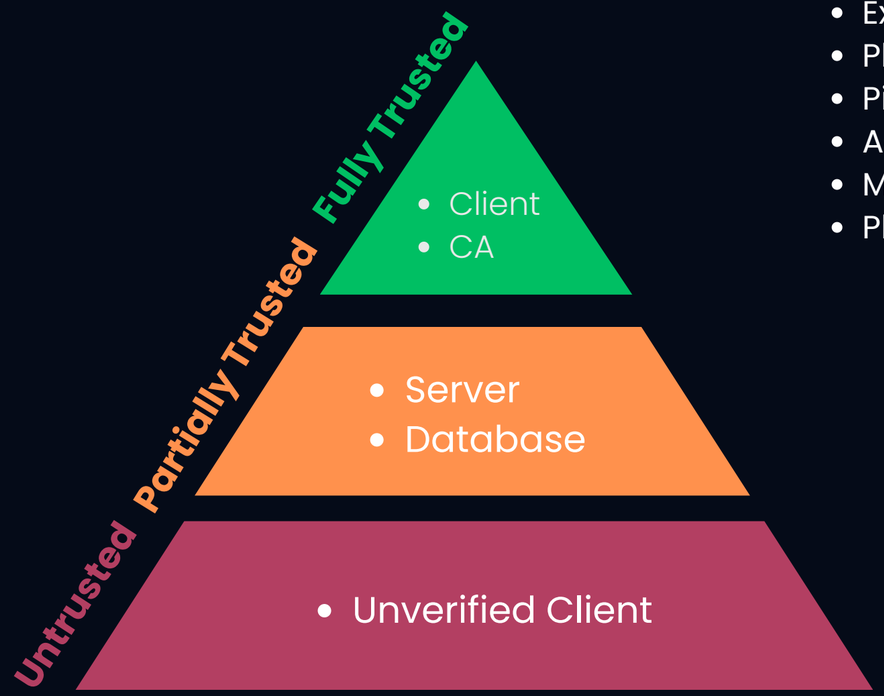
- Add contact by **inserting the public key** manually
- **Side channel** for authentication

SRA3: AVAILABILITY

- Client has always a **backup copy** of his **private key**
- Can request server the message **history**
- **Private key** used to decrypt the symmetric keys so it can access the message's content

ATTACKER MODEL

TRUST LEVELS



ATTACKER CAPABILITIES

- Brute force password
- Exploitation Vulnerabilities
- Phishing
- Ping Flood, DDoS, DoS, Spamming Attacks
- Access to Encrypted Data on client's local database
- Modify or Delete Data
- Physical Device Theft

ATTACKER LIMITATIONS

- MITM
- Intercepting Connections
- Memory Exploits
- Replay Attacks
- Encrypted and Secure Communication
- Unable to send valid messages
- Firewalls
- Database sanitizing and limited privileges

CONCLUSIONS

MAIN RESULTS

End-to-end Encryption

Data Recovery Mechanism

Local Data Security

User Authentication

Secure Code

Key Management

Secure Communication

CONCLUSIONS

Security-Enhancing Technologies

- Authorized access only from the Técnico network

Improved Key Management

- PKI
- Automated Key Revocation List

FUTURE ENCHANCEMENTS

Robust Gateway Architecture

- Simple Gateway Dual-Homed
- DMZ, Firewall, and Routing

IDS (Intrusion Detection System)

- Notify port scanning and flood attacks

2FA Authentication

DEMO

QUESTIONS