Anonymity System

Jacopo Orlandini, 286416

Abstract

The objective of this project is to create a system for anonymizing communication between a client and a server. The Anonymity system may consist of one or more intermediate nodes that properly encrypt and decrypt messages exchange between the two endpoints. The server may be unaware of the anonymity system. Both key exchange and communication confidentiality should be provided.

Work Experience

Il progetto viene sviluppato come sistema a tre nodi: client, proxy e server.
Come strumenti di crittografia viene usato:

- RSA come crittografia asimmetrica
- AES come crittografia simmetrica.

La chiave simmetrica viene inizializzata nei client e server attraverso RSA con:

- dimensioni di 16 byte per la chiave = Network_security
- Iv = 'This is an IV456'
- Cipher Feedback (CFB).

Il sistema è in grado di acquisire da tastiera un messaggio per poi inoltrarlo attraverso il sistema di anonimizzazione al server con conseguente risposta di quest'ultimo.
Il sistema usa socket TCP e non di tipo UDP (datagramm) e gira in localhost : 127.0.0.1
Il client si collega sulla porta 10000.
Il proxy si collega sulla porta 20000.

Workflow e Setup

Moduli usati nel progetto

`modulo` **`Crypto:`** pip install pycrypto

- PublicKey import RSA
- Cipher import AES
- random

`modulo` **`socket`**

Per avviare il sistema

Lanciare proxy.py -> server.py -> client.py

Il ciclo parte dal client che invia un messaggio criptato AES e viene consegnato al server che decifra e risponde al client.

WorkFlow

Il lato PROXY si suddivide in due sezioni: nella prima fase si condividono le chiavi pubbliche fra client e proxy, per poi inviare inizializzare la chiave simmetrica nel client attraverso l'invio dei dati dentro il proxy con chiave asimmetrica.

La seconda fase consiste nello stesso procedimento sopracitato ma con proxy e server.

Una volta completato il setup dei nodi avviene la comunicazione vera e propria di scambio di messaggi attraverso il sistema di anonimizzazione.

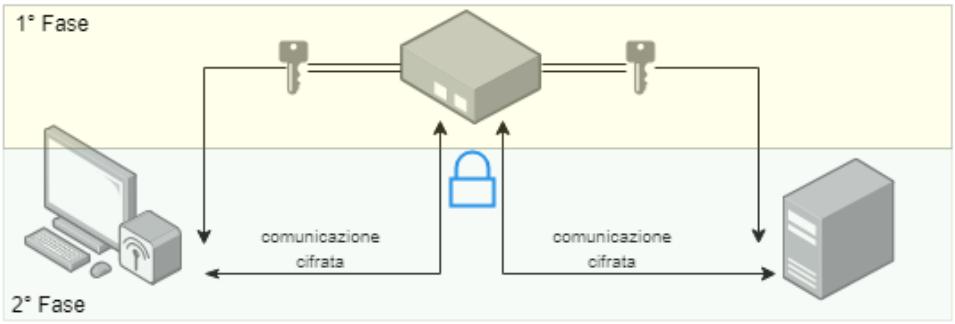
Il client riceve il messaggio testuale da tastiera, lo cifra con la chiave simmetrica AES (che ora è condivisa fra client e server), viene successivamente cifrato con chiave asimmetrica RSA e inviato al PROXY.

Il proxy svolge il ruolo di inoltrare il messaggio al server dopo averlo decifrato dalla chiave pubblica inserita dal client e cifrarlo con la chiave pubblica del server.

Il server decifra il messaggio sia con chiave privata e chiave simmetrica e mostra il messaggio su terminale. A questo punto il lato server prende una stringa da terminale e simmetricamente svolge la stessa operazione per inviare il messaggio al client.

La procedura inizia con il messaggio da lato client e non è possibile mandare il primo messaggio da lato server.

Per chiudere la sessione si deve inviare dal client il messaggio: 'Stop' , il quale permette l'uscita da tutti i cicli e chiude tutte le socket attualmente aperte.



Client

Chiave simmetrica ricevuta dal proxy.

`Chiave pubblica del proxy`

Chiave pubblica e privata del client

Proxy

Chiave pubblica [server, client]
Chiave pubblica e privata del proxy
Dati della chiave simmetrica condivisa

Server

Chiave pubblica del proxy.

Chiave pubblica e privata del server.

Chiave simmetrica ricevuta dal proxy