Consegna S3-L3

Consegna: Crittografia e Firma Digitale

⚜️ **Tag:** #crittografia #python #openssl #sicurezza

Parte 1: Decifrare un messaggio cifrato

- **Messaggio cifrato**: "HSNFRGH"
- Obiettivo: Individuare il testo in chiaro partendo dal messaggio cifrato.

Parte 2: Criptazione e Firma con OpenSSL e Python
Obiettivi dell'esercizio:

1. Generare chiavi RSA.
2. Estrarre la chiave pubblica da una chiave privata.
3. Criptare e decriptare messaggi.
4. Firmare e verificare messaggi.

Strumenti utilizzati:

- OpenSSL per la generazione delle chiavi.
- Libreria `cryptography` in Python.

Comandi per la configurazione:

1. Installazione di OpenSSL:

```
sudo apt update
sudo apt install openssl
```

2. Installazione della libreria Python:

```
sudo apt install python3-pip
pip3 install cryptography
```

Comandi per la generazione delle chiavi:

- Generare chiave privata RSA:

```
openssl genpkey -algorithm RSA -out private_key.pem -
pkeyopt rsa_keygen_bits:2048
```

- Estrarre la chiave pubblica:

```
openssl rsa -pubout -in private_key.pem -out
public_key.pem
```

Parte 3: Implementazione in Python
File `encdec.py`:

- Importa le chiavi private e pubbliche.
- Esegui la criptazione e decriptazione del messaggio.

File `firma.py`:

- Firma digitale del messaggio con la chiave privata.
- Verifica della firma con la chiave pubblica.

Esempio di codice:

- Criptazione del messaggio:

```
encrypted = public_key.encrypt(message.encode(),
padding.PKCS1v15())
```

- Firma digitale:

```
signed = private_key.sign(message.encode(),
padding.PKCS1v15(), hashes.SHA256())
```

Output atteso:

- Base64 della firma.
- Messaggio originale e decriptato.

🔑 **Chiavi:**
crittografia, python, openssl, sicurezza