Makara Ramoabi

20240045

Security

1. Scenario Analysis

In a corporate environment, an employee sends a confidential financial report via email to a manager. To ensure the message has not been altered and truly comes from the sender, the employee digitally signs the email using public-key cryptography. The sender's email client generates a hash of the message and encrypts the hash using the sender's private key. When the manager receives the email, their system decrypts the signature using the sender's public key and compares the result with a newly generated hash of the received message. If both hashes match, the message integrity and authenticity are confirmed. If they do not match, the message may have been tampered with. This process prevents forgery and protects against man-in-the-middle attacks. Digital signatures are widely used in secure email systems, software distribution, and online transactions to ensure authenticity, integrity, and non-repudiation of digital communications.

2. Concept Research

RSA (Rivest–Shamir–Adleman) is a public-key cryptographic algorithm used for secure data transmission. It relies on the mathematical difficulty of factoring large prime numbers. RSA uses two keys: a public key for encryption and a private key for decryption. When someone encrypts data using the recipient's public key, only the matching private key can decrypt it. RSA is also used for digital signatures, where the private key signs data and the public key verifies it. Although slower than symmetric encryption, RSA plays a crucial role in secure communications, including HTTPS, email encryption, and secure key exchange.

3. Tool Practice

Example Command:

```
openssl enc -aes-256-cbc -salt -in report.txt -out report.enc
```

This command encrypts report.txt using AES-256 encryption.

Reflection

Using OpenSSL to encrypt a file helped me understand how cryptography works in practical situations. The command-line interface required me to specify the encryption algorithm and input/output files. After running the command, the original file content became unreadable in the encrypted version, demonstrating confidentiality. I also learned that a password is required to decrypt the file later. This exercise showed how encryption tools protect sensitive data in real-world systems. It reinforced the importance of secure key management and strong passwords when working with encryption technologies in professional environments.

4. Diagram Design

The encryption workflow diagram should begin with the sender creating a plaintext message. The message is then processed by a hashing function (for digital signatures) or encrypted using a public key. The encrypted message travels through an insecure channel, such as the internet. On the receiver's side, the private key is used to decrypt the message or verify the digital signature. The diagram should clearly separate the public key and private key to show their different roles. Arrows indicate the flow of data from sender to receiver. This visual explanation demonstrates confidentiality, integrity, and authentication in a secure communication process.