07/08/2024

Project Report -

Secure Payment with Server Identification Using Schnorr Signature, SM4, and ECDH

Dor Shabat – 316575620

Yuval Rozner - 207756552

Introduction

The aim of this project is to set up a secure payment system that makes sure the sender and receiver are who they claim to be and keeps the data safe. This will involve using three different methods: Schnorr Signature for digital signatures, SM4 for encryption, and Elliptic Curve Diffie-Hellman (ECDH) for securely sharing keys. The project is coded in Python, without relying on high-level cryptographic libraries, so that we can understand the inner workings of cryptography better.

Project Components

Elliptic Curve Diffie-Hellman (ECDH) Key Exchange

- ECDH is used to securely exchange cryptographic keys between two parties over an insecure channel. It leverages elliptic curves to provide enhanced security with smaller key sizes compared to classical Diffie-Hellman.

SM4 Symmetric Encryption

- SM4 is a block cipher standard used in China. It operates on 128-bit blocks and uses a 128-bit key, performing encryption and decryption in 32 rounds with an unbalanced Feistel structure.

Schnorr Digital Signature

- The Schnorr signature scheme ensures the authenticity and integrity of messages, preventing man-in-the-middle attacks and providing non-repudiation.

Project Flow

Our project began with research on the algorithms: SM4 for encryption, ECDH for key exchange, and Schnorr signatures for authentication. We studied the theoretical foundations of each algorithm and created basic simulations to understand their functionalities. Following this, we fully implemented ECDH, SM4, and Schnorr signatures. After integrating these algorithms into a working system, we refined the implementation by optimizing performance and fixing any issues. This structured approach enabled us to effectively combine ECDH, SM4, and Schnorr signatures for a secure payment system.

Obtained Results

The implementation successfully demonstrated the secure exchange of a symmetric key using ECDH, the encryption and decryption of messages using SM4, and the signing and verification of messages using the Schnorr signature scheme. The project ensures data confidentiality, integrity, and authenticity.

Implementation Process

ECDH Key Exchange

- **Key Generation**: Both parties (Alice and Bob) generate their private keys.
- **Public Key Calculation**: Each party calculates their public key by multiplying the generator point on the elliptic curve with their private key.
- **Shared Secret Calculation**: Both parties compute the shared secret by multiplying the received public key with their private key.

SM4 Symmetric Encryption

- **Encryption:** The plaintext is encrypted using the SM4 algorithm with the 128-bit key.
- **Decryption:** The ciphertext is decrypted back to plaintext using the same key in reverse order.

Schnorr Digital Signature

- **Key Generation:** The signer generates a private key and a corresponding public key.
- **Signing:** The signer generates a random nonce, computes a hash of the message and nonce, and generates the signature.
- **Verification:** The verifier uses the signer's public key and the signature to verify the authenticity and integrity of the message.

Conclusions

This project achieved the goal of implementing a secure payment system using fundamental cryptographic principles. The integration of ECDH for secure key exchange, SM4 for encryption, and Schnorr signatures for authentication and integrity provided a comprehensive security solution. Implementing these algorithms from scratch, rather than using high-level libraries, deepened our understanding of their inner workings and the importance of each step in the cryptographic processes. This project highlights the practical application of cryptographic techniques in real-world secure communication systems, demonstrating their effectiveness in ensuring secure data transmission.