Secure Key Exchange Protocol

Using Diffie-Hellman and Signature

Author: Arthur Morain

Date: July 2025

First Encounter with Alice and Bob

---

1 Introduction

In this project, we implemented a simple connection between a server and a client, whose objective is to securely exchange encryption keys in order to communicate confidentially. To achieve this, we use the Diffie-Hellman key exchange protocol, allowing both parties to agree on a shared secret over an insecure channel. This shared key is then used to encrypt messages using a symmetric cipher (AES, for instance).

The implementation is split into several components: a basic TCP network layer, key generation and exchange functions, and future support for encryption and decryption.

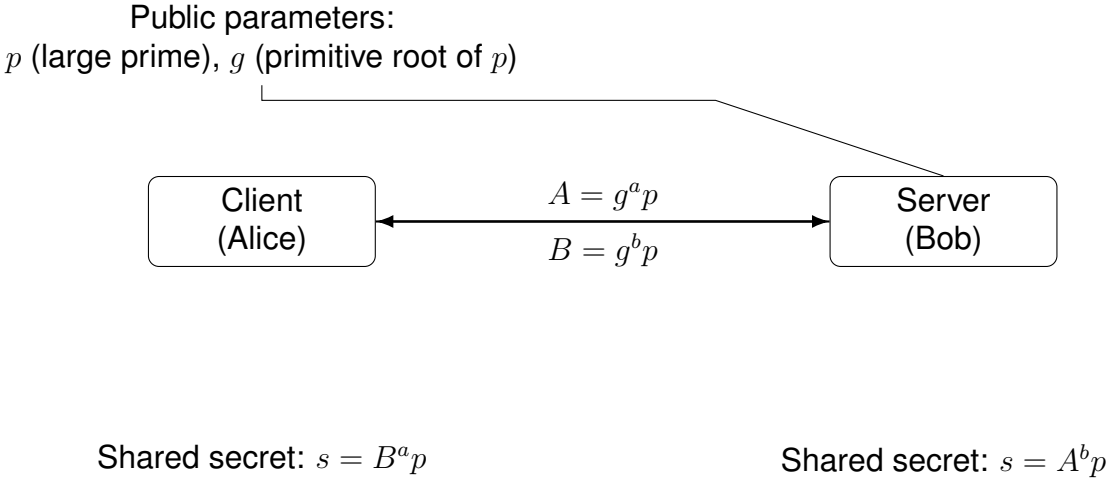

Figure 1: Diffie-Hellman key exchange with public parameters p and g

It is possible to observe the network packets exchanged between the client and the server using Wireshark on the loopback interface (lo). Through this, one can verify that the public keys A and B are indeed exchanged openly over the network during the Diffie-Hellman key exchange process.

The full source code for this project is available at: `https://github.com/Athos-0day/SecureSocketDH`

2 Risk of Man-in-the-Middle (MITM) Attack

In the current setup, the client and the server exchange their public keys openly over the network without any authentication mechanism. This exposes the communication to a Man-in-the-Middle (MITM) attack, where an attacker can intercept and alter the key exchange, making both parties believe they are communicating securely with each other, while in reality, the attacker controls the connection.

Since implementing a realistic MITM attack would require the creation of additional sockets and managing separate key exchanges with both client and server (effectively

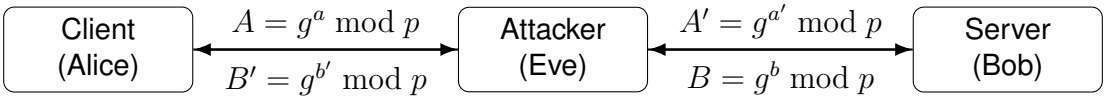

Figure 2: Man-in-the-Middle attack on Diffie-Hellman key exchange

acting as a proxy), this project does not include this feature. However, understanding this vulnerability highlights the importance of authentication mechanisms in secure communications.

3 Preventing MITM Attacks

A major weakness of the basic Diffie-Hellman exchange is its vulnerability to Man-in-the-Middle (MITM) attacks. To mitigate this, we added **digital signature authentication**.

The server signs its public key using its private key. The client verifies the signature using the server's known public key. This prevents an attacker from injecting a forged key.

Key exchange flow with signature:

```
Client                          Server
  |                               |
  | <------- p, g ------------- |
  | <--- public key + sig ----- |
  |                               |
  | -----> public key --------> |
  |                               |
```

This ensures that the public key truly originates from the server, protecting against key tampering.

Technical Conclusion

In this project, we established a secure communication channel between a client and a server using the Diffie-Hellman key exchange protocol, strengthened by the use of digital signatures to prevent Man-in-the-Middle (MITM) attacks. The secure exchange of public keys allows both parties to derive a shared secret, which can be used as the basis for symmetric encryption.

While implementing message encryption was not the focus of this project, various encryption algorithms have been developed and are available in a separate repository. These implementations (including Vigenère, simplified RSA, XOR, and others) can be explored at:

`https://github.com/Athos-0day/Introduction-to-ciphers`

This project lays a strong foundation for building a fully secure communication system that includes key exchange, authentication, and data encryption.

Personal Conclusion

This project was my first personal experience in applied cryptography and network programming. As a self-initiated project, it is intentionally brief and focused on a single goal: enabling secure key exchange between a client and a server using the Diffie-Hellman protocol.

Without academic constraints or predefined instructions, I found it more challenging to define the scope and next steps of the project. Doing my own research, identifying potential security risks like MITM attacks, and integrating cryptographic signatures all required patience and curiosity.

Despite its modest size, this project taught me valuable lessons about how secure communication is established and the importance of authentication in cryptographic systems. It lays a solid foundation for future, more advanced explorations.

References

[1] Cédric VIDAL.
Network programming in C: sockets.
Available at: `http://vidalc.chez.com/lf/socket.html`

[2] Laurie, L.
Crypto 101 — Introduction to Cryptography.
Available at: `https://crypto101.io`

[3] OWASP Foundation.
Man-in-the-Middle (MITM) Attack.
Available at: `https://owasp.org/www-community/attacks/Man-in-the-middle_attack`

[4] National Institute of Standards and Technology (NIST).
Digital Signature Standard (DSS).
Disponible sur : `https://csrc.nist.gov/publications/detail/fips/186/4/final`

[5] Athos-0day.
Introduction to Ciphers.
Available at: `https://github.com/Athos-0day/Introduction-to-ciphers`

[6] Athos-0day. *SecureSocketDH - A personal project implementing authenticated Diffie-Hellman key exchange in C*. Available at: `https://github.com/Athos-0day/SecureSocketDH`