MIT World Peace University

Information and Cyber Security

Assignment 6

Naman Soni Roll No. 10

Contents

1 Aim

Write a program using JAVA or Python or C++ to implement Diffie-Hellman Key Exchange Algorithm

2 Objectives

To learn how to distribute the key

3 Theory

3.1 *Explain Diffie-Hellman algorithm with example.*

The Diffie-Hellman key exchange algorithm is a method for two parties to securely establish a shared secret key over an insecure communication channel, without any prior knowledge of each other's secret keys. The algorithm works as follows:

1. Alice and Bob agree on a public prime number p and a primitive root g of p, which are known to both parties
2. Alice chooses a secret number a, computes $A = g^a$ mod p, and sends A to Bob
3. Bob chooses a secret number b, computes $B = g^b$ mod p, and sends B to Alice.
4. Alice computes the shared secret key $K = B^a$ mod p.
5. Bob computes the shared secret key $K = A^b$ mod p.
6. Alice and Bob now have the same shared secret key K, which they can use for further communication using a symmetric encryption algorithm.

Here's an example of how the Diffie-Hellman algorithm works:

1. Alice and Bob agree on a public prime number p = 23 and a primitive root g = 5 of p.
2. Alice chooses a secret number a = 6, computes $A = g^a$ mod $p = 5^6$mod 23 = 8, and sends A = 8 to Bob.
3. Bob chooses a secret number b = 15, computes $B = g^b$ mod $p = 5^{15}$ mod 23 = 19, and sends B = 19 to Alice.
4. Alice computes the shared secret key $K = B^a$ mod $p = 19^6$ mod 23 = 2.
5. Bob computes the shared secret key $K = A^b$ mod $p = 8^{15}$ mod 23 = 2.
6. Alice and Bob now have the same shared secret key K = 2, which they can use for further communication using a symmetric encryption algorithm.

Note that even though Eve, an eavesdropper who can intercept the messages exchanged between Alice and Bob, knows the values of p, g, A, and B, she cannot compute the shared secret key K without knowing either a or b, which are kept secret by Alice and Bob respectively. This is what makes the Diffie-Hellman key exchange algorithm secure.

4 Code

```
    import sympy
import random

n: int = 0
g: int = 0
repeat: bool = True
while repeat:
    n = int(input("Enter prime number n: "))
    if sympy.isprime(n) and n >= 3:
        repeat = False
        while True:
            g = int(input("Enter g: "))
            if sympy.is_primitive_root(n, g) and sympy.isprime(g):
                break
    else:
        print("Oops not a prime enter again !!!!")
print("Primitive root of n i.e g: ", g)


def get_random_prime_xA():
    while True:
        xA = random.randint(1, n - 1)
        if sympy.isprime(xA):
            print("Private key of User A: ", xA)
            return xA


def get_random_prime_xB():
    while True:
        xB = random.randint(1, n - 1)
        if sympy.isprime(xB):
            print("Private key of User B: ", xB)
            return xB

xA = get_random_prime_xA()
xB = get_random_prime_xB()

yA = (g ^ xA) % n
yB = (g ^ xB) % n
print("Public key of User A: ", yA)
print("Public key of User B: ", yB)

print("Public Key Exchanged")

kA = (yB ^ xA) % n
print("Symmetric key calculated by User A: ", kA)

kB = (yA ^ xB) % n
print("Symmetric key calculated by User B: ", kB)
```

5 Conclusion

In this assignment we have learned how to distribute the key using Diffie-Hellman algorithm.

6 FAQ's

6.1 *What are other key exchange protocols, other than DH algorithm?*

There are several other key exchange protocols that are used to establish a shared secret key between two parties. Here are some examples:

- **RSA Key Exchange:** This protocol is based on the RSA encryption algorithm and involves the exchange of encrypted keys between two parties. It is commonly used in TLS/SSL protocols to establish a secure communication channel between a client and a server.
- **Elliptic Curve Diffie-Hellman (ECDH):** This protocol is a variation of the Diffie-Hellman algorithm that uses elliptic curves instead of prime numbers to generate the shared secret key. ECDH is more efficient than the original Diffie-Hellman algorithm and is commonly used in applications where resource-constrained devices are involved.
- **Station-to-Station (STS) Protocol:** This protocol is an extension of the Diffie-Hellman algorithm that includes mutual authentication between the two parties. It is commonly used in secure email protocols such as PGP (Pretty Good Privacy) and S/MIME (Secure/Multipurpose Internet Mail Extensions).
- **Kerberos Protocol:** This protocol is a network authentication protocol that uses symmetric key cryptography to provide secure communication between two parties. It is commonly used in enterprise environments to provide centralized authentication and authorization services.
- **Secure Remote Password (SRP) Protocol:** This protocol is a password-authenticated key agreement protocol that allows two parties to securely establish a shared secret key based on a user's password. It is commonly used in secure online authentication systems such as VPNs (Virtual Private Networks) and online banking.

Each key exchange protocol has its own advantages and disadvantages and is suited for different applications. The choice of which protocol to use depends on the specific requirements of the application and the level of security needed.

6.2 *Explain the different types of keys.*

In cryptography, there are several different types of keys that are used for various purposes. Here are some of the most common types of keys:

- **Symmetric Key:** A symmetric key is a type of cryptographic key that is used for symmetric encryption algorithms. In this type of encryption, the same key is used for both encryption and decryption. This means that both the sender and the recipient must have the same key in order to communicate securely.
- **Asymmetric Key:** An asymmetric key, also known as a public key, is a type of cryptographic key that is used for asymmetric encryption algorithms such as RSA or elliptic curve cryptography (ECC). In this type of encryption, two different keys are used for encryption and decryption. One key is known as the public key and is used for encrypting data, while the other key, known as the private key, is used for decrypting the data.
- **Session Key:** A session key is a type of cryptographic key that is used for encrypting data during a single session. It is typically generated dynamically and is used only for a limited period of time.
- **Master Key:** A master key is a type of cryptographic key that is used to generate other keys. It is typically a long-term key that is used to generate session keys or other types of keys.
- **Root Key:** A root key is a type of cryptographic key that is used as the basis for generating other keys. It is typically a long-term key that is used to generate other keys used for specific purposes, such as encryption or digital signatures.
- **Private Key:** A private key is a type of cryptographic key that is kept secret by its owner and is used for decrypting data that has been encrypted with the corresponding public key. Private keys are typically used for digital signatures, where the owner of the private key signs a message to prove its authenticity.

- **Public Key:** A public key is a type of cryptographic key that is freely distributed and used for encrypting data that can only be decrypted with the corresponding private key. Public keys are typically used for digital certificates, where the owner of the public key is identified and verified.

Each type of key serves a different purpose in cryptography and is used for different types of encryption and decryption algorithms. It is important to use the right type of key for the right purpose to ensure the security of the communication.

6.3 *Explain different key management issues.*

Key management refers to the process of generating, storing, distributing, and revoking cryptographic keys. It is a critical aspect of cryptography and plays a vital role in ensuring the security of communication. Here are some of the key management issues that need to be addressed:

- **Key Generation:** The process of generating strong and random cryptographic keys is a critical aspect of key management. Keys should be generated using a secure random number generator and should be of sufficient length to ensure that they cannot be easily guessed or brute-forced.
- **Key Storage:** The storage of cryptographic keys is another important aspect of key management. Keys should be stored securely to prevent unauthorized access or theft. They should also be protected against hardware failures, natural disasters, and other forms of data loss.
- **Key Distribution:** The secure distribution of cryptographic keys to authorized parties is another key management issue. Keys should be distributed using secure channels and protocols to prevent interception or tampering. The distribution process should also ensure that only authorized parties have access to the keys.
- **Key Revocation:** The revocation of cryptographic keys is another important aspect of key management. Keys should be revoked if they are compromised or if the authorized parties no longer require access to them. The revocation process should be timely and efficient to prevent unauthorized access or use of the keys.
- **Key Rotation:** The regular rotation of cryptographic keys is another key management issue. Keys should be rotated periodically to prevent their compromise and to ensure that they remain secure. The key rotation process should be carefully managed to prevent disruptions to communication.
- **Key Escrow:** The process of key escrow involves the storage of cryptographic keys by a trusted third party. This is often used in situations where the loss of keys could result in a catastrophic event. The key escrow process should be carefully managed to prevent unauthorized access or theft of the keys.
- **Key Hierarchy:** The use of a key hierarchy involves the creation of a hierarchy of cryptographic keys with different levels of access and control. This can help to improve the security of communication by limiting access to sensitive keys and data.

These key management issues are critical for ensuring the security of communication and should be carefully managed to prevent the compromise of cryptographic keys and data.