Public Ciphers

Caleb Dennis

CPE321-03

## Task 1: Implement Diffie-Helman key exchange

The Diffie-Helman key exchange allows users to create a shared secret without communicating any information that would make the secret insecure. In task 1 Alice and Bob independently compute the same shared secret which can be used as a key to securely exchange information. To find the shared secret an adversary must find the prime number that Alice or Bob each chose. To find this number the adversary needs to compute the log base g of Alice’s public key. An adversary could check all primes, but if Alice has chosen a reasonably large prime this process takes a lot of computing.

from Crypto.Util.number import getPrime

from Crypto.Util.Padding import pad, unpad

from Crypto.Random import get\_random\_bytes

from Crypto.Hash import SHA256

from Crypto.Cipher import AES

from Crypto.Cipher import AES

def main():

    p = 37

    g = 5

    Alice\_prime = getPrime(16)

    Bob\_prime = getPrime(16)

    print(f"Alice's prime: {Alice\_prime}")

    print(f"Bob's prime: {Bob\_prime}")

    Alice\_public = pow(g, Alice\_prime) % p

    Bob\_public = pow(g, Bob\_prime) % p

    Alice\_shared\_secret = bytes(pow(Bob\_public, Alice\_prime) % p)

    Bob\_shared\_secret = bytes(pow(Alice\_public, Bob\_prime) % p)

    Alice\_hasher = SHA256.new()

    Alice\_hasher.update(Alice\_shared\_secret)

    alice\_key = Alice\_hasher.digest()

    Bob\_hasher = SHA256.new()

    Bob\_hasher.update(Bob\_shared\_secret)

    bob\_key = Bob\_hasher.digest()

    iv = get\_random\_bytes(16)

    print(f"Bob's key {alice\_key}\nAlice's key {bob\_key}\n")

    Alice\_msg = pad(bytes("Hi Bob, i'm Alice", "ascii"), 16)

    print(f"Alice: {unpad(Alice\_msg, 16)}")

    Alice\_encrypter = AES.new(alice\_key, AES.MODE\_CBC, *iv*=iv)

    Alice\_sends\_to\_Bob = Alice\_encrypter.encrypt(Alice\_msg)

    Bob\_encrypter = AES.new(bob\_key, AES.MODE\_CBC, *iv*=iv)

    Bob\_received\_from\_Alice = Bob\_encrypter.decrypt(Alice\_sends\_to\_Bob)

    print(f"Bob received: {unpad(Bob\_received\_from\_Alice, 16)}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

## Task2:

Alice's prime: 63647

Bob's prime: 49603

Bob's key b"\xe3\xb0\xc4B\x98\xfc\x1c\x14\x9a\xfb\xf4\xc8\x99o\xb9$'\xaeA\xe4d\x9b\x93L\xa4\x95\x99\x1bxR\xb8U"

Alice's key b"\xe3\xb0\xc4B\x98\xfc\x1c\x14\x9a\xfb\xf4\xc8\x99o\xb9$'\xaeA\xe4d\x9b\x93L\xa4\x95\x99\x1bxR\xb8U"

Alice: b"Hi Bob, i'm Alice"

Bob received: b"Hi Bob, i'm Alice"

Mallory Intercepted: b"Hi Bob, i'm Alice"

Alice's prime: 50051

Bob's prime: 63367

Bob's key b'm\xb6\_\xd5\x9f\xd3V\xf6r\x91@W\x1b[\xcdk\xb3\xb84\x92\xa1n\x1b\xf0\xa3\x88DB\xfc<\x8a\x0e'

Alice's key b'm\xb6\_\xd5\x9f\xd3V\xf6r\x91@W\x1b[\xcdk\xb3\xb84\x92\xa1n\x1b\xf0\xa3\x88DB\xfc<\x8a\x0e'

Alice: b"Hi Bob, i'm Alice"

Bob received: b"Hi Bob, i'm Alice"

Mallory intercepted: b"Hi Bob, i'm Alice"