CS 535 **Thanga Rani Prabhu** **Week#3**
Dr.Prog.Henry Chang

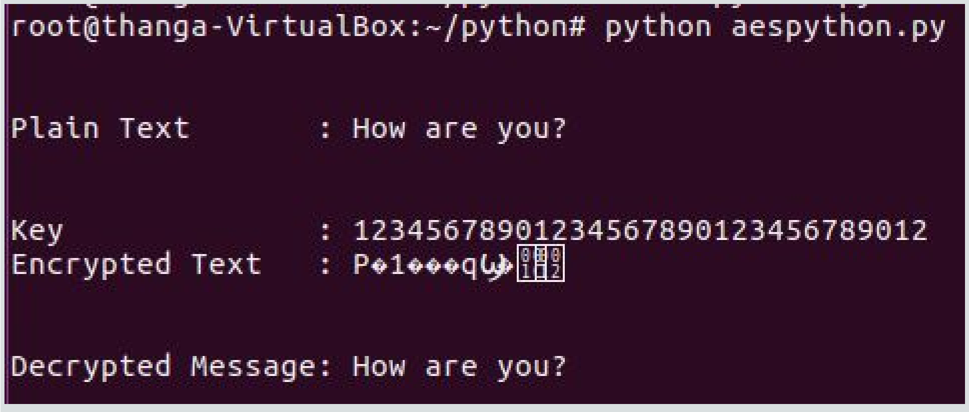
1.

Using CTR mode to encrypt and decrypt the message "How are you?"

Source code:

```
from Crypto.Cipher import AES
from Crypto.Util import Counter
from Crypto import Random
import binascii

# AES supports multiple key sizes: 16 (AES128), 24 (AES192), or 32 (AES256).
key_bytes = 32
key='12345678901234567890123456789012'

# Takes as input a 32-byte key and an arbitrary-length plaintext and returns a
# pair (iv, ciphtertext). "iv" stands for initialization vector.
def encrypt(key, plaintext):
    assert len(key) == key_bytes
    # Choose a random, 16-byte IV.
    iv = Random.new().read(AES.block_size)
    # Convert the IV to a Python integer.
    iv_int = int(binascii.hexlify(iv), 16)
    # Create a new Counter object with IV = iv_int.
    ctr = Counter.new(AES.block_size * 8, initial_value=iv_int)
    # Create AES-CTR cipher.
    aes = AES.new(key, AES.MODE_CTR, counter=ctr)
    #Encrypt and return IV and ciphertext.
    ciphertext = aes.encrypt(plaintext)
    print("\n")
    print("Plain Text       : "+plaintext)
    print("\n")
    print("Key              : "+key)
    print ("Encrypted Text   : "+ciphertext)
    print("\n")
    return (iv, ciphertext)
# Takes as input a 32-byte key, a 16-byte IV, and a ciphertext, and outputs the
# corresponding plaintext.

def decrypt(key, iv, ciphertext):
    assert len(key) == key_bytes
    # Initialize counter for decryption. iv should be the same as the output of
    # encrypt().
    iv_int = int(iv.encode('hex'), 16)
    ctr = Counter.new(AES.block_size * 8, initial_value=iv_int)
    # Create AES-CTR cipher.
    aes = AES.new(key, AES.MODE_CTR, counter=ctr)
    plaintext = aes.decrypt(ciphertext)
    return (plaintext)

(iv, ciphertext) = encrypt(key, 'How are you?')
print ("Decrypted Message: " +decrypt(key, iv, ciphertext))
print("\n")
```

Output



```
root@thanga-VirtualBox:~/python# python aespython.py


Plain Text        : How are you?


Key               : 12345678901234567890123456789012
Encrypted Text    : P�1���qɰ▯


Decrypted Message: How are you?
```

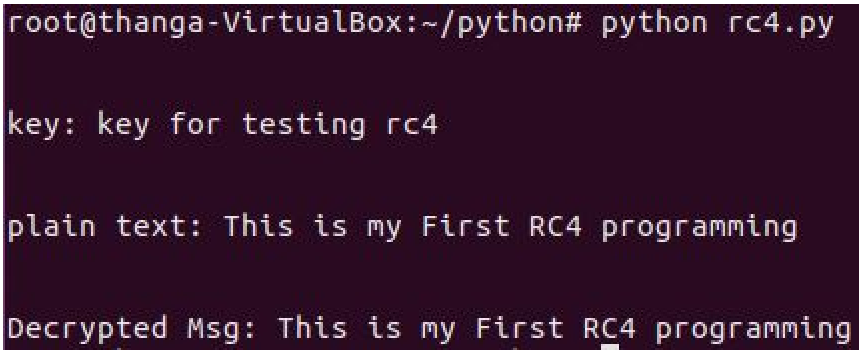
2.

Using RC4 mode to encrypt and decrypt the message.

Source code

```
#! /usr/bin/python
from Crypto.Cipher import ARC4
from Crypto.Hash import SHA256
from Crypto import Random

def enc(key,p):
        return ARC4.new(key).encrypt(p)

def dec(key,msg):
    return ARC4.new(key).decrypt(msg)

def main():
    print("\n")
    key = 'key for testing rc4'
    p = 'This is my First RC4 programming'
    print("key: "+key)

    print("\n")
    print("plain text: "+p)
    print("\n")
    nonce=Random.new().read(16)
    key += nonce
    key = SHA256.new(key).digest()
    #key is no more than 256 bytes

    print ("Decrypted Msg: "+dec(key,enc(key,p)))

if __name__=='__main__':
                    main()
```

Output



```
root@thanga-VirtualBox:~/python# python rc4.py


key: key for testing rc4


plain text: This is my First RC4 programming


Decrypted Msg: This is my First RC4 programming
```

Discussion Board:

A.

1. **How many keys are required for N number of users if symmetric key cryptography is used?**

   N(N−1)/2 keys.

2. **How many keys are req. for N number of users if asymmetric key cryptography is used?**

   2N

Let us assume for how many keys are required for secure communication among 1000 person?

Symmetric Key Cryptography:

N(N−1)/2

(1000×999)/2 = 499500

So they would need 499500 symmetric keys to have a secure communication between all of them.

Asymmetric Key Cryptography:

Each one would have 2 keys, so a total of 2000 keys.

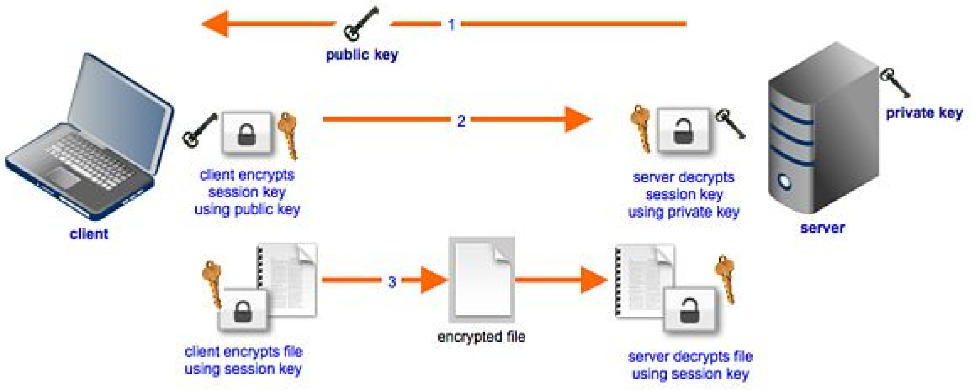
B.

Why Asymmetric key cryptography alone cannot resolve Internet security issue?

- **Size of cryptogram**: symmetric encryption does not increase the size of the cryptogram (asymptotically), but asymmetric encryption does. If we take the example of RSAES-OAEP in PKCS#1v2 with a 1024-bit key and 160-bit SHA-1 hash, a 1024-bit cryptogram can convey a maximum of 688 bit of useful information. Thus data enciphered in this way would cost 49% more space to store, or more time to move over a given link
- **Speed**: Even though Asymmetric key is secure but it is not as quick as Symmetric, because of Asymmetric key cryptography have a lot a mathematical structure.
- **Computational Cost:** The public key algorithms known thus far are relatively computationally costly compared with most symmetric key algorithms of apparently equivalent security.

Because both symmetric and asymmetric key cryptography have their own advantages, modern file transfer systems typically employ a hybrid of the two. Some hybrid cryptosystems are: SSL (used in FTPS and HTTPS), SSH (used in SFTP), and OpenPGP, all of which are supported by JSCAPE MFT Server.

Hybrid cryptosystems employed in an SFTP or FTPS server use **asymmetric keys to initially encrypt symmetric keys known as session keys**. The session keys are then the ones used to encrypt the actual data. As its name implies, a session key is only used in one session. After the session, the key is simply discarded. That's a good thing because even if a session key is compromised, only data sent within that particular session will be at risk.


1
public key
client
2
client encrypts session key using public key
server decrypts session key using private key
private key
server
3
client encrypts file using session key
encrypted file
server decrypts file using session key

C.

Please assess the following statements and give your reasons:

- **Encryption in symmetric key cryptography provides authentication.**
- **Encryption in asymmetric key cryptography provides authentication.**

Answer:

Encryption in asymmetric key cryptography provides authentication Because A symmetric key can be used to check the identity of the individual who originated a particular set of data, but this authentication scheme can encounter some thorny problems involving trust.

Fortunately, asymmetric algorithms can be used to solve these problems by performing the same basic operations but encrypting the hash using a private key (belonging to an asymmetric key pair) that one individual and only one individual knows. Then anyone can use the associated public key to verify the hash. This effectively eliminates the problems of trust and repudiation.This technique is called a **digital signature.**

D.

NSA prefers exportable security algorithms easier to break or harder to break?

Harder to break

E.

To achieve the same level of security, which one need to use a larger key size? Symmetric key cryptography or asymmetric key cryptography? Please explain your assessment.

Symmetric key cryptography should need to use larger key size to get the same level of security.

Reason: As the symmetric key is easily hacked by the middle man. Larger the key size, a lot of computational power is needed to get the guess the key. That’s why it attains more secure.

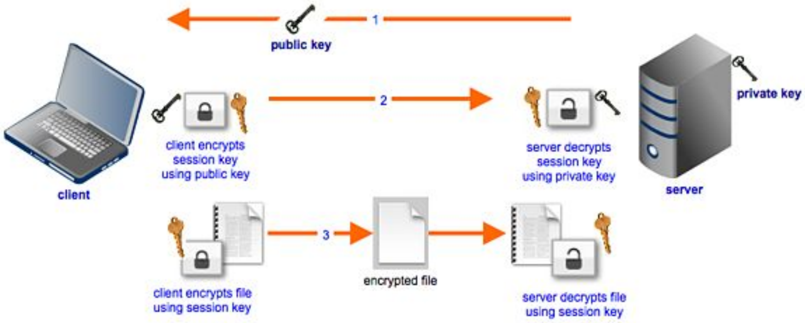
F.

Symmetric key cryptography and asymmetric key cryptography are complementary. Please explain why and how?

Hybrid Encryption Methods

- Use of Symmetric and Asymmetric Algorithms as complementary methods
- Symmetric key is generally used to encrypt the message
- Asymmetric key is used to encrypt the Symmetric key
- This process is more commonly called as digital envelope

Session Keys • Single use symmetric key that is used to encrypt/decrypt communication between two users for a single session • Its much secure than static symmetric keys • Peers decide on the session key and continue to use it till the session is over • Eavesdropping is difficult, breaking the keys is futile


1
public key
private key
2
client encrypts session key using public key
server decrypts session key using private key
client
server
3
client encrypts file using session key
encrypted file
server decrypts file using session key

G.

What are the principal ingredients of a public-key cryptosystem?

- Plaintext
- Encryption algorithm
- Public and private keys
- Ciphertext
- Decryption algorithm

H.

List and briefly define three uses of a public-key cryptosystems?

- Encryption/decryption: The sender encrypts a message with the recipient's public key.
- Digital signature: The sender "signs" a message with its private key.
- Key exchange: Two sides cooperate to exchange as session key. Several different approaches are possible, involving the private key(s) of one or both parties

---

I.

What is the difference between a private key and a secret key?

- The key used in conventional encryption is typically referred to as a secret key. The two keys used for public-key encryption are referred to as the public key and the private key.
- Secret key is known between two people who are in transaction. But Private key is known only to an individual.

J.

How can public-key encryption be used to distribute a secret key?

- Diffie-hellman exchange
- RSA (Rivest, Shamir and Adleman) key exchange
