《物联网安全》实验课

加解密算法与加密通信设计与实现

目录

PART 01

基础原理

BASIC PRINCIPLE

实验原理

加密算法分类

- 对称加密
- DES加密算法
- 非对称加密
- RSA加密算法

根据加密与解密是否采用同一个密钥，可以将现代密码学算法分为对称加密算法和非对称加密算法量大类型，这两种算法各有优势和相应的应用场景。

实验原理

- 加密算法分类
- 对称加密
- DES加密算法
- 非对称加密
- RSA加密算法

优点

1.加解密过程简单快速
2.可以用于大量数据加密

缺点

1.加解密使用相同的密钥，当通信成员数量增加时，为保证两两通信都采用独立的密钥，密钥数量与成员的平方成正比，面临密钥管理难题
2.如何解决密钥分发问题

实验原理

加密算法分类

对称加密

DES加密算法

非对称加密

RSA加密算法

DES(Data Encryption Standard)算法，于1977年得到美国政府的正式许可，是一种用56位密钥来加密64位数据的分组加密算法。下面是DES算法的基本步骤：

1.密钥生成:
使用密钥生成算法生成一个56位的密钥（通常以64位形式表示，其中包括8个奇偶校验位），再通过旋转重组生成16个48位子密钥。

2.初始置换:
对输入的64位明文按照固定的算法进行初始置换，将其重新排列。

3.加密轮:
在加密轮中，使用子密钥将明文分成左右两部分。对每个加密轮，右半部分经过扩展置换（Expansion Permutation）扩展为48位。将扩展后的右半部分与子密钥进行异或运算。异或运算的结果经过S盒置换和P置换。
左半部分与经过S盒置换和P置换后的结果进行异或运算，得到下一轮的左半部分和右半部分。重复以上步骤执行16轮加密运算。

4.交换置换（Swap）:
在最后一轮加密后，将左右两部分进行交换并连接。

5.逆初始置换:
对交换置换后的右左两部分按照进行重新排列，撤销初始置换。

实验原理

- 加密算法分类
- 对称加密
- DES加密算法
- 非对称加密
- RSA加密算法

优点

1.加解密使用不同的密钥，公钥可以完全公开，无须安全传输保证，私钥由用户自行保管，不参与任何通信传输
2.可用于加密和签名

缺点

1.计算复杂度高，相比对称加密差好几个数量级
2.不能加密超过密钥长度的数据

实验原理

加密算法分类

对称加密

DES加密算法

非对称加密

RSA加密算法

RSA算法的步骤如下：

1）选择两质数 p、q。
2）计算 n = p*q。
3）计算 n 的欧拉函数 ϕ(n)=(p-1)*(q-1)。
4）选择整数 e，使e 与ϕ(n)互质，且 1<e<ϕ(n)。
5）计算 d，使d * e ≡ 1 (mod ϕ(n))。
其中，公钥 KU ={e,n}，私钥 KR ={d,n} 。
对于明文M
加密：C≡ M^e mod n。
解密：M≡ C^d mod n。

例：取两个质数p=11，q=13，p和q的乘积为n=p×q=143，算出欧拉函数ϕ(n)=(p-1)×(q-1)=120；再选取一个与ϕ(n)=120互质的数，例如e=7，则公开密钥=（n，e）=（143，7）。对于这个e值，可以算出其逆：d=103。因为e×d=7×103=721，满足e×d mod ϕ(n) =1；即721 mod 120=1成立。则秘密密钥=（n，d）=（143，103）。

实验原理

所以要结合非对称加密和对称加密的优点，以混合加密来保护通信安全，具体做法是用非对称加密来安全地传递少量数据给通信的另一方，再以这些数据为密钥，采用对称加密来安全高效地大量加密传输数据，这种由多种加密算法组合的应用形式称为“密码学套件”。

PART 02

实验环境

EXPERIMENTAL ENVIRONMENT

实验环境

python3运行环境

Wireshark等抓包工具

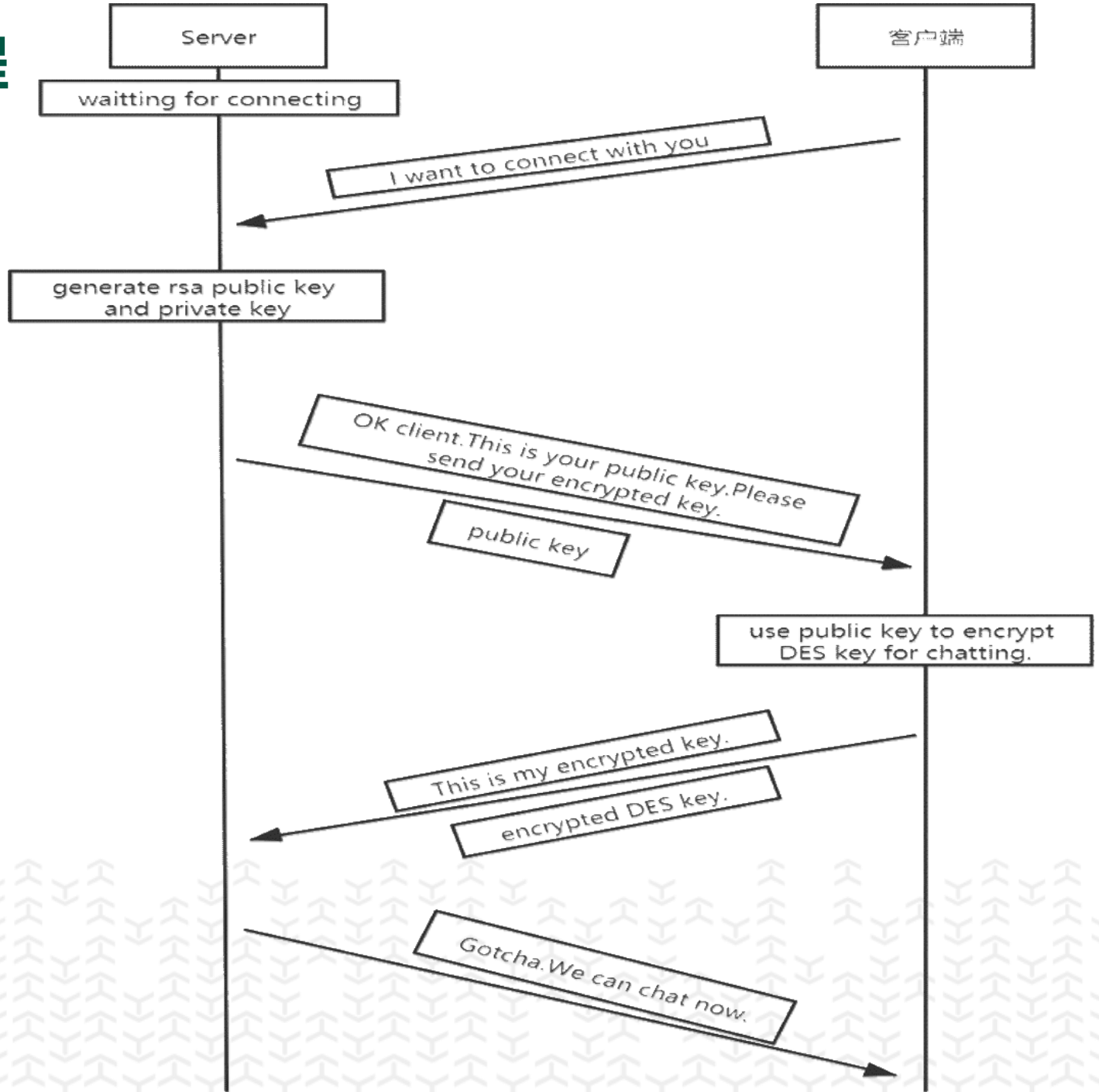
pycryptodome加密算法库

PART 03

实验步骤

EXPERIMENTAL PROCEDURE

通信过程


Server
客户端
waitting for connecting
I want to connect with you
generate rsa public key
and private key
OK client.This is your public key.Please
send your encrypted key.
public key
use public key to encrypt
DES key for chatting.
This is my encrypted key.
encrypted DES key.
Gotcha.We can chat now.

实验步骤

01 安装环境

1.pip3 install pycryptodome
2.安装Wireshark抓包工具

02 使用Crypto生成私钥和公钥

```
from Crypto import Random
from Crypto.PublicKey import RSA
random_generator = Random.new().read
rsa = RSA.generate(2048, random_generator)
# 生成私钥
private_key = rsa.exportKey()
# 生成公钥
public_key = rsa.publickey().exportKey()
```

03 使用公钥对key加密

```
public_key = RSA.importKey(public_key_bytes)
cipher = PKCS1_v1_5.new(public_key)
key_text = cipher.encrypt(key)
encrpted_key = base64.b64encode(key_text)
```

实验步骤

04 使用私钥解密key

```
# 解密key
def decrypt_key(encrypt_msg):
    private_key = get_key('rsa_private_key.pem')
    cipher = PKCS1_v1_5.new(private_key)
    back_text =
cipher.decrypt(base64.b64decode(encrypt_msg),
0)
    return back_text
```

05 加密通信

利用socket和threading实现加密通信

```
2023-09-05 17:22:03 send public_key:
b'-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0E
2023-09-05 17:22:03 recv encrypted DES-key:b'Ckor
2023-09-05 17:22:03 decrypt DES-key:b'6A4B3f7D'
2023-09-05 17:22:03 服务器>>>:
2023-09-05 17:22:31
收到客户端加密信息:dBzle0kyfQwlO1tWfjRiEGyNWW035pQ0
信息解密为:你好，星期三
```

06 使用Wireshark抓包

通过ip或端口抓包，观察加密通信过程并记录
tcp.srcport == 8888 or tcp.dstport == 8888

PART 04

验收考核

Inspection And Acceptance

验收考核

1.密钥传递过程清晰
2.加密通信过程清晰
3.Wireshark抓包能清楚反映实验过程

谢谢