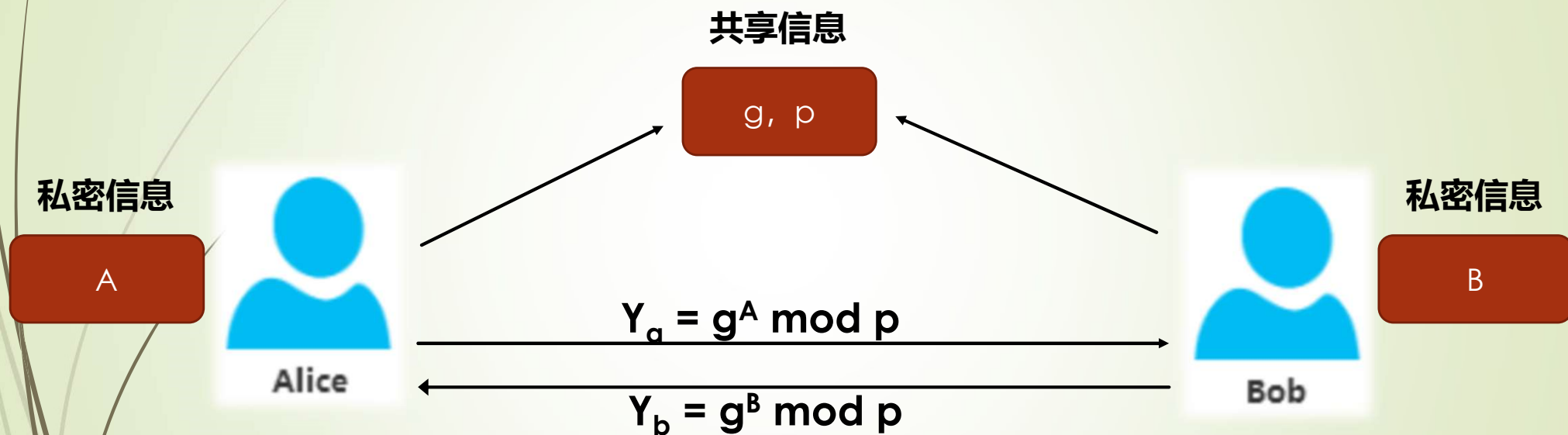


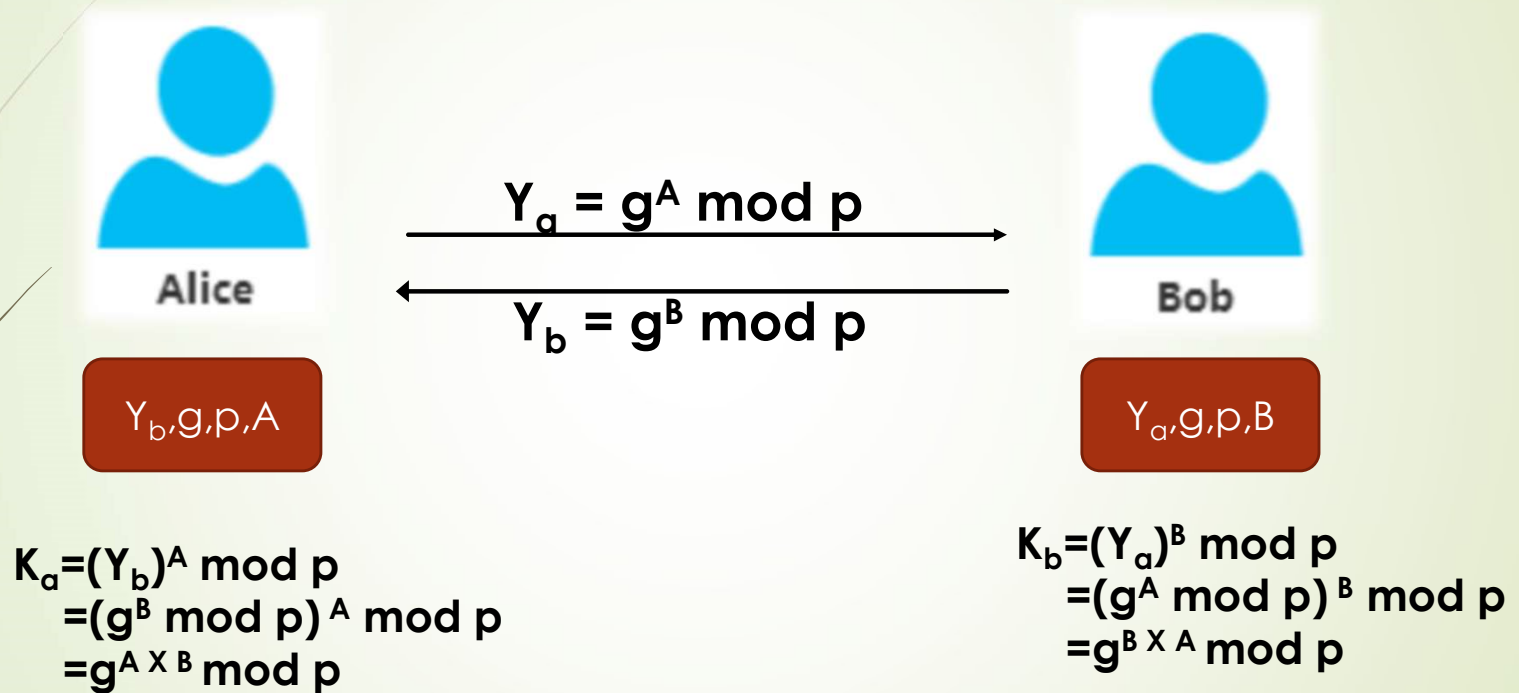


## 针对DH密钥协商的中间人攻击

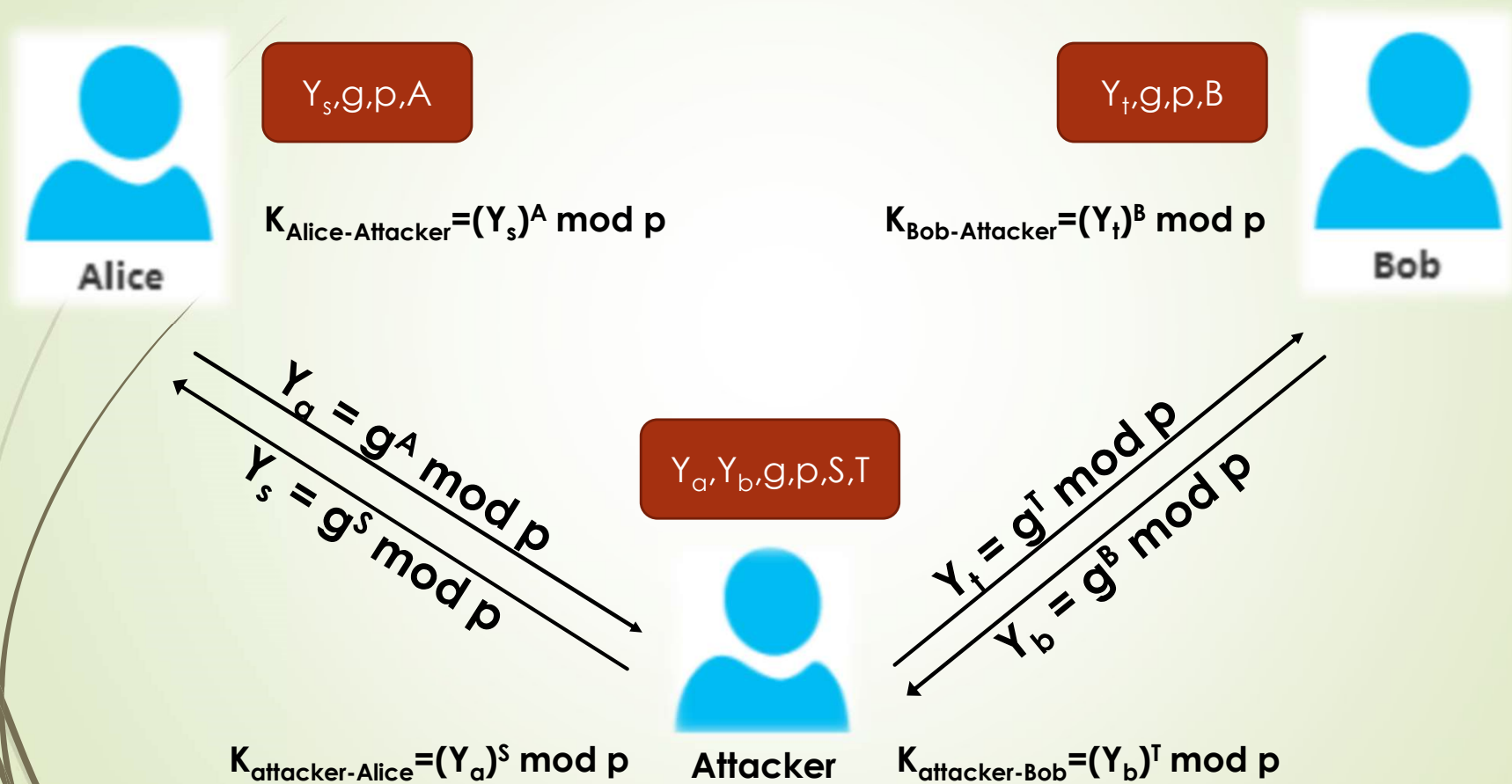
## DH密钥协商的基本原理



## DH密钥协商的基本原理



## DH密钥协商的中间人攻击



## DH密钥交换的实现



- 1) 选择的 $p$ 是一个**质数**
- 2)  $g$ 是 $p$ 的**本原根**
- 2)  $g$ 、 $p$ 作为共享信息，**明文**传输



# DH密钥交换的实现

## 判断一个输入数据是否是质数

```
def isprime() :  
    #判断是否素数, 直至输入为素数为止  
    count = 1  
    while count:  
        n = int(input("输入一个质数(p): "))  
        for i in range(2, n):  
            if n % i == 0:  
                print("%d不是一个质数! " % n)  
                break  
        else:  
            return n
```

# DH密钥交换的实现

## 求质数p的本原根

```
def get_generator( p):  
    # 获取一个原根  
    # 素数必存在至少一个原根  
    #  $g^{(p-1)} = 1 \pmod p$  当且仅当指数为p-1的时候成立  
    a=2  
    while 1:  
        if a** (p-1) % p == 1:  
            num = 2  
            mark = 0  
            while num < p-1:  
                if a**num % p == 1:  
                    mark = 1  
                    num += 1  
            if mark == 0:  
                return a  
        a += 1
```

# DH密钥交换的实现

## 计算 $g^A \bmod p$ 和 密钥 Key

```
def get_cal(a, p, rand):  
    # 获得计算数  
    cal = (a**rand) % p  
    return cal
```

## 计算密钥 Key

```
def get_key(cal_A, cal_B, p):  
    # 获得密钥  
    key = (cal_B ** cal_A) % p  
    return key
```



# 截获Alice和Bob的通信

## 如何截获Alice和Bob之间的通信呢?

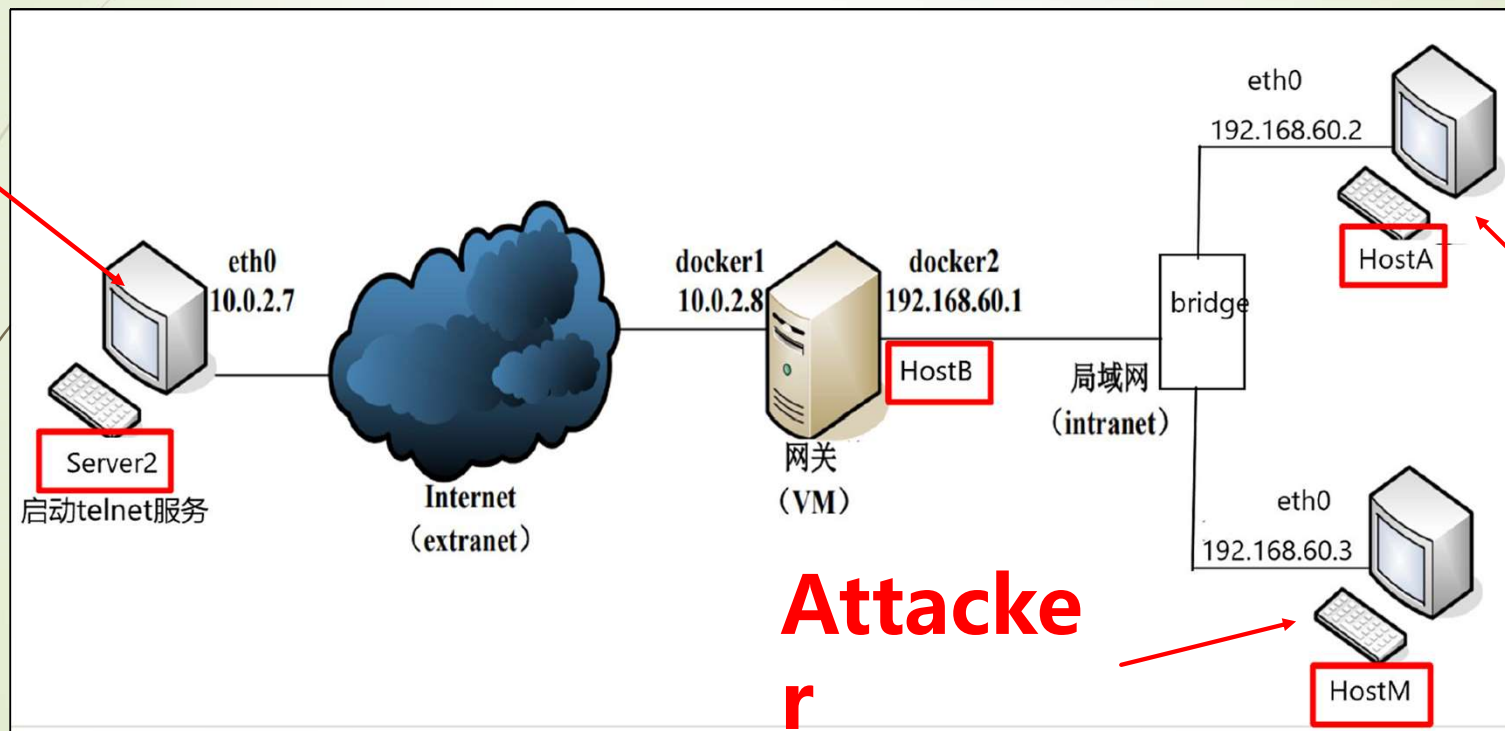
### 实验8 ~~ARP缓存中毒~~

#### 主要内容

- ❑ **MAC和ARP**协议
- ❑ **ARP**缓存中毒攻击
- ❑ 利用**ARP**缓存中毒实施中间人攻击

## 截获Alice和Bob的通信

**Bob**



**Alice**

**Attacker**

# 创建实验环境

- 在 VM 上创建 docker 网络 extranet

```
$ sudo docker network create --subnet=10.0.2.0/24 --gateway=10.0.2.8 --opt  
"com.docker.network.bridge.name"="docker1" extranet
```

- 在 VM 上创建 docker 网络 intranet

```
$ sudo docker network create --subnet=192.168.60.0/24 --gateway=192.168.60.1 --opt  
"com.docker.network.bridge.name"="docker2" intranet
```

- 在 VM 上新开一个终端，创建并运行容器 Server2

```
$ sudo docker run -it --name=Server2 --hostname=Server2 --net=extranet --ip=10.0.2.7 --  
privileged "seedubuntu" /bin/bash
```

- 在 VM 上新开一个终端，创建并运行容器 HostA

```
$ sudo docker run -it --name=HostA --hostname=HostA --net=intranet -- ip=192.168.60.2 --  
privileged "seedubuntu" /bin/bash
```

- 在 VM 上新开一个终端，创建并运行容器 HostM

```
$ sudo docker run -it --name=HostM --hostname=HostM --net=intranet --ip=192.168.60.3 --  
privileged "seedubuntu" /bin/bash
```

## 设计DH密钥协商的报文格式

DH密钥协商算法规定了密钥协商需要交换的**信息**和计算密钥的**方法**

具体信息在网络中传输的**协议**？

- ◆ 如何区分发送共享信息和计算信息
- ◆ 共享信息、计算信息的长度
- ◆ 双方协商报文交换的次序
- ◆ ...

# 发送和接收报文 使用UDP套接字进行数据传输

## 服务器端代码

```
root@Server2: /home/seed 80x24
from socket import *

udp_srv = socket(AF_INET, SOCK_DGRAM)
udp_srv.bind(('10.0.2.7', 9000))
data, addr = udp_srv.recvfrom(1024)

print(data)
print(addr)

print('send the reply')

udp_srv.sendto('hello too', addr)
```



# 发送和接收报文 使用UDP套接字进行数据传

输  
客户端代码

```
root@HostA: /home/seed 67x24
from socket import *

udp_client = socket(AF_INET, SOCK_DGRAM)
udp_client.sendto('hello'.encode('utf-8'), ('10.0.2.7', 9000))

data, addr = udp_client.recvfrom(1024)

print(data)
print(addr)
```



# 中间人攻击

在使用ARP缓存中毒方法，HostM上截获DH交换报文，修改之...

```
from scapy.all import *

def spoof_pkt(pkt):
    print("Original Packet.....")
    print("Source IP : ", pkt[IP].src)
    print("Destination IP :", pkt[IP].dst)

    a = IP()
    b = UDP()
    data = pkt[UDP].payload

    #deal with DH exchange data btween hostA and hostM or server2 and hostM

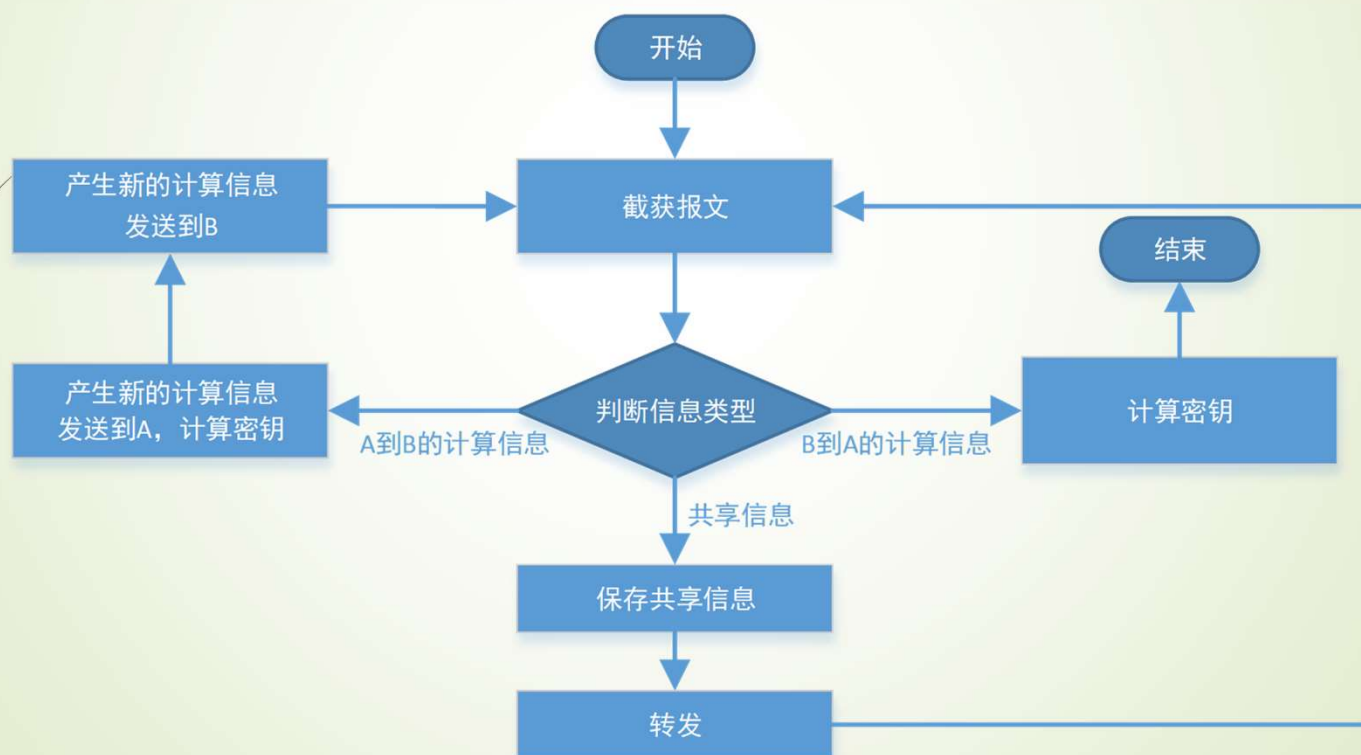
    newpkt = a/b/data


    print("Spoofed Packet.....")
    print("Source IP : ", newpkt[IP].src)
    print("Destination IP :", newpkt[IP].dst)
    send(newpkt)

pkt = sniff(filter='udp',prn=spoof_pkt)
```

# 中间人攻击

## 中间人HostM上对接获报文的主要处理流程





## 实验任务1

在HostA和Server2之间实现基于UDP协议的DH密钥协商，要求：

- ◆ 设计DH密钥协商的**报文格式**，提交到学习通
- ◆ 在HostA实现客户端程序，Server2实现服务器程序
- ◆ 在网关上Wireshark捕获报文，解析报文得到DH密钥协商的**共享信息**和双方的**计算信息**，将截图提交到学习通上
- ◆ 在HostA和Server2上输出协商的**会话密钥**，并将截图提交到学习通

## 实验任务2

在完成实验任务1的基础上，在HostM上开启针对DH密钥协商的中间人攻击，要求：

- ◆ 在网关上Wireshark捕获报文，**解析报文**得到hostA发送的**计算信息**和HostM冒充Server2返回的**计算信息**，并将截图提交到学习通上
- ◆ 在网关上Wireshark捕获报文，解析报文得到hostM冒充HostA发送的**计算信息**和Server2返回的**计算信息**，并将截图提交到学习通上
- ◆ 在HostA、hostM和Server2分别输出**会话密钥**，提交截图到学习通上