先对数据加密再传输,对方收到后解密再使用,这样虽数据仍要经过万水千山,但攻击者面对密文也无可奈何.

加密算法都需要密钥.加密者用密钥加密明文得到密文.解密者必须用同样的密钥才能解出明文.(这也被称为对称加密)

加解密使用同样的密钥,互为逆向过程

所以浏览器和服务器如何确定出一个同样的密钥呢?

如果由一方生成,再先以明文的方式直接传输过去,中间的攻击者自然也能轻松的拦截下来.

攻击者一旦也有了密钥,那么后续的加密就没有意义了.

让通信双方线下交流不失为一个完美的方案,但每次上个网都要坐火车去和站长接个头,这毫不现实.

更实际的解决方法是使用非对称加密.

非对称加密是一个和对称加密相呼应的概念.

我们说对称加密的加解密使用的是同一个密钥,具有对称性.

而在非对称加密中,密钥总是成对出现的,分别称之为公钥和私钥.

用公钥加密的数据只能被私钥解密,公钥自己也无法解密;

同样私钥加密的数据只能被公钥解密,私钥自己也无法解密,也就是说加解密的过程并不对称.

如此,服务器先将自己的公钥发送给浏览器,浏览器生成一个随机的数据,用服务器的公钥进行加密,再发送给服务器

服务器用自己的私钥解密,如此,双方就得到了一个同样的随机数据.

这个随机数据便可以作为对称加密的密钥,对真正要传递的数据进行加密传输.

在这个过程中,即便是攻击者拦截到了服务器的公钥也无济于事.

因为公钥无法解密由它自身加密的数据

使用非对称加密协商出一个相同的密钥,然后用这个密钥进行对称加密传输正式数据

客户端:你好

服务端:你好

服务器:服务器公钥

服务器:打招呼结束

客户端:对称加密密钥

服务器:好的

对称加密

这就是 https 协议中 s 实现的大致原理.

这是一套独立于 http 的流程,也被称之为(安全套接字层)(Secure Socket Layer)

而这个密钥协商的过程也被称之为 SSL 握手.

从 1994 年到 1996 年,SSL 分别经历了 1.0,2.0 和 3.0 的版本升级.

而在 3.0 的基础上,互联网工程事务组(IETF)决定将其标准化

最终于 1999 年以 TLS 这个新的称呼被发布.

如果攻击者在服务器传递给浏览器自己公钥的过程中把它拦截,并替换成自己的公钥再发送给浏览器,又当如何?

浏览器收到后,它无法知道这是被篡改过的.

仍然用它来加密作为后续对称加密公钥的随机数据.攻击者收到后,因为是被自己的公钥加密的数据,所以自然可以用自己的私钥解密.

得到明文,然后再用服务器的公钥对其加密,再发送给服务器.服务器用自己的私钥解密.

攻击者就像一个黑中介一样,两头骗.

这样虽然通信的双方协商出了对称加密的密钥

但攻击者也知道了,所以接下来的加密变得毫无意义.

问题的根本在于(公钥并不能表明自己属于谁)

所以解决的方式是让其具有表明自己身份的能力

这需要引入一个第三方的角色.

现在服务器除了自己的公钥以外,还把自己的域名,组织名,以及所申请的第三方机构名等信息放在一起,形成一个数据集合,

然后拿着这份数据去找这个第三方机构,该机构也有一个公私钥对,用自己的私钥对这些数据进行加密,得到一个密文,这被称为签名.

然后把签名数据和原始明文放在一起,发送给服务器的管理员,这就是所谓的 TLS 证书.

这个第三方机构也被称为 CA

现在服务器传递给浏览器的不再是自己的公钥,而是这个能够表明自己身份的证书,浏览器拿到这个证书后需要先进行验证而不是选择直接相信.

方法也很简单,拿 CA 机构公开的公钥对证书中的密文进行解密,如果解密后的结果和证书中的明文一致,通过验证.

然后从证书中提取出服务器的公钥,加密随机数据发送,双方协商出对称加密的密钥.

如果结果不一致则认为证书不合法,风险提示就该出现了.

人们提出了证书透明的方案

在加入 CT 的安全机制下,要求 CA 机构每次颁发一个证书的时候,都要向一个叫日志服务的角色提交证书的详情.

日志服务负责将其记录下来 同时向 CA 返回一个SCT,CA 将SCT数据加入证书的扩展中,把这个携带SCT

的证书颁发给服务器,在TLS握手时,浏览器拿到服务器给它的这份携带SCT的证书,除了验证证书本身,

还要向日志服务验证 SCT.

第一回合

客户→服务器:你好.

服务器→客户:你好,我是服务器.

客户→服务器:XXX

因为消息是在网络上传输的,有人可以冒充自己是服务器来向客户发送信息.例如上面的信息可以被黑客截获如下:

客户→服务器:你好

服务器→客户:你好,我是服务器

客户→黑客:你好 //黑客在客户和服务器之间的某个路由器上截获客户发给服务器的信息,然后自己冒充服务器.

黑客→客户:你好,我是服务器.

因此客户在接到消息后并不能肯定这个消息就是由服务器发出的,某些黑客也可以冒充服务器发出这个消息.如何确定信息是由服务器发过来的呢?

有一个解决方法,因为只要服务器有私钥,所以只要能够确认对方有私钥,那么对方就是服务器.

因此通信过程可以改进如下:

客户→服务器:你好.

服务器→客户:你好,我是服务器.

客户→服务器:向我证明你就是服务器.

服务器→客户:你好,我是服务器{你好,我是服务器}[私钥丨RSA]

注意:这里约定,{}表示RSA加密后的内容.

[丨]表示用什么密钥和加密算法进行加密,后面的例子中都用这种表示方式,

例如上面的{你好,我是服务器}[私钥丨RSA]就表示用私钥对"你好,我是服务器"进行加密后的结果.

为了向客户证明自己是服务器,服务器把一个字符串用私钥加密,把明文和加密后的密文一起发给客户.

对于这里的例子来说,就是把字符串"你好,我是服务器"和这个字符串用私钥加密后的内容

{你好,我是服务器}[私钥丨RSA]发给客户.

客户收到消息后,它用自己持有的公钥解密密文,和明文进行对比.如果一致,说明信息的确是由服务器发过来的.也就是说客户把{你好,我是服务器}[私钥丨RSA]这个内容用公钥进行解密,然后和"你好,我是服务器"对比.因为由服务器用私钥加密的内容,由并且只能由公钥进行解密,私钥只能由服务器持有,所以如果解密过来的信息是能够对的上的,那说明信息一定是由服务器发过来的.

让我们讨论一下建立兔子种群数量模型的问题，假如你一年有X只兔子

那么你明年会有几个兔子呢

某个时间点顾客的到达,与其他顾客的到达情况无关

在时间轴上,我们可以将其划分为多个小的时间段,在某个时间段,顾客可能会以一个

概率p进入队列,如果我们将一个时间单位划分为n个小时间段,那么我们可以预期在一个时间单位内

有$n \cdot p$个顾客到达,为了精确描述这一过程,我们选择让n趋近于无穷大,而不是离散的时间段.

为了避免遇到无穷多顾客到来的情况,

我们根据时间段数量的改变

来调整概率p,使其等于λ除以时间段数量n

在一个时间单位内,我们平均可以得到λ个顾客到来,

其中λ是一个常数.

在这段时间内,有 k 个顾客到来的概率是多少?

每个顾客到达的概率是$\frac{\lambda}{n}$,因此在这些事件段k次到达的概率是$\left(\frac{\lambda}{n}\right)^k$

而在其他时间段,没有顾客到达的情况,要乘以$\left(1-\frac{\lambda}{n}\right)^{n-k}$

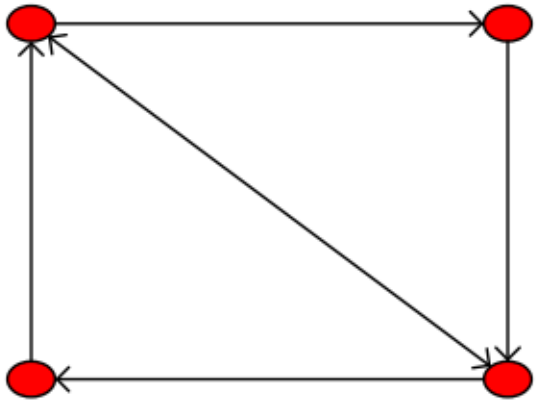
因为还有其他可能的时间段组合可以使这k个顾客到达.所以要乘C_n^k

以包括所有可能的顾客到来方式.

$p(x) = C_n^x \cdot p^x \cdot q^{n-x}$

$x = 0,1,2,\cdots,n$

$(0 < p < 1, p + q = 1)$



假设有一家餐厅只供应三种食物:汉堡,热狗,和比萨.

但是,它们遵循一个奇怪的规则,在任何一天,它们只供应三种食物中的一种,并取决于它们昨天供应的是什么.

换句话说,只要你知道了他们今天供应的是什么,有一种方法可以知道他们明天会供应什么.

例如,如果今天供应的是汉堡,那么明天供应比萨的概率是 60%,每种食物代表了一种状态.

```
from scapy.all import *
a=sniff(filter="tcp port 4444",prn=lambda x:x.sprintf("{TCP:%TCP.seq%\n}"),count=1)
ack=int([p[TCP].seq for p in a][0])+1
seq=int([p[TCP].ack for p in a][0])
sport=int([p[TCP].dport for p in a][0])
dport=int([p[TCP].sport for p in a][0])
dst=str([p[IP].src for p in a][0])
src=str([p[IP].dst for p in a][0])
send(IP(src=src,dst=dst)/
TCP(dport=dport,
    sport=sport,
    ack=ack,
    seq=seq,
    flags='R'),
    verbose=True)

import socket
s=socket.socket(…)
s.bind(…)
s.listen(…)
a=socket.socket(…)
a.connect(…)
```

三次握手在 tcp flag reset 前被发送

Ele 实验室,你来鉴赏一下我这篇文章~</>

Edit

Preferences

Protocols

TCP

Relative sequence number(相对序列号)(requires Analyze TCP sequence)(需要勾选分析 TCP 序列号)

1.gpedit.msc

2.计算机配置(computer configuration)

3.Windows 设置(Windows Settings)

4.安全设置(Security Settings)

5.本地策略(Local Policies)

6.安全选项(Security Options)

7.用户账户控制:以管理员批准模式运行所有管理员(User Account Control:Run all administrators in Admin Approval Mode)

8.禁用(Disabled)

---

cmd /c echo y | powershell Enable-PnpDevice (Get-PnpDevice -Class Camera).InstanceId

cmd /c echo y | powershell Disable-PnpDevice (Get-PnpDevice -Class Camera).InstanceId

或 devmgmt.msc

Cameras

Integrated Webcam

Disable Device

---

https://sqlitebrowser.org/dl/

File

Open Database

All files(*)

C:\Users\%username%\Appdata\Local\Microsoft\Edge\User Data\Default\History

C:\Users\%username%\Appdata\Roaming\Mozilla\Firefox\Profiles\<>.default-release\places.sqlite

Browse Data

Table(urls)(moz_origins)

---

firewall.cpl\control panel\system and security\windows defender firewall

secpol.msc\local security policy

ncpa.cpl\Control Panel\Network and Internet\Network Connections

powercfg.cpl\control panel\hardware and sound\power options(控制面板\硬件和声音\电源选项)

microsoft.windows.camera:\camera

mstsc\remote desktop connection(远程桌面连接)

powershell_ise.exe\windows powershell ISE

taskmgr\Task Manager(Windows 任务管理器)

cmd.exe\Command Prompt(命令提示符)

powershell\Windows powershell

msedge.exe\Microsoft edge

iexplore\Windows Internet Explorer

explorer.exe\File Explorer(库,文件资源管理器)

eventvwr.exe\Event Viewer

python.exe\python

gpedit.msc\Local Group Policy Editor(本地组策略编辑器)

control\control panel(控制面板)

notepad\NotePad(记事本)

ms-availablenetworks:\

ms-settings:network\NetWork Status

ms-settings:\settings

calc.exe\calculator(计算器)

services.msc\services(服务)

wsreset.exe\Microsoft Store

windowsdefender:\windows security

wmplayer\windows media player

netplwiz\User Accounts(用户账户)

shell:RecycleBinFolder\RecycleBin(回收站)

psr\steps recorder(问题步骤记录器)

conhost\command propmt(命令提示符)

osk\on screen keyboard(屏幕键盘)

DpiScaling\settings

nslookup/

compmgmt.msc/Computer Management()

diskmgmt.msc/device manager(磁盘管理)

control folders/file explorer options(文件夹选项)

virtmgmt.msc/hpyer-v manager

appwiz.cpl/Control Panel\Programs\Programs and Features(控制面板\程序\程序和功能)

shell:appsfolder\Applications

shell:thispcdesktopfolder

winver\about windows(关于 Windows)

iexpress\IExpress Wizard()

vmconnect\Virtual Machine Connection()

fsquirt\Bluetooth File Transfer()

taskschd.msc\Task scheduler()

sysdm.cpl\System Properties (系统属性)

devmgmt.msc\Device Manager(设备管理器)

考虑这样一个分类问题,平面上有两类点,有三个属于 A 类,有两个属于 B 类

G 中的全体有限阶元素构成 G 的一个子群.
