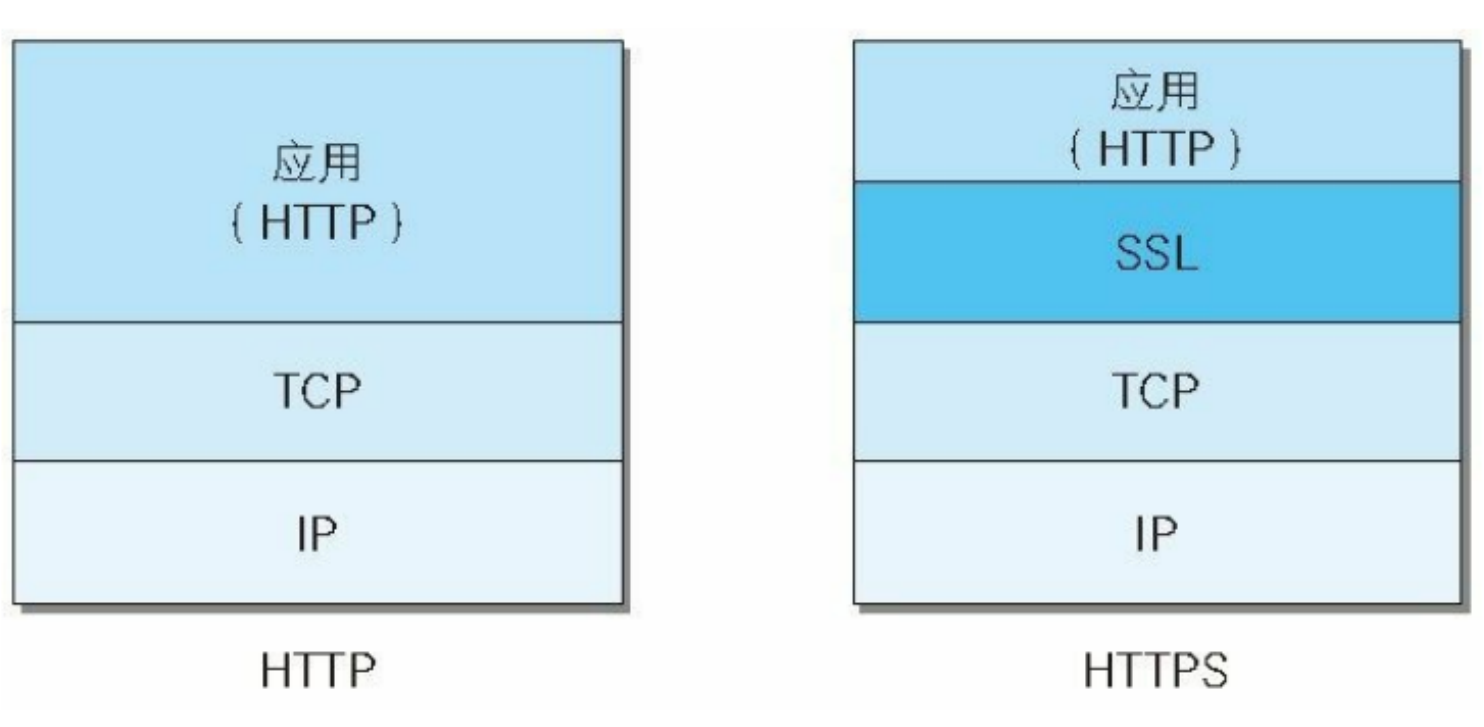
HTTPS加密（握手）过程

Artifacts 关注
2 2019.05.16 22:20:31 字数 2,690 阅读 45,934

41赞
赞赏
更多好文

参考HTTPS的加密流程|一篇文章读懂HTTPS及其背后的加密原理|HTTPS协议详解|Https加密过程|Https握手过程

`HTTPS`（全称： Hypertext Transfer Protocol Secure，`超文本传输安全协议`），是以安全为目标的HTTP通道，简单讲是`HTTP`的安全版。



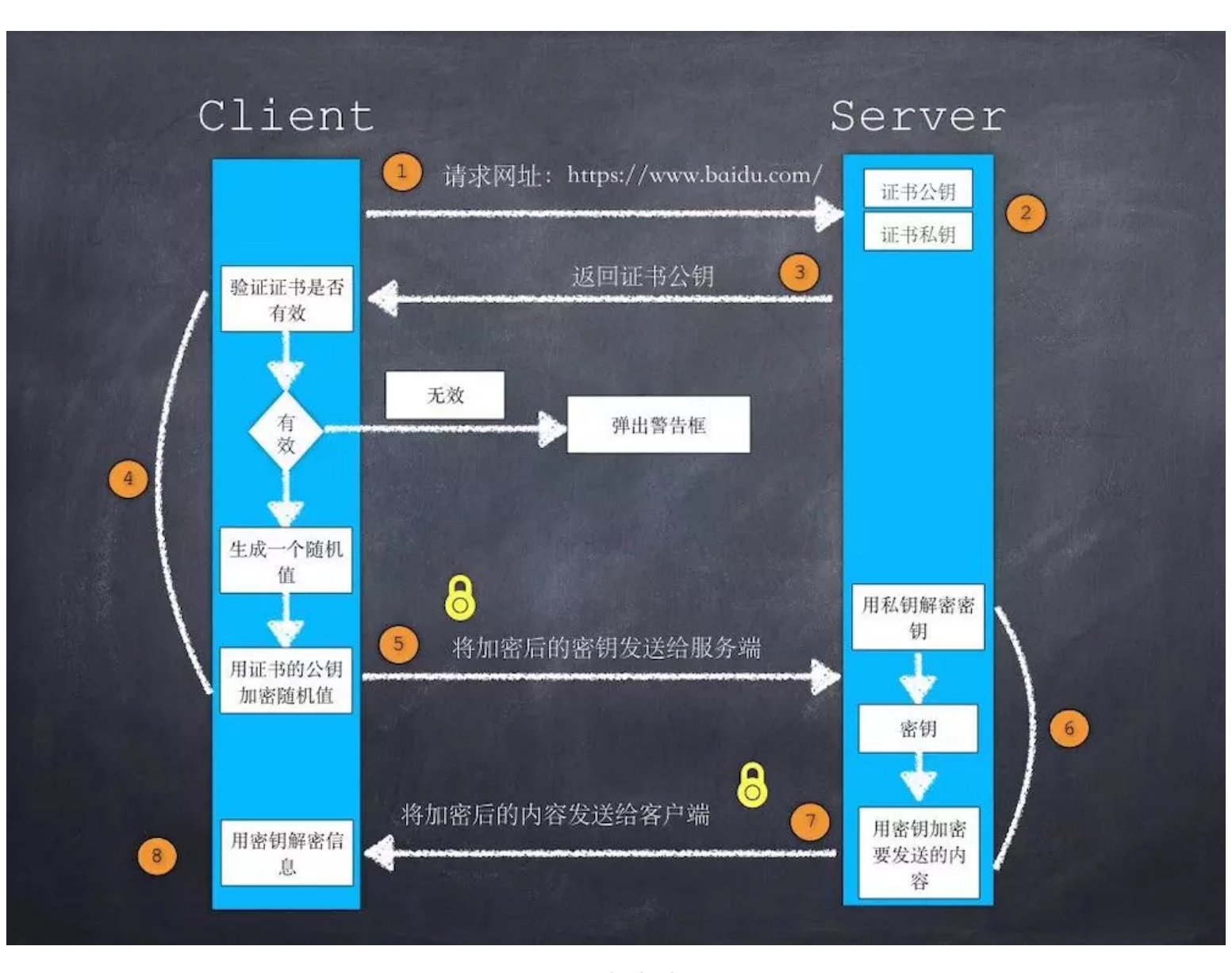
HTTP+加密+认证+完整性保护 = HTTPS

- HTTP： 直接通过明文在浏览器和服务器之间传递信息。
- HTTPS： 采用 对称加密 和 非对称加密 结合的方式来保护浏览器和服务端之间的通信安全。
  对称加密算法加密数据+非对称加密算法交换密钥+数字证书验证身份=安全

> HTTPS其实是有两部分组成：HTTP + SSL / TLS，也就是在HTTP上又加了一层处理加密信息的模块。服务端和客户端的信息传输都会通过TLS进行加密，所以传输的数据都是加密后的数据。

1. 传统的HTTP协议通信：传统的HTTP报文是直接将报文信息传输到TCP然后TCP再通过TCP套接字发送给目的主机上。
2. HTTPS协议通信：HTTPS是HTTP报文直接将报文信息传输给SSL套接字进行加密，SSL加密后将加密后的报文发送给TCP套接字，然后TCP套接字再将加密后的报文发送给目的主机，目的主机将通过TCP套接字获取加密后的报文给SSL套接字，SSL解密后交给对应进程。

具体是如何进行加密，解密，验证的，且看下图。



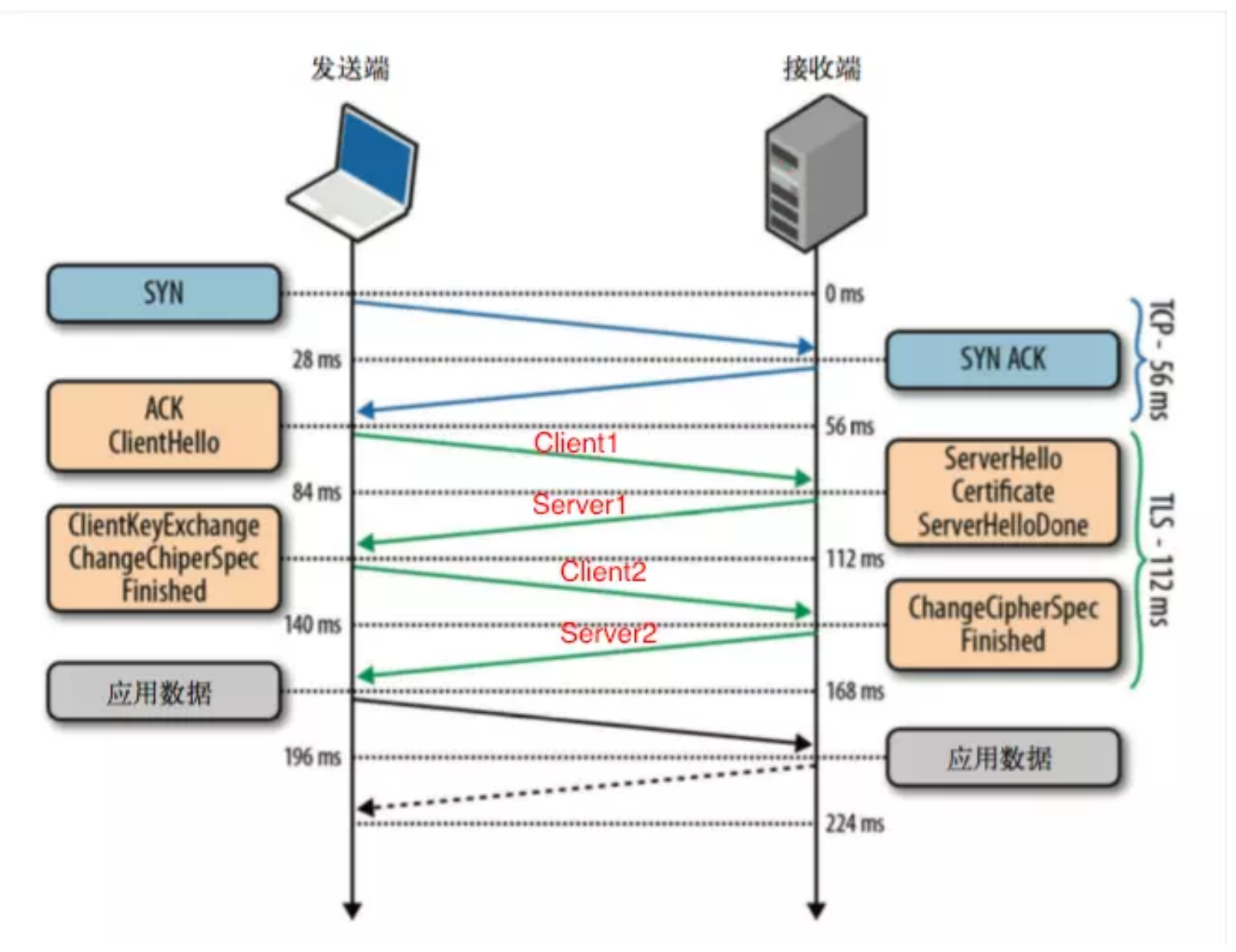
HTTPS加密过程

HTTPS加密请求（一次握手）过程

- 首先，客户端发起握手请求，以明文传输请求信息，包含版本信息，加密-套件候选列表，压缩算法候选列表，随机数，扩展字段等信息(`这个没什么好说的，就是用户在浏览器里输入一个HTTPS网址，然后连接到服务端的443端口。`)
- 服务端的配置，采用HTTPS协议的服务器必须要有一套数字证书，可以自己制作，也可以向组织申请。区别就是自己颁发的证书需要客户端验证通过，才可以继续访问，而使用受信任的公司申请的证书则不会弹出提示页面。`这套证书其实就是一对公钥和私钥。`如果对公钥不太理解，可以想象成一把钥匙和一个锁头，只是世界上只有你一个人有这把钥匙，你可以把锁头给别人，别人可以用这个锁把重要的东西锁起来，然后发给你，因为只有你一个人有这把钥匙，所以只有你才能看到被这把锁锁起来的东西。
- 服务端返回协商的信息结果，包括选择使用的协议版本 version，选择的加密套件 cipher suite，选择的压缩算法 compression method、随机数 random_S 以及证书。(`这个证书其实就是公钥，只是包含了很多信息，如证书的颁发机构，过期时间等等。`)
- 客户端验证证书的合法性，包括可信性，是否吊销，过期时间和域名。(`这部分工作是由客户端的SSL/TLS来完成的，首先会验证公钥是否有效，比如颁发机构，过期时间等等，如果发现异常，则会弹出一个警示框，提示证书存在的问题。如果证书没有问题，那么就生成一个随机值。然后用证书（也就是公钥）对这个随机值进行加密。就好像上面说的，把随机值用锁头锁起来，这样除非有钥匙，不然看不到被锁住的内容。`)
- 客户端使用公匙对对称密匙加密，发送给服务端。(`这部分传送的是用证书加密后的随机值，目的是让服务端得到这个随机值，以后客户端和服务端的通信就可以通过这个随机值来进行加密解密了。`)
- 服务器用私钥解密，拿到对称加密的密匙。(`服务端用私钥解密后，得到了客户端传过来的随机值，然后把内容通过该随机值进行对称加密，将信息和私钥通过某种算法混合在一起，这样除非知道私钥，不然无法获取内容，而正好客户端和服务端都知道这个私钥，所以只要加密算法够彪悍，私钥够复杂，数据就够安全。`)
- 传输加密后的信息，这部分信息就是服务端用私钥加密后的信息，可以在客户端用随机值解密还原。
- 客户端解密信息，客户端用之前生产的私钥解密服务端传过来的信息，于是获取了解密后的内容。整个过程第三方即使监听到了数据，也束手无策。

加密

客户端和服务端之间的加密机制：



TLS握手

TLS协议是基于TCP协议之上的，图中第一个蓝色往返是TCP的握手过程，之后两次橙色的往返，我们可以叫做TLS的握手。握手过程如下：

- client1：TLS版本号+所支持加密套件列表+希望使用的TLS选项

- Server1:选择一个客户端的加密套件+自己的公钥+自己的证书+希望使用的TLS选项+（要求客户端证书）；

- Client2:(自己的证书)+使用服务器公钥和协商的加密套件加密一个对称秘钥（自己生成的一个随机值）；

- Server2:使用私钥解密出对称秘钥（随机值）后，发送加密的Finish消息，表明完成握手

这里可能要提一下什么是对称加密和非对称加密：
一般的对称加密像这样：

```
encrypt(明文，秘钥) = 密文
decrypt(密文，秘钥) = 明文
```

> 共享密钥加密也称对称密钥加密。采用的是使用相同密钥对报文进行加密解密
> 我们可以将共享密钥加密这样理解：我们把我们要给别人的东西放到一个箱子里面，
> 然后给箱子上了一把锁。当箱子到了我们想给的那个人
> 身上时，他也需要这把钥匙才能开锁。
> 这样就产生了一个问题了，我们怎么将这把钥匙安全的交给对方呢，如果钥匙在半路
> 被人截取了，那么对箱子有没有加锁有什么区别呢。因此
> 共享密钥加密需要解决的一个大问题就是如何安全的将密钥交给解密方。

也就是说加密和解密用的是同一个秘钥。而非对称加密是这样的：

```
encrypt(明文，公钥) = 密文
decrypt(密文，私钥) = 明文
```

> 假设客户发送报文，服务器接收报文。客户在发送报文的时候需要对报文加密，服务器有两把密钥，一把私钥，一把公钥。客户在发送报文前需要先向服务器获取公钥进行报文的加密。这把公钥只能加密，不能解密，因此被任何人截获到都没什么用处。服务器收到加密后的报文，使用私钥解密。整个过程中只涉及到公钥的获取，加密，密文传输，解密。并不涉及到解密用的私钥传输。因此这种方式是安全的。但是涉及到太多细节，整个流程
> 下来耗时耗费资源。
> 再拿上面那个例子，这时候我们还想把一些东西锁到箱子里给某人，我们称他为大傻。我们先跟大傻联系，大傻身上有两把钥匙，我们称为钥匙A和钥匙B。钥匙A可以用来造锁，但是造好的锁自己却不能开，只能通过钥匙B来开。跟大傻取得联系后大傻把钥匙A给我们。我们拿着钥匙A找造锁师傅造了一把锁，并且给箱子上锁。然后将带锁的箱子通过物流发给大傻，就算钥匙A被强盗截取了，强盗也开不了箱子。大傻收到箱子后使用那把钥匙B进行开锁，拿到东西。由于钥匙B一直再大傻身上，所以不用担心被人拿走。

加密和解密是需要不同的秘钥的。
经过这几次握手成功后，客服端和服务端之间通信的加密算法和所需要的密钥也就确定下来了，之后双方的交互都可以使用对称加密算法加密了。

HTTPS为了追求性能，又要保证安全，采用了共享密钥加密和公开密钥加密混合的方式进行报文传输。
还是拿上面的锁和箱子的例子来说明。现在我们嫌弃每次加锁都要造个新的锁效率太慢了。我们现在有两个箱子，一个箱子用于方我们要给大傻的东西，并且

这个箱子加上了锁。另一个箱子用于存放那把锁的钥匙。我们这时候找大傻拿到钥匙A造了一把锁后将那个存放钥匙的箱子锁起来，然后将这个箱子给大傻，

大傻拿到箱子使用钥匙B开锁拿到钥匙。这时候我们将那个存放了东西的箱子给大傻，大傻就可以通过这把钥匙开锁拿到东西了。这样以后我们就可以一直通过

这把锁和箱子互相给东西了，而不用发一次数据造一次锁了。

就是说采用共有密钥加密方式传输共享密钥，当共享密钥安全到达服务端后往后的数据就都采用该密钥进行加密解密。

41人点赞 >　　网络基础 ···

Artifacts
总资产15 共写了20.8W字 获得177个赞 共66个粉丝 关注

写下你的评论...

精彩评论 1

Systemerrprint
13楼 2020.11.02 09:51

其实很简单 用非对称加密加密对称加密的秘钥发出 私钥解密密文得到对称加密的秘钥 这样就没问题了 采用非对称加密因为安全性 采用对称加密是因为他加解密速度快 必须要知道这个才能更好理解

5 回复

85164633f233
04.19 13:14

1. 客户端获取服务端的公钥, 对对称加密的秘钥进行加密, 传给服务端.

2. 客户端采用对称加密 加密数据, 并传给服务端,

3.服务端用之前的传来的对称加密密文 进行解密,

大佬 是这样的把 ...

回复

添加新评论

全部评论 5

只看作者　　按时间倒序　按时间正序

Systemerrprint
13楼 2020.11.02 09:51

其实很简单 用非对称加密加密对称加密的秘钥发出 私钥解密密文得到对称加密的秘钥 这样就没问题了 采用非对称加密因为安全性 采用对称加密是因为他加解密速度快 必须要知道这个才能更好理解

5 回复

85164633f233
04.19 13:14

1. 客户端获取服务端的公钥, 对对称加密的秘钥进行加密, 传给服务端.

2. 客户端采用对称加密 加密数据, 并传给服务端,

3.服务端用之前的传来的对称加密密文 进行解密,

大佬 是这样的把 ...

回复

添加新评论

Issho
12楼 2020.10.22 14:55

非对称加密，公钥加密只能通过对应的私钥解密，私钥加密只能通过对应的公钥解密。没记错的话好像是这样的

赞 回复

Hello_Kugou
11楼 2020.09.27 18:04

，服务器有两把密钥，一把私钥，一把公钥。客户在发送报文前需要先向服务器获取公钥进行报文的加密。这把公钥只能加密，不能解密，因此被任何人截获到都没什么用处。楼主非对称性加密公钥也是可以解密私钥加密的数据的， 所以文中说的，公钥只能加密不能解密不正确😛

赞 回复

Systemerrprint
2020.11.02 09:55

获取公钥不是为了报文的加密而是要加密自己的对称加密秘钥 将自己的对称加密秘钥通过非对称加密的方式发给服务端 这样之后的数据都是对称加密的方式传输 但是传输秘钥的过程就相当于被屏蔽了 这样采取的数据相对就比较安全了

回复

添加新评论

被以下专题收入，发现更多相似内容

+收入我的专题　2021面试学习　Https　计算机原理　加密类

推荐阅读
HTTPS加密(握手)过程
阅读 501
「HTTPS」如何通俗易懂的给你讲明白HTTPS?
阅读 251
SSL/TSL通过四次握手，主要交换三个信息
阅读 166
HTTPS 原理解析
阅读 725
图解HTTP阅读笔记4 HTTPS
阅读 111
转义符