

## 24 | 固若金汤的根本（上）：对称加密与非对称加密

2019-07-22 Chrono

《透视HTTP协议》

课程介绍 >



讲述：Chrono

时长 10:14 大小 11.73M



在上一讲中，我们初步学习了 HTTPS，知道 HTTPS 的安全性是由 TLS 来保证的。

你一定很好奇，它是怎么为 HTTP 增加了机密性、完整性，身份认证和不可否认等特性的呢？

先说说机密性。它是信息安全的基础，缺乏机密性 TLS 就会成为“无水之源”“无根之木”。

实现机密性最常用的手段是“**加密**”（encrypt），就是把消息用某种方式转换成谁也看不懂的乱码，只有掌握特殊“**钥匙**”的人才能再转换出原始文本。

领资料

这里的“**钥匙**”就叫做“**密钥**”（key），加密前的消息叫“**明文**”（plain text/clear text），加密后的乱码叫“**密文**”（cipher text），使用密钥还原明文的过程叫“**解密**”（decrypt），是加密的反操作，加密解密的操作过程就是“**加密算法**”。



所有的加密算法都是公开的，任何人都可以去分析研究，而算法使用的“密钥”则必须保密。那么，这个关键的“密钥”又是什么呢？

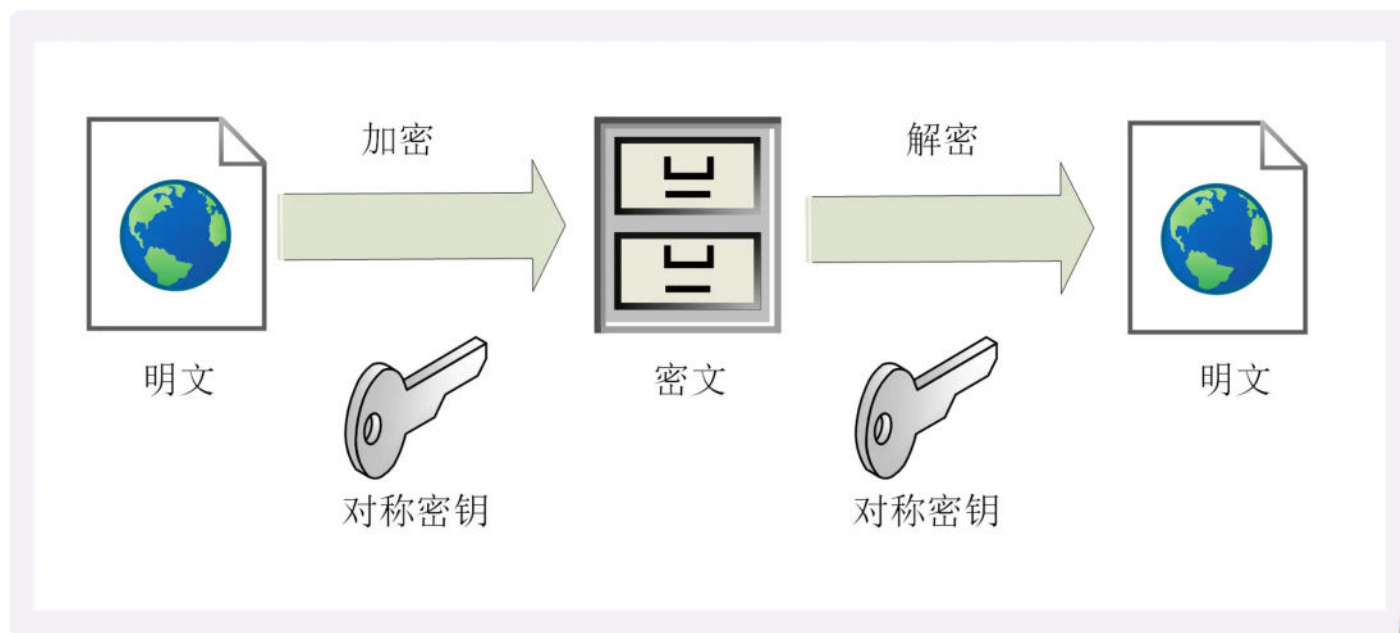
由于 HTTPS、TLS 都运行在计算机上，所以“密钥”就是一长串的数字，但约定俗成的度量单位是“位”（bit），而不是“字节”（byte）。比如，说密钥长度是 128，就是 16 字节的二进制串，密钥长度 1024，就是 128 字节的二进制串。

按照密钥的使用方式，加密可以分为两大类：**对称加密**和**非对称加密**。

## 对称加密

“对称加密”很好理解，就是指加密和解密时使用的密钥都是同一个，是“对称”的。只要保证了密钥的安全，那整个通信过程就可以说具有了机密性。

举个例子，你想要登录某网站，只要事先和它约定好使用一个对称密钥，通信过程中传输的全是用密钥加密后的密文，只有你和网站才能解密。黑客即使能够窃听，看到的也只是乱码，因为没有密钥无法解出明文，所以就实现了机密性。



TLS 里有非常多的对称加密算法可供选择，比如 RC4、DES、3DES、AES、ChaCha20 等，但前三种算法都被认为是不安全的，通常都禁止使用，目前常用的只有 AES 和 ChaCha20。

AES 的意思是“高级加密标准”（Advanced Encryption Standard），密钥长度可以是 128、192 或 256。它是 DES 算法的替代者，安全强度很高，性能也很好，而且有的硬件还会做特

领资料



殊优化，所以非常流行，是应用最广泛的对称加密算法。

ChaCha20 是 Google 设计的另一种加密算法，密钥长度固定为 256 位，纯软件运行性能要超过 AES，曾经在移动客户端上比较流行，但 ARMv8 之后也加入了 AES 硬件优化，所以现在不再具有明显的优势，但仍然算得上是一个不错的算法。

## 加密分组模式

对称算法还有一个“**分组模式**”的概念，它可以让算法用固定长度的密钥加密任意长度的明文，把小秘密（即密钥）转化为大秘密（即密文）。

最早有 ECB、CBC、CFB、OFB 等几种分组模式，但都陆续被发现有安全漏洞，所以现在基本都不怎么用了。最新的分组模式被称为 AEAD（Authenticated Encryption with Associated Data），在加密的同时增加了认证的功能，常用的是 GCM、CCM 和 Poly1305。

把上面这些组合起来，就可以得到 TLS 密码套件中定义的对称加密算法。

比如，AES128-GCM，意思是密钥长度为 128 位的 AES 算法，使用的分组模式是 GCM；ChaCha20-Poly1305 的意思是 ChaCha20 算法，使用的分组模式是 Poly1305。

你可以用实验环境的 URI“/24-1”来测试 OpenSSL 里的 AES128-CBC，在 URI 后用参数“key”“plain”输入密钥和明文，服务器会在响应报文里输出加密解密的结果。

复制代码

```
1 https://www.chrono.com/24-1?key=123456
2
3 algo  = aes_128_cbc
4 plain = hello openssl
5 enc   = 93a024a94083bc39fb2c2b9f5ce27c09
6 dec   = hello openssl
```

领资料

## 非对称加密

对称加密看上去好像完美地实现了机密性，但其中有一个很大的问题：如何把密钥安全地传递给对方，术语叫“**密钥交换**”。



因为在对称加密算法中只要持有密钥就可以解密。如果你和网站约定的密钥在传递途中被黑客窃取，那他就可以在之后随意解密收发的数据，通信过程也就没有机密性可言了。

这个问题该怎么解决呢？

你或许会说：“把密钥再加密一下发过去就好了”，但传输“加密密钥的密钥”又成了新问题。这就像是“鸡生蛋、蛋生鸡”，可以无限递归下去。只用对称加密算法，是绝对无法解决密钥交换的问题的。

所以，就出现了非对称加密（也叫公钥加密算法）。

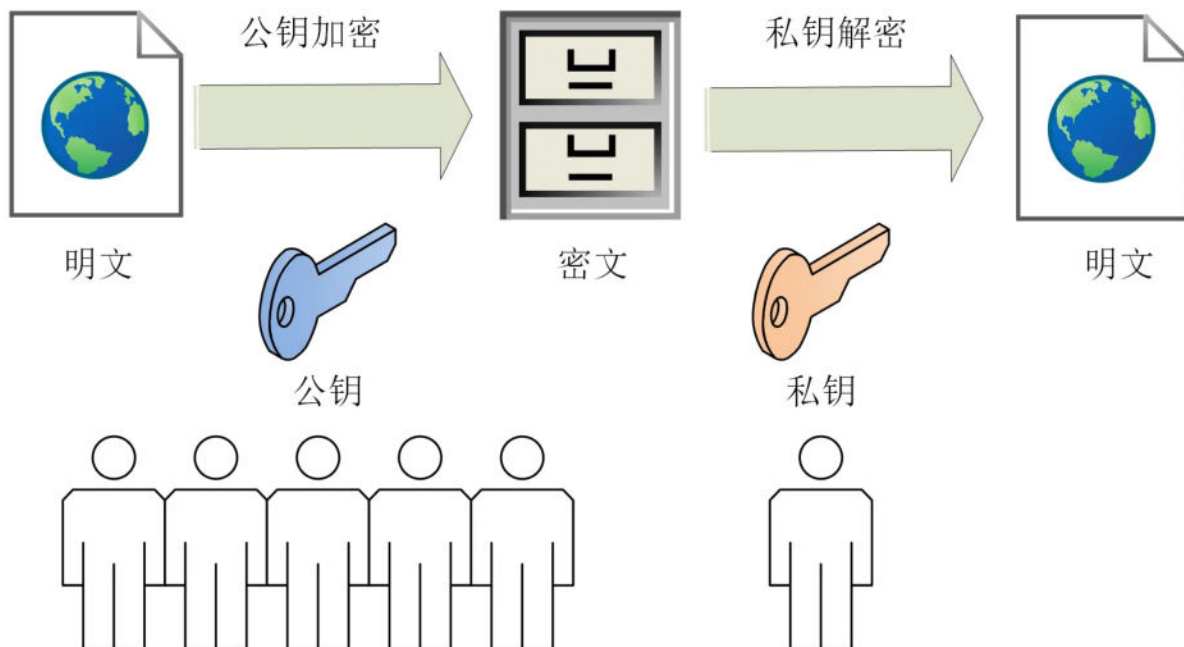
它有两个密钥，一个叫“**公钥**”（public key），一个叫“**私钥**”（private key）。两个密钥是不同的，“不对称”，公钥可以公开给任何人使用，而私钥必须严格保密。

公钥和私钥有个特别的“**单向**”性，虽然都可以用来加密解密，但公钥加密后只能用私钥解密，反过来，私钥加密后也只能用公钥解密。

非对称加密可以解决“密钥交换”的问题。网站秘密保管私钥，在网上任意分发公钥，你想要登录网站只要用公钥加密就行了，密文只能由私钥持有者才能解密。而黑客因为没有私钥，所以就无法破解密文。

领资料





非对称加密算法的设计要比对称算法难得多，在 TLS 里只有很少的几种，比如 DH、DSA、RSA、ECC 等。

RSA 可能是其中最著名的一个，几乎可以说是非对称加密的代名词，它的安全性基于“**整数分解**”的数学难题，使用两个超大素数的乘积作为生成密钥的材料，想要从公钥推算出私钥是非常困难的。

10 年前 RSA 密钥的推荐长度是 1024，但随着计算机运算能力的提高，现在 1024 已经不安全，普遍认为至少要 2048 位。

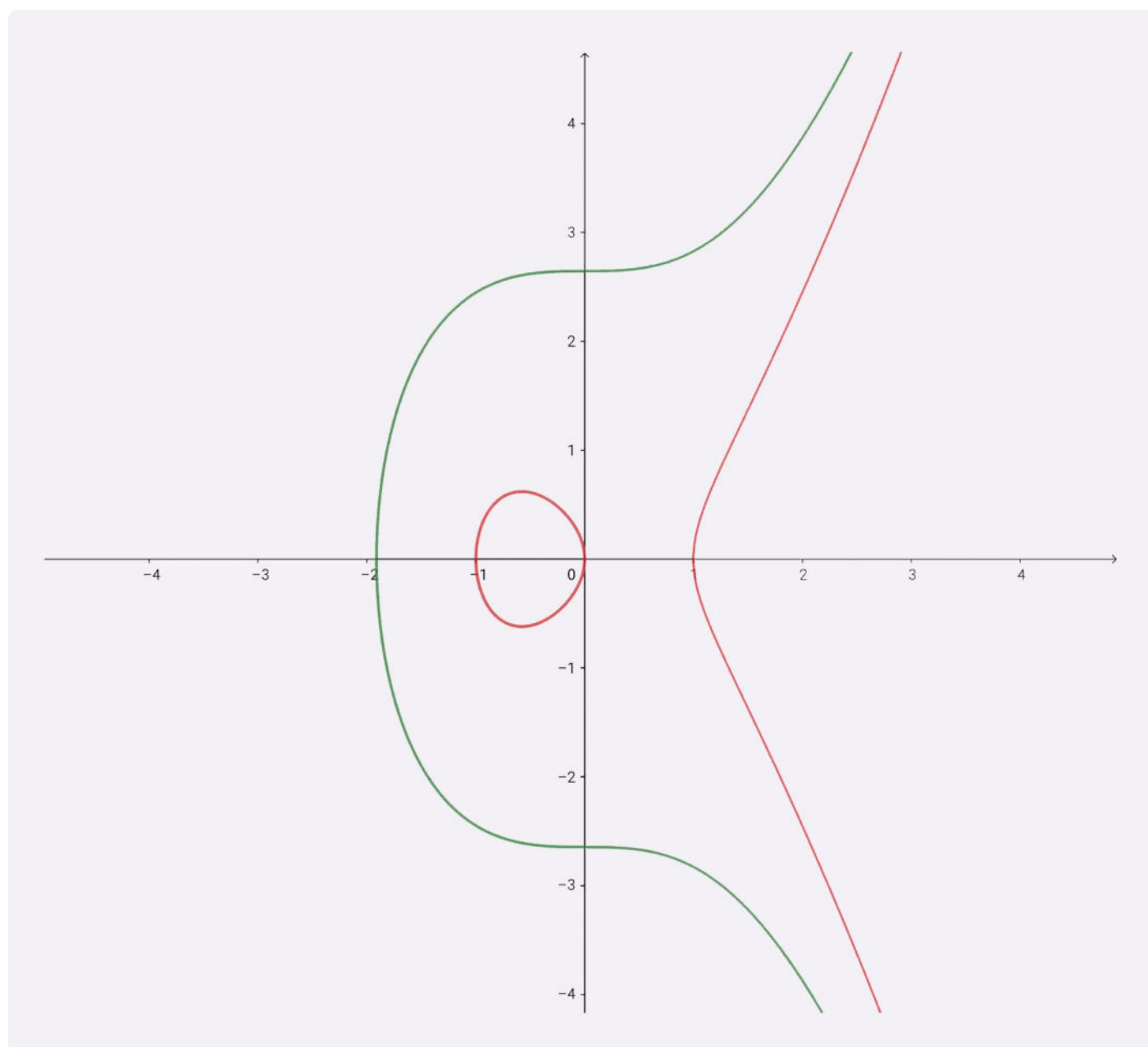
ECC (Elliptic Curve Cryptography) 是非对称加密里的“后起之秀”，它基于“**椭圆曲线离散对数**”的数学难题，使用特定的曲线方程和基点生成公钥和私钥，子算法 ECDHE 用于密钥交换，ECDSA 用于数字签名。

目前比较常用的两个曲线是 P-256 (secp256r1, 在 OpenSSL 称为 prime256v1) 和 x25519。P-256 是 NIST (美国国家标准技术研究所) 和 NSA (美国国家安全局) 推荐的曲线，而 x25519 被认为是最安全、最快速的曲线。

领资料



ECC 名字里的“椭圆”经常会引起误解，其实它的曲线并不是椭圆形，只是因为方程很类似计算椭圆周长的公式，实际的形状更像抛物线，比如下面的图就展示了两个简单的椭圆曲线。



两个简单的椭圆曲线： $y^2=x^3+7$ ， $y^2=x^3-x$

比起 RSA，ECC 在安全强度和性能上都有明显的优势。160 位的 ECC 相当于 1024 位的 RSA，而 224 位的 ECC 则相当于 2048 位的 RSA。因为密钥短，所以相应的计算量、消耗的内存和带宽也就少，加密解密的性能就上去了，对于现在的移动互联网非常有吸引力。

实验环境的 URI“/24-2”演示了 RSA1024，你在课后可以动手试一下。

## 混合加密


领资料



看到这里，你是不是认为可以抛弃对称加密，只用非对称加密来实现机密性呢？

很遗憾，虽然非对称加密没有“密钥交换”的问题，但因为它们都是基于复杂的数学难题，运算速度很慢，即使是 ECC 也要比 AES 差上好几个数量级。如果仅用非对称加密，虽然保证了安全，但通信速度有如乌龟、蜗牛，实用性就变成了零。

实验环境的 URI“/24-3”对比了 AES 和 RSA 这两种算法的性能，下面列出了一次测试的结果：

 复制代码

```
1 aes_128_cbc enc/dec 1000 times : 0.97ms, 13.11MB/s
2
3 rsa_1024 enc/dec 1000 times : 138.59ms, 93.80KB/s
4 rsa_1024/aes ratio = 143.17
5
6 rsa_2048 enc/dec 1000 times : 840.35ms, 15.47KB/s
7 rsa_2048/aes ratio = 868.13
```

可以看到，RSA 的运算速度是非常慢的，2048 位的加解密大约是 15KB/S（微秒或毫秒级），而 AES128 则是 13MB/S（纳秒级），差了几百倍。

那么，是不是能够把对称加密和非对称加密结合起来呢，两者互相取长补短，即能高效地加密解密，又能安全地密钥交换。

这就是现在 TLS 里使用的**混合加密**方式，其实说穿了也很简单：

在通信刚开始的时候使用非对称算法，比如 RSA、ECDHE，首先解决密钥交换的问题。

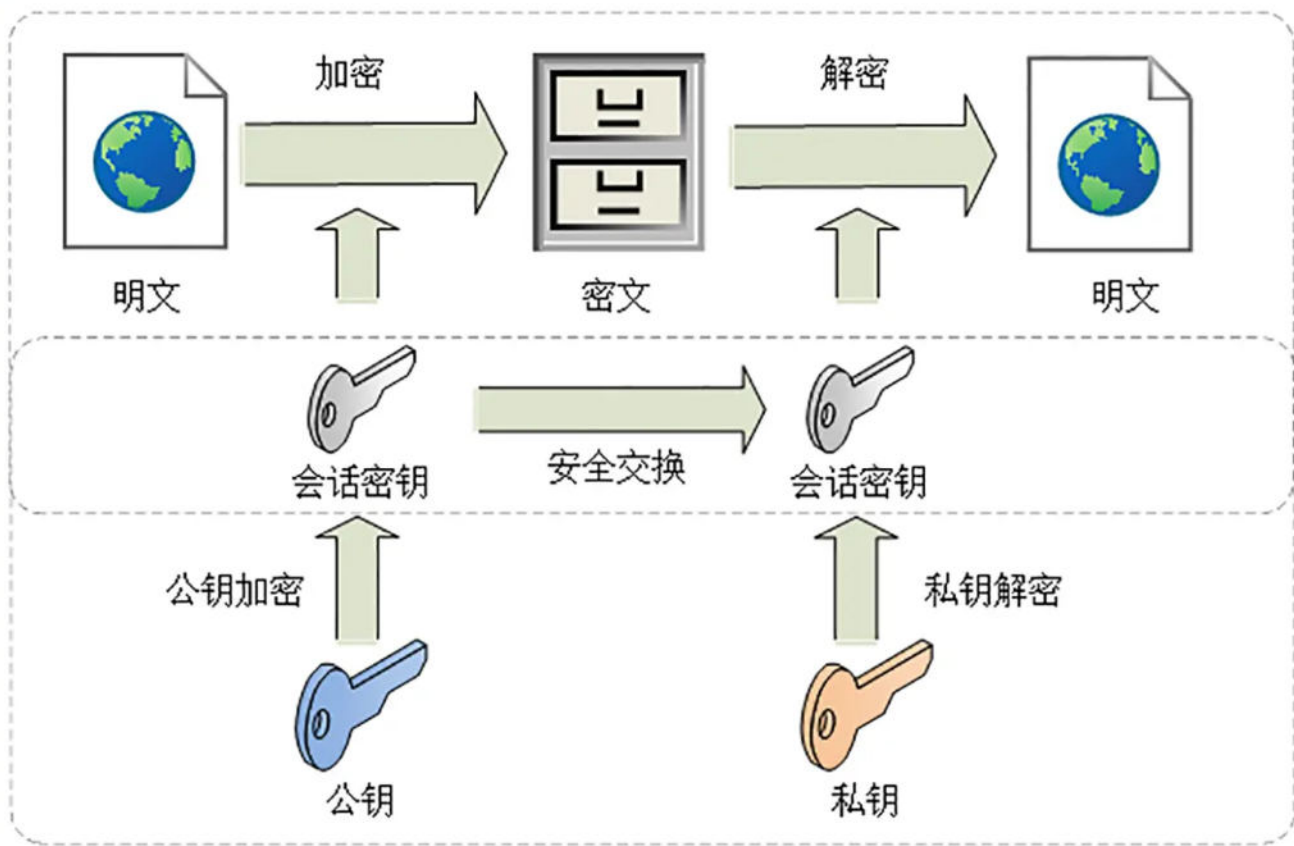
然后用随机数产生对称算法使用的“**会话密钥**”（session key），再用公钥加密。因为会话密钥很短，通常只有 16 字节或 32 字节，所以慢一点也无所谓。

对方拿到密文后用私钥解密，取出会话密钥。这样，双方就实现了对称密钥的安全交换，后续就不再使用非对称加密，全都使用对称加密。

领资料







这样混合加密就解决了对称加密算法的密钥交换问题，而且安全和性能兼顾，完美地实现了机密性。

不过这只是“万里长征的第一步”，后面还有完整性、身份认证、不可否认等特性没有实现，所以现在的通信还不是绝对安全，我们下次再说。

## 小结

1. 加密算法的核心思想是“把一个小秘密（密钥）转化为一个大秘密（密文消息）”，守住了小秘密，也就守住了大秘密；
2. 对称加密只使用一个密钥，运算速度快，密钥必须保密，无法做到安全的密钥交换，常用的有 AES 和 ChaCha20；
3. 非对称加密使用两个密钥：公钥和私钥，公钥可以任意分发而私钥保密，解决了密钥交换问题但速度慢，常用的有 RSA 和 ECC；
4. 把对称加密和非对称加密结合起来就得到了“又好又快”的混合加密，也就是 TLS 里使用的加密方式。

领资料



## 课下作业



1. 加密算法中“密钥”的名字很形象，你能试着用现实中的锁和钥匙来比喻一下吗？
2. 在混合加密中用到了公钥加密，因为只能由私钥解密。那么反过来，私钥加密后任何人都可以用公钥解密，这有什么用呢？

欢迎你把自己的学习体会写在留言区，与我和其他同学一起讨论。如果你觉得有所收获，也欢迎把文章分享给你的朋友。



## == 课外小贴士 ==

- 01 严格来说对称加密算法还可以分为块加密算法 (block cipher) 和流加密算法 (stream cipher)，DES、AES 等属于块加密，而 RC4、ChaCha20 属于流加密。
- 02 ECC 虽然定义了公钥和私钥，但不能直接实现密钥交换和身份认证，需要搭配 DH、DSA 等算法，形成专门的 ECDHE、ECDSA。RSA 比较特殊，本身即支持密钥交换也支持身份认证。
- 03 比特币、以太坊等区块链技术里也用到了 ECC，它们选择的曲线是 secp256k1。
- 04 由于密码学界普遍不信任 NIST 和 NSA，怀疑

领资料




secp 系列曲线有潜在的弱点，所以研究出了“x25519”，它的名字来源于曲线方程里的参数“ $2^{255} - 19$ ”。另有一个更高强度的曲线“x448”，参数是“ $2^{448} - 2^{224} - 1$ ”。

05 在 Linux 上可以使用 OpenSSL 的命令行工具来测试算法的加解密速度，例如“openssl speed aes”“openssl speed rsa2048”等。

06 TLS1.2 要求必须实现 TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA，TLS1.3 要求必须实现 TLS\_AES\_128\_GCM\_SHA256，并且因为前向安全的原因废除了 DH 和 RSA 密钥交换算法。

分享给需要的人，Ta订阅超级会员，你将得 50 元

Ta单独购买本课程，你将得 20 元

 生成海报并分享

领资料

 赞 27

 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。



上一篇 23 | HTTPS是什么？SSL/TLS又是什么？

## 学习推荐

# JVM + NIO + Spring

各大厂面试题及知识点详解

限时免费



### 精选留言 (76)

写留言



xiaolin777

2019-09-19

老师，今天面试官问我非对称加密为什么慢，非对称加密除了慢外还有什么缺点，您能帮我解答一下吗？

作者回复: 非对称加密基于大数运算，比如大素数或者椭圆曲线，是复杂的数学难题，所以消耗计算量，运算速度慢。

除了慢，可能还有一个缺点就是需要更多的位数，相同强度的对称密钥要比非对称密钥短。

对称密钥一般都128位、256位，而rsa一般要2048位，不过椭圆曲线的会短一点。



57

领资料



TerryGoForIt

2019-07-22

简单来说，SSL 就是通信双方通过非对称加密协商出一个用于对称加密的密钥。

作者回复: ✓



**-W.LI-**

2019-07-22

混合加密:用非对称加密，加密对称加密的私钥。对称加密的私钥又是会话级的随机数=一次会话一个私钥。就算别人baoli破解也只是破解了一个会话。

作者回复: ✓



29

**鱼向北游**

2019-07-22

私钥加密用公钥解是为了做身份认证，不可抵赖，因为默认私钥只有持有人知道

作者回复: ✓

共 2 条评论 &gt;

25

**啦啦啦**

2019-12-12

今天刚看了另外一个课程的老师讲的，密钥，在这个词里念mi yue

作者回复: 应该是钥匙的“yao”吧。

我搜了一下。

【mì yuè】读音下的“密钥”的意思：紧密的锁闭。这里的用法用了“密钥”的动词性质。

【mì yào】读音下的“密钥”的意思：密码学中的专有名词，指解密所需要的特殊代码。这里用了“密钥”的名词性。

密钥现代用的最多的是密码学中的意思，在英文中解释为key，中文意思偏向于钥匙。在密码学中，特别是公钥密码体系中，密钥的形象描述往往是房屋或者保险箱的钥匙。因此在技术词典中，密钥被标注为【mì yào】。

在一些词典中原来把密钥标注为【mì yuè】，由于权威性带来了一些影响，所以也有很多人把密钥念作【mì yuè】。由于钥在读作【yuè】也可以作“钥匙”的解释。但是钥被念作【yuè】时，往往偏向于钥的动词性，这种性质就跟“血”的用法相似。

共 5 条评论 &gt;

21

**Shopee内推码: NTAGx...**

2019-07-22

领资料



加密的分组模式，是怎么实现的，具体怎么做，这块不是很理解。方便老师指导下不

作者回复: 拿ECB来举例子，假设使用aes128，密钥长度是16字节，那么就把明文按16字节分组，然后每个分组用密钥加密。

其他的cbc、ofb等的方法类似，但细节不同，例如cbc增加了初始向量。



19



Geek\_66666

2019-08-25

在实际传输过程中，大家（包括其他人）怎么知道双方通信用的哪个公钥，是传输过程公开的，大家都可以获取看到的吗？如果大家知道了，用私钥加密的内容都能被别人用公钥解密，这部分内容是不是不安全？

作者回复: 公钥的传输必须使用证书，把公钥和持有者身份绑在一起，否则就会有信任问题，容易被别人冒充。

私钥加密的作用是签名，实现身份认证而不是数据保密，签名是公开的，所以不存在安全问题。

共 2 条评论 >

16



青莲居士

2019-08-02

老师，你好，我不明白对称加密为啥会有密钥交换的过程，对称加密就一个密钥，客户端服务端各保存一份就可以了，为啥要传输交换呢？

作者回复: 关键是“如何各保存一份”，两边加密通信必须要使用相同的密钥才行，不交换如何才能保持一致呢？

而且简单的一对一还好说，现实情况是网站要面对成千上万的用户，如何与这么多的客户端保持一致？

还有，如果总使用一个密钥，就很容易被破解，风险高，需要定期更换，最好是一次一密。

所以，为了安全起见，每次通信前双方都要交换密钥，这样就实现了“各保存一份”，用完就扔掉，下次重新交换。

共 4 条评论 >

14



Fstar

领资料





## 思考题第1题：

（这里举个比较勉强的例子）假设 a 持有私钥，b 持有公钥，然后他们用一个加了锁的盒子进行通信。

1. a 把信件放到盒子里，然后用一排连接为锁链的锁将盒子锁起来，然后寄给 b。只要公钥能解开其中一个锁，那对方就能拿到信件。（可能换成能识别具有某些特征密码的密码锁的比喻会更好一些）
2. b 用公钥开锁拿到了信件，然后他写了一封回信，同样放到盒子里，然后挂上一个只有私钥才能打开的锁，寄给 a。
3. 只有 a 有对应的钥匙（私钥），于是 a 拿到了回信。

思考题第2题不是很清楚题意，大概是问只要有公钥就能解密，私钥有什么意义？

答：虽然任何公钥都可以对私钥加密的数据解密，但这个解密后的数据如果是某个公钥持有人用自己私有的密钥加密（对称加密）后的加密数据，那其他人拿到是加密后的数据，无法得到真正的数据，于是可以保证机密性。

作者回复：回答的很认真。

第二个问题，问的是私钥加密公钥解密有什么作用，能够干什么。

因为私钥只能由一个人秘密持有，所以它加密的数据谁都可以解密，没有私密性，但这就是它的价值所在，可以证明这个数据就是私钥持有人发布的，可以用来做身份认证。



15



钱

2020-03-30

1：加密算法中“密钥”的名字很形象，你能试着用现实中的锁和钥匙来比喻一下吗？

没有锁的大门是不安全的，谁想进就进，谁想拿的啥就拿点啥，就好像在网络中奔跑的HTTP报文。

为了安全性需要给明文加密，同样为了安全性需要给大门上锁，加密的明文变成了密文，没有解密是看不懂的，没有钥匙的人打不开锁是进不了门的。

一把钥匙一把锁，其他钥匙开不了，这就是对称加密。一把钥匙锁门，N把其他的钥匙都能打开，这就是非对称加密，那那把锁门的钥匙有啥用呢？可以证明这个院子是我的，证明锁是我上的。

2：在混合加密中用到了公钥加密，因为只能由私钥解密。那么反过来，私钥加密后任何人都可以用公钥解密，这有什么用呢？

第一眼感觉好像没啥用，后来发现可以凭支票去银行取钱。数字签名和身份认证，也是相当有

领资料



用。

这节很有意思。

对称加密、非对称加密、混合加密，对于加密二字很容易理解，对于明文都能看懂，加密就是通过一些步骤把明文变密文，让人看不懂，只有使用密钥解密一下，密文变明文了大家又都能看懂了。

那啥是对称？啥是非对称？啥是混合？

对称强调A钥匙加密只有A钥匙能解码

非对称强调A钥匙加密后只有B钥匙能解密，B钥匙加密后只有A钥匙能解码。这个感觉好神奇，就好像我上锁后谁都能打开，上锁是没用的只能证明锁是我上的。

混合强调非对称和对称的组合使用，非对称用于密钥交换，对称用于数据的安全传输。

作者回复: 总结的很好，非常认真，amazing!

共 3 条评论 >

👍 10



**Geek\_steven\_wang**

2019-08-24

分组模式：DES和AES都属于分组密码，它们只能加密固定长度的明文。如果需要加密任意长度的明文，就需要对分组密码进行迭代，而分组密码的迭代方法就称为分组密码的“模式”。

主要模式：

ECB模式：Electronic Code Book mode（电子密码本模式）

CBC模式：Cipher Block Chaining mode（密码分组链接模式）（推荐使用）

CFB模式：Cipher FeedBack mode（密文反馈模式）

OFB模式：Output FeedBack mode（输出反馈模式）

CTR模式：CounTeR mode（计数器模式）（推荐使用）

作者回复: 补充的非常好。

共 2 条评论 >

👍 9



**allen**

2019-08-23

非对称加密通信的时候是不是会互相把自己的公钥发给对方？

作者回复: 是的，公钥的分发通常都使用证书的形式，防止伪造。



👍 9



**Demon.Lee** 🏆

2019-10-19

领资料





非对称加密可以解决“密钥交换”的问题。网站秘密保管私钥，在网上任意分发公钥，你想要登录网站只要用公钥加密就行了，密文只能由私钥持有者才能解密。而黑客因为没有私钥，所以就无法破解密文。

有留言说：非对称加密通信的时候是互相把自己的公钥发给对方

老师，有点糊涂，到底这个 非对称加密 是一个有公钥一个有私钥，还是都有公私钥？

作者回复: 取决于双向认证还是单向认证。

如果是单向认证，也就是目前大多数的用法，只发送服务器的公钥，验证服务器的身份。

如果是双向认证，那么服务器和客户端都要发送各自的公钥，互相验证对方的身份，一个常见的场景就是网银的U盾。



7



周曙光爱学习

2019-12-16

老师你好，看到回复中有同学说每次https请求都需要走一次完整流程，即先通过非对称加密获取对称加密的密钥，然后再用对称加密密钥解密数据，这种说法不准确吧？因为是长链接，只要这个链接不断，不管多少次请求，这个对称加密应该都是同一个吧？只有链接断开重新建立才需要走一遍完整流程？求老师解答

作者回复: 是的，你的理解是正确的。

有时候回答问题时难免有考虑不周，答案可能不是非常准确，你这种认真的态度值得肯定。

回到https上，因为https是建立在tls之上，所以一次tls握手成功后，只要不断开，连接一直是处于加密状态，所以可以在这个长连接上收发多次http报文。



5

领资料



永钱

2019-07-22

- 1.现在很流行密码锁，密钥就是你设置的密码，没有密码，开不了锁
- 2.私钥加密叫加密，公钥加密叫签名，防止抵赖

作者回复: 2不太正确，感觉是弄反了。



其实两者在密码学上都可以叫加密，互相加密解密。只是一般习惯上的说法是公钥加密私钥解密，私钥签名公钥验签。



5



爱学习不害怕

2020-06-19

老师好，有个问题想问一下。

文中提到：“网站秘密保管私钥，在网上任意分发公钥，你想要登录网站只要用公钥加密就行了，密文只能由私钥持有者才能解密。而黑客因为没有私钥，所以就无法破解密文。”

1.是不是我们普通的用户也会持有自己的私钥？将公钥发给需要通信的网站，防止从网站发回的隐私信息泄露。

2.如果普通用户也有，那么这样的私钥和公钥是怎么生成的呢？我猜想是，如果是某个专用的app比如外卖或者电商，可以在app运行时在本地就生成直接传输。请问老师是这样的吗。还有就是是像浏览器这样的客户端怎么生成呢？

作者回复：

1.当然了，公私钥任何人都可以持有，不只是服务器，只是现在服务器用的最多，给人以误解。

2.公钥私钥的生成方式有很多，比如银行常见的U盾，就是在硬件内部生成，Linux也可以用ssh-keygen这样的命令自己生成。

app或者浏览器生成公钥私钥也是可以的，因为非对称算法就是个算法，公钥私钥就是个数字，怎么生成都可以，没有什么特别神秘的地方。

但如果需要证书，那就必须有ca参与了，可以看后面的课。



3



missing~~

2019-09-18

老师好，对于混合加密这块不是很理解，实质是通过非对称加密传递一个会话级别的密钥，假如客户端A用公钥加密了一个对称加密的密钥传递给服务端B,B收到后通过私钥解出来这个对称加密密钥然后做对称解密。客户端A第二次请求又走同样的逻辑。不知道我这样理解对不对，如果对那么为什么还要再做一次对称解密感觉没有必要这样不是更影响效率吗？

作者回复：这是为了安全起见，如果长时间都使用一个对称密钥加解密就容易被破解，所以每次通信都要选择新的密钥，保证安全。

当然这有效率的问题，所以tls就出现了会话复用，在一定的有效期内可以直接重用上次的对称密钥，提高效率。



3

领资料





Keep-Moving

2019-07-22

然后用随机数产生对称算法使用的“会话密钥” (session key)

这个能详细说一下吗？

作者回复: 这个其实很简单，就是产生一个随机数，比如16字节，然后用公钥加密后安全传递给对方。



3



牛

2021-01-22

关于混合加密，会话密钥应该不是通过网络传输的，而是两端独立生成的：  
非对称加密进行前，两端已经交换了各自生成的一个随机数  
非对称加密传输了客户端生成的另一个随机数  
两端根据这三个随机数生成会话密钥。

作者回复: good



2



彻夜繁星

2020-10-22

老师好，我有个小白问题：做实验/24-2的时候发现，公钥、私钥、明文都不变的情况下，刷新网页重新执行加解密，密文却会不断变化。

为什么公钥、私钥、明文、算法都不变的情况下，每次执行加密，会得到不同的密文呢？

作者回复: 这个就是GCM等密码算法的优势了，增加随机性，防止重复攻击、选择明文攻击。

如果用ECB等算法，每次的密文就是一样的，容易被密码分析破解出密钥。

再补充一点，比较好理解的做法就是在运算的时候加一点随机的“盐” (salt) 。



2

领资料

