



下载APP



## 20 | 综合案例：如何解决约会难题？

2021-01-08 范学雷

实用密码学

[进入课程 >](#)**讲述：范学雷**

时长 12:37 大小 11.57M



你好，我是范学雷。

今天是我们正式篇的最后一讲了，我们这场密码学之旅就要暂时结束了。还记得我们第一次的讨论吗？那时候，我们讨论了牛郎织女约会会碰到的困难，详细地分析了每一个障碍。

当时，我们说密码学可以解决牛郎织女的约会难题。这一路走来，每多学习一点知识，我都会再提一次这个难题，不知道你对这个问题，是不是有了很多新的认识和想法？



现在，我们已经学习了对称密码系统的基本框架了，一切有终有始，我们是时候回头看看，能不能使用我们学到的知识，来帮助牛郎织女解决好他们的约会问题了。

## 回顾一下约会难题

我们先来看一看第一次讨论时，我们罗列出来的问题。牛郎和织女需要考虑的问题主要有五点：

问题一：怎么证明双方的身份？

问题二：怎么能使消息不泄露？

问题三：怎么防止内容被篡改？

问题四：怎么确保信件能收到？

问题五：怎么防止翻脸不认账？

然后，我们通过这五个问题，分析出来了信息安全的六个需求：

需求一：识别身份，确定牛郎就是牛郎；

需求二：认证身份，验证牛郎就是牛郎；

需求三：管理特权，授予织女看信权利；

需求四：信息保密，没有权限不能看信；

需求五：信息完整，保护内容不被篡改；

需求六：信息可用，保持信息获取能力。

我们还提到，一般地，对称密码技术可以通过加密、解密，解决“机密性”的问题；单向散列函数可以解决“完整性”问题；非对称密码技术可以解决授权和认证的问题。

我们通过对这三项基础技术的综合运用，就可以提高系统的“可用性”。虽然，非对称密码技术不在这个专栏的讨论范围，不过，这并不妨碍我们在对称密码技术里寻找替代方案。

下面，我们就分别来看看这些问题可以怎么解决。

## 怎么解决机密性问题？

既然，对称密码技术可以解决“机密性”的问题，那么，想要使用对称密码技术，首先就要有对称密钥，而且牛郎和织女都要持有相同的密钥。接下来的问题是，对称密钥怎么来的呢？

## 对称密钥怎么来的？

对称密钥的两个来源，一个是用户持有的秘密，另一个是计算机持有的秘密。

对于计算机持有的秘密来说，每一个计算机持有的秘密都是不同的。即使牛郎和织女都有计算机，独自使用各自的计算机持有的秘密也不能直接演化出两个相同的对称密钥。

所以，我们能想到的，就是从用户持有的秘密着手，看看有没有两个人都知道的秘密，然后各自演化出两个相同的对称密钥。

牛郎和织女有没有共同的秘密呢？这个我们当然无法知晓，但是，我们就假设他们有共同的秘密，比如两个孩子的生辰八字。一般来说，一个人的生辰八字是敏感信息，是需要保密的。

牛郎和织女有一个男孩一个女孩，我们假设男孩的生辰八字是“庚子辛巳乙亥丙子”（庚子年，辛巳月，乙亥日，丙子时），女孩的生辰八字是“戊戌庚申甲申甲子”（戊戌年，庚申月，甲申日，甲子时）。

这样，这个共同的秘密就可以用来推演对称密钥了。怎么推演这个对称密钥呢？

还记得我们说过的，怎么使用口令生成对称密钥吗？在这里，生辰八字就可以当作口令用，基于口令的密钥推导算法可以使用 PBKDF2。下一步面临的问题是，两个人共有的小秘密可能很多，织女怎么才能知道牛郎选择使用孩子的生辰八字这个两人共有的秘密？

还有，使用的是哪一个孩子的生辰八字？生辰八字又是怎么推演出密钥的？

其实，问题很好解决，牛郎在信件里告诉织女这些信息就行了。这些信息并不需要加密，使用人人都能看的明文就行，这就是所谓的算法公开。我们讲过，算法公开并不会影响数据的安全性。

好，现在，让我们把基本思路整理一下：

牛郎要给织女发送约会信息，“织女，七月初七晚七点，鹊桥相会。不见不散。牛郎字”。这部分是私密信息，需要保密。

牛郎要告诉织女怎么解密信息，这一部分是公开信息。这部分信息包括：

1. 使用两个孩子的生辰八字作为推导对称密钥的秘密；
2. 使用什么密钥推导算法来从共有的秘密推导对称密钥；
3. 使用什么对称密钥算法来加密、解密约会信息。

现在有了推演对称密钥的基本思路了，接下来，我们要考虑算法的细节问题了。那么，牛郎该选用哪些基本的密码学算法？

## 怎么选密码学算法？

这个问题，其实就是说，在算法的选择上，我们要考虑什么问题。

首先，我们要确定所需的安全强度。如果牛郎希望约会信息的秘密维持得越长越好，256 位的安全强度是首选。

然后，我们来确定所需的密码算法。牛郎要发送的信息，既包含私密信息，又包含公开信息。私密信息和公开信息，都不能被篡改。这种情况下，我们可以使用带关联数据的认证加密算法，也就是 AEAD 算法。AEAD 算法可以从现在流行的三种算法里面挑选一个。

结合安全强度的要求，我们可以选择 ChaCha20/Poly1305 算法。

然后，我们去查查 ChaCha20/Poly1305 算法需要的条件。这个算法需要四个输入数据：

1. 一个 256 位的对称密钥；
2. 一个 96 位的随机数；
3. 待加密数据明文，也就是私密信息；
4. 关联数据，也就是公开信息。

怎么推导出这个 256 位的对称密钥呢？也就是使用我们上面说到的基于口令的密钥推导算法。

这个推导算法，我们可以挑选 PBKDF2，接着，我们要再去查查 PBKDF2 算法需要的条件。这个算法需要五个输入数据：


1. 用户的口令；
2. HMAC 算法；
3. 盐值，类似于我们讨论过的初始化向量；
4. 迭代次数；
5. 导出密钥长度。

接下来，我们还要选择 PBKDF2 使用的 HMAC 算法。先看看安全强度，我们需要选择至少 256 位安全强度的算法。HMAC 算法的强度，通常是由对称密钥决定。所以，我们可以选用最流行的 HmacSHA256 算法，推导出 256 位的密钥。

PBKDF2 算法需要的盐值，我们随机选取一个数值就可以了，迭代次数不妨就选用一次。这样，我们就可以使用 PBKDF2 算法了。它的输入数据看起来像下面的样子：

1. 用户的口令：“庚子辛巳乙亥丙子戊戌庚申甲申甲子”；
2. HMAC 算法：“HmacSHA256”；
3. 盐值：“3B 07 A6 CB CF 98 48 F0 68 11 28 40 E7 6F 98 66”
4. 迭代次数: 1;
5. 导出密钥长度: 256 位。

有了这些输入数据，我们就可以根据 PBKDF2 算法推导出 256 位的对称密钥了。

 复制代码

```
1 37 af 4f bd dd 22 7c f3 bc 66 d4 c0 2c 3d e4 5a
2 e4 b0 da f4 58 0f 37 19 b2 31 93 63 fc 61 61 9d
```

有了对称密钥，我们再去看 ChaCha20/Poly1305 的运算。在 ChaCha20/Poly1305 算法里，96 位的随机数，还是随机生成一个就可以了。它的输入数据看起来像下面的样子：

1. 一个 256 位的对称密钥：使用上面 PBKDF2 算法推导的对称密钥；
2. 一个 96 位的随机数：“0A 00 00 00 00 00 00 4B 00 00 00 ED”；
3. 数据明文：“织女，七月初七晚七点，鹊桥相会。不见不散。牛郎字”；
4. 关联数据：未知。

除了关联数据外，我们其他的数据都已经就绪了。那么，关联数据该怎么定义呢？

## 怎么定义通信协议？

其实，这就需要自己定义关联数据，也就意味着我们要定义自己的通信协议。通信协议里，我们一定要定义清楚通信双方都要遵循的格式和规范。在这个案例里，关键就是要告诉织女这封信件的格式，以及怎么解密这封信件。

其中，需要包含我们上述提到的算法的选择，以及对应的参数。

大致地，我们可以这么写这封信：

这封信使用 ChaCha20/Poly1305 算法加密，随机数选取的是“0A 00 00 00 00 00 00 4B 00 00 00 ED”（十六进制表示），关联数据是本信件内容除去加密数据之外的所有数据，加密密钥是使用 PBKDF2 算法推导出 256 位的对称密钥。PBKDF2 算法的用户口令，使用的男孩和女孩的生辰八字，HMAC 算法使用的是 HmacSHA256 算法，盐值选取的是“3B 07 A6 CB CF 98 48 F0 68 11 28 40 E7 6F 98 66”（十六进制表示），迭代一次。此后为加密数据。6E 2E 35 9A 25 68 F9 80 41 BA 07 28 DD 0D 69 81 ..... 5A F9 0B BF 74 A3 5B E6 B4 0B 8E ED

有了这封信，ChaCha20/Poly1305 算法要使用的关联数据也就确定下来了。就是信件里，除了加密数据之外所有的数据。有了关联数据，我们就可以执行 ChaCha20/Poly1305 的运算了。

织女收到这封信后，就可以按照信件明文部分的描述，结合自己掌握的两个孩子的生辰八字，解密加密信息，获取私密的约会信息了。

很明显，私密的约会信息被隐藏了起来，即使有人截获了这封信，也不会知道私密信息的内容。这个协议很好地解决了第二个问题，也就是怎么能使消息不泄露的问题。

## 约会问题都解决了吗？

那么，其他的问题呢？我们依次来看一看其他的四个问题。

第一个问题，怎么证明双方的身份？

由于孩子的生辰八字只有孩子的父母知道，牛郎知道，只有织女才能读到这封信的约会信息；而织女也知道，只有牛郎才能这样加密约会信息。身份证明的问题，我们就解决了。

第三个问题，怎么防止内容被篡改？

由于这封信采用的是带关联数据的认证加密算法，无论是信的私密内容，还是公开内容，都没有办法被篡改而不被察觉。信息完整的问题，我们也解决了。

第四个问题，怎么确保信件能收到？

遗憾的是，这个问题并没有彻底地得到解决。由于发信人、收信人以及信件的真实内容都没有泄露，别人截留这封信件的动机可能就没有那么强烈，传递信件的喜鹊被请去吃酒喝茶的概率也会急剧减少。

即便喜鹊被请去了，也不能解密私密信息的内容，无论是利诱还是威逼，都没有用。这样，喜鹊的人身安全风险急剧降低。从这个方面说，这个方案提高了信件送达的概率。剩下的，就要依靠喜鹊飞越银河的能力了。

第五个问题，怎么防止翻脸不认账？

由于孩子的生辰八字只有孩子父母知道，织女收到信后，她就知道，只有牛郎才能这样加密约会信息。因为使用的是只有这两个人才知道的共同秘密，牛郎就不能抵赖说不是自己发送的信件。

可是，牛郎怎么能够确认，织女收到了信，还不能抵赖说没收到呢？我们把这个悬念留作今天的思考题。你可以好好地思考一下。



## Take Away (今日收获)

今天，我们回顾了一下牛郎织女约会这个案例，讨论了怎么使用我们前面学过的密码学知识来解决具体的问题。

为什么我们讨论的方案能解决牛郎织女约会的问题呢？最基本的原因主要有两点：

使用共有的秘密，这样就解决了身份认证的问题；

使用 AEAD 算法，这样就同时解决了数据完整性和数据机密性问题，同时提高了信息的可用性。

其实，更流行的方案是使用基于非对称密钥的密钥交换技术，或者类似于 Kerberos 的基于对称密钥的密钥交换技术，不过，这两个方案都不在我们的讨论范围里。

有兴趣的话，你可以找一下相关的资料看一看，想一想基本的思路。今天的讨论，主要是让你感受一下密码学解决问题的思路，尤其是怎么把多种算法结合起来解决具体的问题。

## 思考题

正如我们前面提到的，今天我们讨论的方案，只能解决牛郎不能翻脸不认账的问题，还没有解决织女翻脸不认账的问题。织女翻脸不认账的问题，该怎么解决呢？

也就是说，牛郎怎么能够确认，织女收到了信，还不能抵赖说没收到呢？我们上述的方案该怎么改进？或者你有没有其他的方案？

欢迎在留言区留言，记录、讨论你的想法。

好了，课程到这里，就基本结束了，下一次我们结束语见。

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 19 | 量子时代，你准备好了吗？

下一篇 结束语 | 深挖坑、广积粮

## 精选留言 (5)

写留言



John

2021-01-11

> 牛郎和织女有没有共同的秘密呢？这个我们当然无法知晓，  
> 但是，我们就假设他们有共同的秘密，比如两个孩子的生辰八字。  
> 一般来说，一个人的生辰八字是敏感信息，是需要保密的。  
能知道孩子生辰八字(出生时间)的人可能会很多，除了其父母，还可能有稳婆，邻居，亲戚，朋友，甚至是一直监视牛郎织女的王母娘娘的手下.....

展开 ∨



qinsi

2021-01-08

王母娘娘的对策：

- \* 把指向织女住所的路标改成指向别的地方
- \* 发现是飞往织女住所的喜鹊一律请去喝茶
- \* 发现有些地方在偷偷给织女转发二次打包的信件，于是规定收信的单位必须先报备，不...

展开 ∨

作者回复：+ 如果收信地址不详，一律截留。



qinsi

2021-01-08

PBKDF2只迭代一次是认真的吗...

展开 ∨

作者回复：哈哈，常用的迭代数目是1万。我选用1次的时候，还在想，会不会有人问1次安全强度够不够？1次的安全强度是不够的，但是要讲清楚迭代次数的影响，需要很多篇幅。要了解更深

入，自己去查查PBKDF2的规范吧。具体到这个例子，迭代次数还不是最大的安全问题。



**小动物**  
2021-01-08

通过回执来回确认来判断是否都收到了信息。但这会有鸡生蛋蛋生鸡的问题，我怎么知道对方知道我知道了。所以要多次确认。但具体几次确认才能完全确认有些绕迷糊了。若真的无限循环了，那就可以通过概率来确认，每次消息发送时附带时间戳，目的是让双方都知道每次消息传输的耗时。然后通过耗时来估计最后一条消息是否在预定时间前送达了。

展开

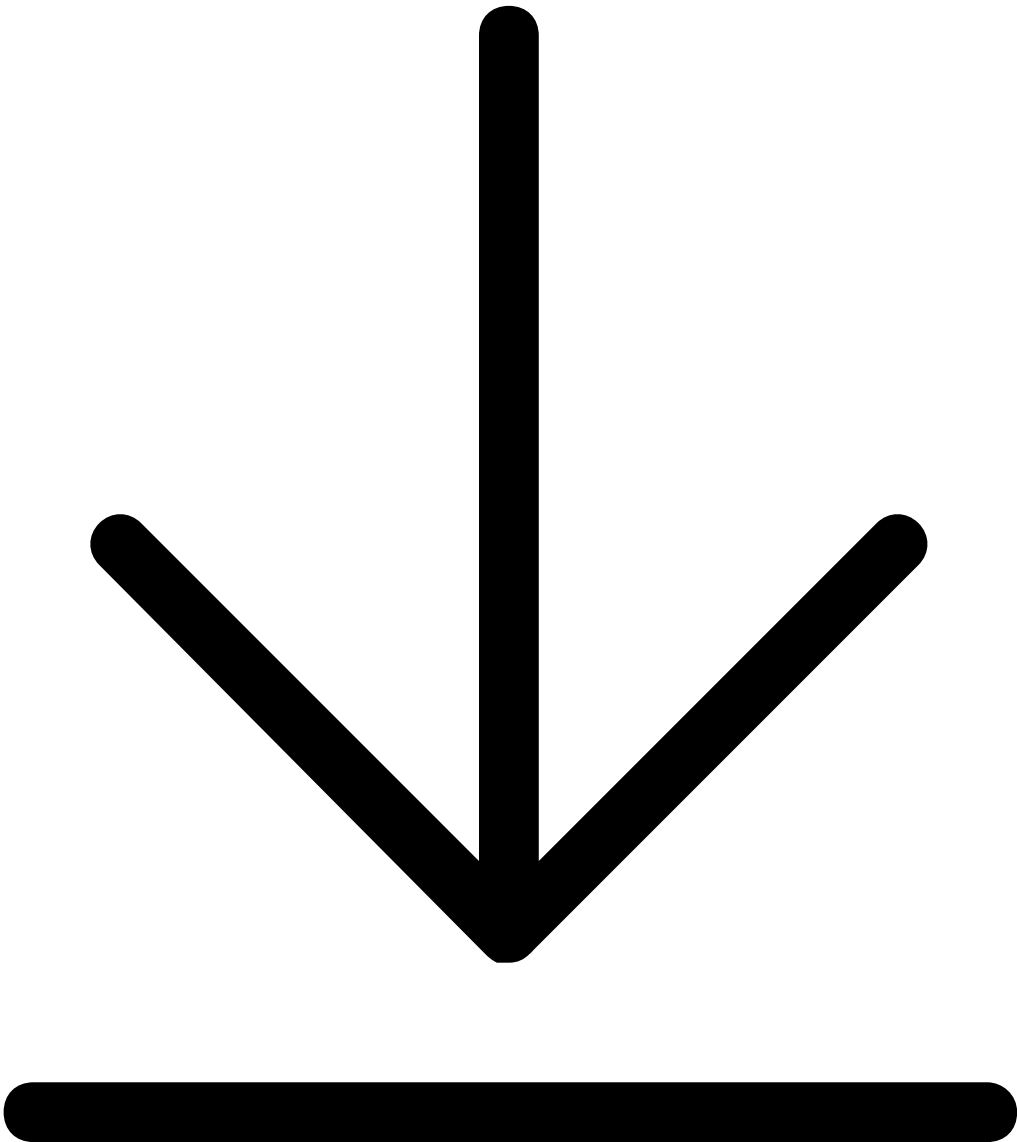
作者回复: 看看TCP协议怎么设计的？



**孜孜**  
2021-01-08

让喜鹊带一封回执信回去。但是如果织女收到信后把喜鹊关起来，那就没办法了。

作者回复: 哈哈，织女关喜鹊的动机是什么？



拖拽到此处  
图片将完成下载