19 | 量子时代，你准备好了吗？

2021-01-06 范学雷

实用密码学 进入课程 >

讲述：范学雷
时长 10:54 大小 9.99M

你好，我是范学雷。

上一讲，我们讨论了怎么管理对称密钥，强调了要优先使用即用即弃的对称密钥。我们还一起分析了对称密钥的一些问题，相信你也感受到了，其规模化背后的麻烦。

不过，我们又留了一个小尾巴，就是对称密钥能不能应对未来量子计算时代的计算能力？

也就是说，现在使用对称密钥加密的数据，在量子计算时代能不能被破解？我们又该怎么保护我们的敏感数据，不受量子计算时代的算力影响？这是我们这一次要讨论的话题。

量子算力有多强？

要讨论量子计算的影响，我们首先要有一个概念，那就是量子算力有多强。

2020 年 12 月 4 日，中国量子计算原型机“九章”问世。理论上，这个计算机比目前最快的计算机还要快一百万亿倍。也比 2019 年谷歌发布的量子计算原型机“悬铃木”快一百亿倍。

你看看，现在量子计算的性能提升，是百亿倍这个数量级的。

还记得我们总提到的安全强度吗？那时候，我们有一个粗略的判断，使用 10 亿台 1 纳秒计算一次的计算机，破解 128 位的安全强度，需要一千万个十亿年。还有同学留言估算，使用目前最快的计算机 Fugaku，破解 128 位的安全强度，需要 2500 个十亿年。

而“九章”比目前最快的计算机还要快一百万亿倍，按照这个计算能力，如果我们对照 Fugaku 计算机，破解 128 位的安全强度，只需要 2.5% 的年，也就是大约 9 天。

量子计算机的研发还在原型阶段，成熟的量子计算机，性能可能还会有大幅度的提高。到了量子计算机成熟的时候，破解 128 位的安全强度，可能只是微秒或者纳秒级别的计算。

所以，量子计算时代，128 位的安全强度也就不再安全了。

我们再回顾一下第 3 讲，欧洲的 ECRYPT-CSA 的密码安全强度建议。ECRYPT-CSA 建议，128 位的安全强度的密码学算法可用于 2028 年之前，2028 年之后就要使用 256 位的安全强度了。这个建议，就是考虑到了量子计算的影响。

2028 年，离现在已经不是很远了。

那 256 位的算法安全吗？

是不是说，256 位安全强度的密码学算法，在量子计算时代都是安全的？

想要得出答案，我们还可以使用“九章”量子计算原型机的数据，来估算一下破解 256 位安全强度的密码学算法需要多长时间。为了简化计算，现在我们假设量子计算机一纳秒就可以破解 128 位的安全强度，我们有 10 亿台这样的量子计算机。

借用我们前面对 128 位安全强度算法的估算数据，破解 256 位的安全强度，要一千万个十亿年。

所以，我们可以有这样一个印象，在量子计算时代，256 位的安全强度，大致相当于今天 128 位的安全强度。今天 128 位的安全强度，是足够安全的。**在量子计算时代，256 位的安全强度也是足够安全的。**

我相信，你应该已经有这样的意识了：攻击者不会只按照算法设计者设计的路线攻击，他们有的是千奇百怪、出乎意料的办法。那么，对于 256 位的安全强度，有没有密钥算法承受不了量子计算时代的算力的？这个答案是肯定的。

尤其让人遗憾的是，所有现在流行的非对称密码算法，都不能抵御量子计算时代的算力。

不过，这和现在的非对称密码算法的设计思路有关系，业界现在也在紧锣密鼓地设计、遴选量子计算时代非对称密码算法。让人欣慰的是，256 位安全强度的对称密钥算法和单向散列函数，包括 AES 算法，在量子计算时代还是安全的。

非对称密码算法不能抵御量子计算时代的算力，那么由非对称密码推导出来的对称密钥，是不是也不能抵御量子计算时代的算力？回答这个问题之前，我们先来了解一个概念：前向保密性。

什么是前向保密性？

什么是前向保密性呢？

在密码学里，前向保密性指的是即使用来协商或者保护数据加密密钥的长期秘密泄漏，也不会导致数据加密密钥的泄漏。换个角度看，虽然数据加密密钥是由长期的秘密衍生出来的或者保护的，但是数据加密密钥不能再一次通过长期秘密推导出来。

是不是感觉有点绕口，不知道该怎么理解？

我们来看一看前面讨论过的，从用户口令推导出数据加密对称密钥的算法。每次需要使用这个对称密钥的时候，都可以通过用户口令再次推导出来，而不需要把这个对称密钥保存到硬盘上。

这个过程就不是前向保密的。如果用户口令泄漏了，这个用来加密数据的对称密钥能够再次被推导出来，对应的加密数据也能够被破解。

数据加密密钥可以从长期秘密衍生出来，但是又不能再一次推导出来，这听起来有点别扭。

关键就在于使用用完就丢、而且用完就不能再找回的秘密，比如说随机数。如果随机数也参与数据加密密钥的衍生，只要这个衍生算法是安全的，那这个数据加密密钥也就具有了随机性。

只要生成这个数据加密密钥的随机数被干干净净地丢弃了，别人就不太可能重新找回这个随机数，也就不能再次推导出同样的数据加密密钥了。

那单纯地使用随机数生成的数据加密密钥，具不具备前向保密性呢？比如说，如果我们就使用随机数生成的对称密钥，然后保存到读写权限受到限制的保密文件里。每次加密或者解密需要使用对称密钥的时候，我们就把它从文件里读取出来。那么，这样的密钥能不能前向保密呢？

能反复长时间使用的密钥，就是人们常说的静态密钥，也是我们前面提到过的需要留存的密钥。上面说的，就是一个静态密钥的例子。**静态密钥不能前向保密**，因为，静态的密钥一旦泄漏，数据保密性也就无从谈起了。

我们把需要留存的密钥排除出去后，那具备前向保密性的密钥一定只能是即用即弃的密钥了。而且，这个密钥的衍生，一定要有即用即弃的随机数的参与。

这样，我们就找到了具备前向保密性的对称密钥的两个特点：

密钥的产生需要有即用即弃的随机数的参与；

密钥的本身是即用即弃的密钥，而不能是静态的密钥。

具备了这两个特点以后，即使协商或者保护数据加密密钥的长期秘密泄漏，也不会导致数据加密密钥的泄漏。这就大大降低了长期秘密泄漏带来的数据泄密的风险。

这就是前向保密性要解决的问题。好了，我们讨论了前向保密性。接下来的问题是，前向保密性和抵御量子计算时代的算力，有什么关系呢？

怎么使用前向保密性？

要想现在就开始筹划抵御量子计算时代的算力的事情，有两件事是一定要考虑的。

第一件事，就是使用量子时代依旧安全的密码学算法。我们要使用 256 位安全强度的对称密钥算法和单向散列函数这样的密码学算法，来确保量子计算时代的算力无法蛮力地破解加密数据。

第二件事，就是使用具有前向保密性的对称密钥来保护数据，特别是密文能够泄漏的数据，来确保量子计算时代的算力无法获得加密使用的密钥。

我们提到，现在流行的非对称密码算法，不能抵御量子计算时代的算力。那基于非对称密码的密钥交换或者密钥协商算法，当然也不能抵御量子计算时代的算力。那么，是不是基于非对称密码的密钥交换或者密钥协商算法衍生出来的对称密钥，也不能抵御量子计算时代的算力呢？

这要看对称密钥衍生算法的细节。如果对称密钥的衍生算法里，有即用即弃的随机数的参与，或者用来衍生对称密钥的非对称密钥也是即用即弃的，而且这个对称密钥还是即用即弃的密钥，那使用它加密的数据，也能够抵御量子计算时代的算力。

具体的算法细节，依赖于我们对非对称密码算法的理解，我们以后有机会再聊。

到目前为止，这个专栏就接近尾声了。我们从专栏之初就开始操心的牛郎织女约会问题，下一讲来看一看，我们能不能通过在这个专栏里讨论的算法，来解决这个约会难题。

Take Away（今日收获）

今天，我们讨论了量子计算时代的算力，相信你对量子计算时代的算力有了一个大致的印象。然后，我们还提到了，256 位安全强度的对称密钥算法和单向散列函数，包括 AES 算法，在量子计算时代还是安全的；而非对称密码算法，不能抵御量子计算时代的算力。

量子计算时代的非对称密码算法目前还在遴选中，而主流的密钥交换算法一般都是建立在非对称密码技术之上的。在后量子时代的非对称密码算法普及之前，我们有没有办法抵御量子计算时代的算力呢？**这个问题的答案，就是要使用具备前向保密性的对称密钥。**

当然，前向保密性不仅仅是用来应对量子计算时代的算力的。它也是我们现在协议设计和实现里，实现深度防御的一个重要考虑因素。不具备前向保密性的对称密钥，不能使用在加密数据有可能外泄的场景里，这几乎会断送加密的意义。

所以，对于现在的安全协议来说，要求具备前向保密性是一个硬指标。

通过今天的讨论，我们要：

了解量子计算时代的算力；

知道 256 位安全强度的对称密钥算法和单向散列函数在量子计算时代还是安全的；

知道现有的非对称密码算法，不能抵御量子计算时代的算力；

了解前向保密性要解决的问题和它的特点。

思考题

今天的思考题，是一个动手题。

在你正在开发的项目中，或者你关注的开放源代码项目中，试着搜索一下其中的对称密钥是如何产生的、有没有留存、具不具备前向保密性？对称密钥的强度能不能应对量子计算时代的算力？你有什么改进的建议？

这是一个能够帮助你提前评估量子计算影响的好办法，也能够帮助你为量子时代的计算能力做好准备。欢迎在留言区留言，记录、讨论你的发现和建议。

好的，今天就这样，我们下次再聊。

上一篇 18 | 如何管理对称密钥?

下一篇 20 | 综合案例：如何解决约会难题?

精选留言 (2)

写留言

王啸
2021-01-08

有没有java的例子来说明一下这个对称加密的秘钥生成和加密的数据生成?

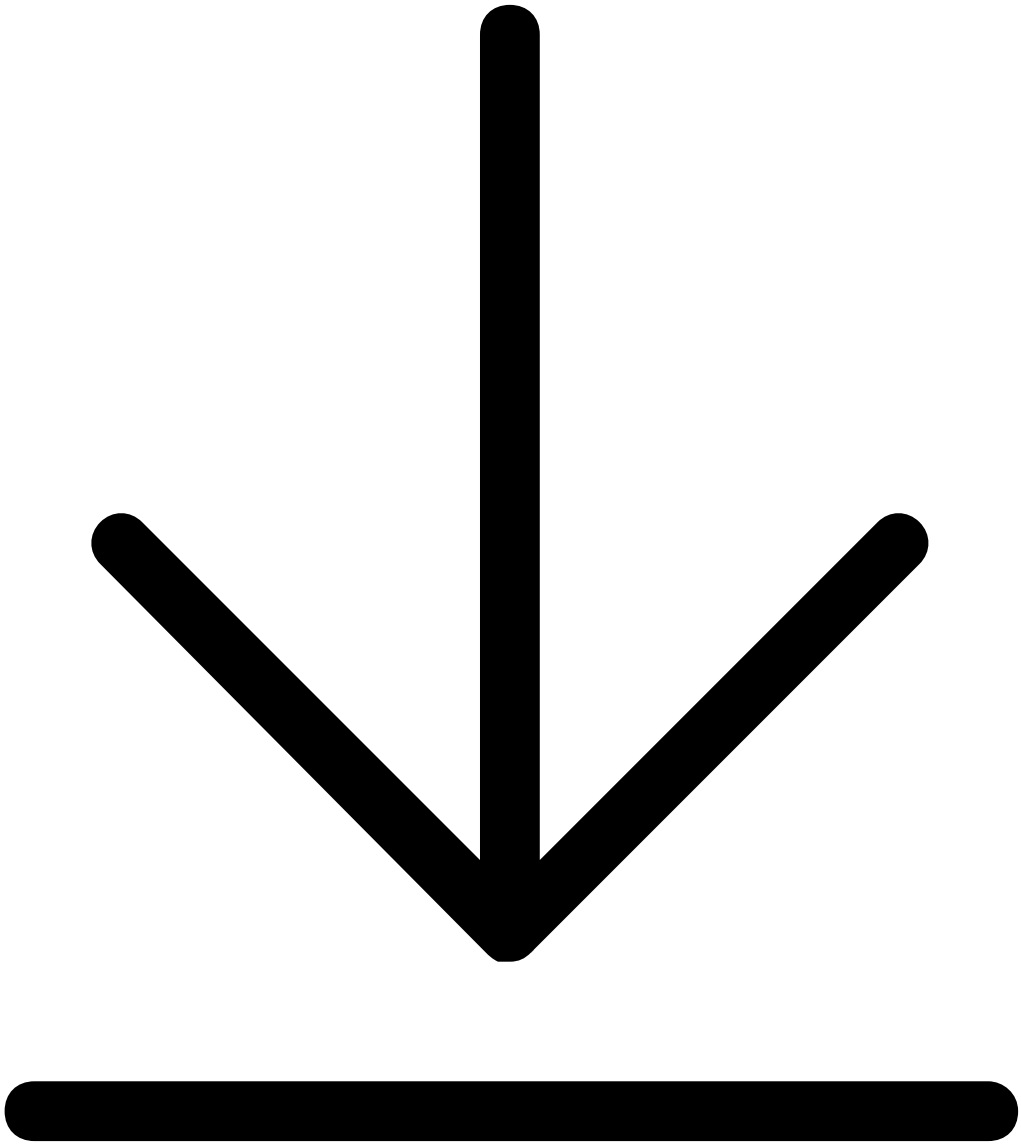
作者回复: 搜索一下，应该可以找到很多代码。


彩色的沙漠
2021-01-06

在TLS协议里面RSA秘钥交换不是向前保密性的，DH秘钥交换是向前保密性的（每次客户端和服务器端都会生成一堆公私钥，交换公钥之后，就可以能推到除会话秘钥，关键是即用即弃的特点），所以其TLS1.3协议里面已经移除了RSA秘钥交换算法

展开

作者回复: 赞！ DH秘钥交换->DHE秘钥交换

拖拽到此处
图片将完成下载