18 | 如何管理对称密钥？

2021-01-04 范学雷

实用密码学 进入课程 >

讲述：范学雷
时长 10:06 大小 9.25M

你好，我是范学雷。

上一讲，我们讨论了合格的对称密钥需要具备什么条件，以及对称密钥是怎么产生的。但是，了解对称密钥是怎么产生的，是远远不够的，我们还要了解怎么管理这些对称密钥。

在我们讨论怎么管理之前，我们还要再给对称密钥分个类，划分的标准就是对称密钥要不要留存。那么，哪种密钥不需要留存呢？我们该怎么管理呢？这是我们这一次要讨论的问题。

对称密钥要不要留存？

你是不是有些迷惑，难道对称密钥不需要留存？如果不留存，为什么还要做密钥管理？

为什么我们要讨论对称密钥要不要留存？因为最好的管理，就是不管理或者少管理。按照这个思路，我们可以把对称密钥分成两类：即用即弃的对称密钥和需要留存的对称密钥。

即用即弃的对称密钥

即用即弃的对称密钥，就是用的时候才生成出来，不需要保存，用完了就扔掉的对称密钥。

那这种对称密钥的适用场景是什么呢？**即用即弃的对称密钥可以用在加密数据不需要保存的场景**。比如说， 像 HTTPS 这样的端到端传输协议，它的网络传输数据是加密的，而加密的网络传输数据是不需要保存到硬盘里的。

这种情况下，对称密钥可以使用计算机持有的秘密，也就是使用随机数来生成。还记得什么是计算机持有的秘密吗？你可以复习一下 17 讲，来回顾一下这个知识点。

当然，有的时候，加密数据需要保存的场景，也可能要使用即用即弃的对称密钥。但是，对称密钥不是被扔掉了吗？没有对称密钥，怎么解密加密数据呢？我知道你可能会有这样的疑问。

其实，思路很简单，我们在解密的时候，再生成一个完全相同的对称密钥就行了。这种情况下，对称密钥可以使用用户持有的、不需要存储的秘密，比如口令，在需要对称密钥的时候，即时地把它推导出来。

即用即弃的对称密钥，是我们推荐使用的对称密钥。下一节，我们再来聊推荐原因。

需要留存的对称密钥

和即用即弃的对称密钥相对的，就是用完了不能丢弃的对称密钥，也就是需要留存的对称密钥。

需要留存的对称密钥，大部分出现在用户无法参与的计算环境里，比如自动启动的服务器。因为，在用户能够参与的计算环境里，不应该使用需要留存的对称密钥。取而代之的，应该是由用户持有的、不需要存储的秘密推导出来的即用即弃的对称密钥。

要知道，对称密钥需要保密。毫无疑问地，留存的对称密钥需要得到额外的照顾，避免对称密钥的泄漏。比如说，保存对称密钥的文件，它的权限需要设置成只有它的拥有者才能阅读、修改。在高度保密的环境下，甚至，我们需要把对称密钥保存到专用的芯片里。

需要注意的是，在我们设计的软件架构里，应该尽量避免使用需要留存的对称密钥。无论对称密钥是保留在专用的芯片里，还是保密的文件里，随着时间的推移，留存的密钥都有泄漏的风险。

另外，我们前面已经讨论过，既然对称密钥需要保密，我们就要把对称密钥当做是超级敏感的信息来处理。这些处理方式，包括但不限于，要及时清理内存里的对称密钥，而不能依赖系统的内存回收机制；不要把对称密钥有意无意地泄漏出去，比如把对称密钥写到系统日志里。

对称密钥有什么麻烦？

接下来，我们来看看对称密钥会有什么麻烦？这会利于我们分析对称密钥的管理问题。

我们已经知道，所谓的对称密钥，就是加密和解密都使用相同的密钥。如果加密和解密都是同一个参与者，自己加密的数据自己解密，那么只要持有一份对称密钥就行了。

可是，如果牵涉到两个或者两个以上的参与者，那么，每一个解密的参与者就都要持有和加密的参与者相同的对称密钥，解密才能成功。这就带来了很多麻烦。

让我们来看一个例子，假设一个系统有三个参与者 A、B 和 C。

如果每两人之间的通信都使用不同的密钥，那么 A 和 B、B 和 C、A 和 B 之间，都需要不同的对称密钥。也就是说，三个参与者的系统，需要 3 对不同的对称密钥。类似地，我们可以计算出：

5 个参与者的系统，需要 10 对不同的对称密钥；

10 个参与者的系统，需要 45 对不同的对称密钥；

100 个参与者的系统，需要 4950 对不同的对称密钥。

随着参与者的增加，需要的对称密钥数量急剧地膨胀。这种膨胀的速度，就给密钥的管理带来了很多麻烦，也会使得系统的效率急剧地下降。显然，大量的这种一对一的密钥的设计，不适合有众多参与者的应用。

但是，如果无论参与者有多少，都使用一个相同的对称密钥呢？如果这些参与者之间，是可以信任的，这样做的问题不大。比如说，一个公司内部的远程视频会议，就可以使用同一个对称密钥加密视频数据，然后把加密后的视频数据分发给每一个参会者。

这样，每一份视频数据，就只需要一份加密，每一个参会的都能够解密，看到会议的内容。那要是每一个参会者之间都使用不同的对称密钥呢？那也就意味着，视频的发送端，需要给每一个参会者都发送不同的加密数据。

一个有 100 个人参与的视频会议，每一份视频，都需要有 99 份的加密。你可以想象一下，和只使用一份加密数据的方案相比，这么大的计算量会给这个视频会议系统带来多大的性能麻烦。

而且，也不是每一个场景里，它的参与者都是可以信任的。比如说，我们可以想一想即时通信系统里记录的联系人。这些人，有时候也被叫做朋友圈。可是，虽然叫做朋友圈，圈子里的不一定都是熟人，更不一定都是见过面的人。

当然，朋友圈有我们可以信任的朋友，也有我们不能信任的陌生人。如果我们和不同的联系人通信，都使用相同的对称密钥加密通信数据，那我们不信任的陌生人，就也知道了这个对称密钥。

他们就都能解密我们和每一个联系人的通信数据。即使我们可以信任的朋友，也不意味着他们之间就不能有两两之间的秘密。**所以，每个人之间都使用相同的对称密钥是不行的。**

如果每两个人之间的通信数据，都使用不同的、只有这两人才知道对称密钥加密，也就是所谓的端到端的加密技术了。想一想，如果有 2000 个联系人使用端到端的加密，也就意味着需要 2000 个对称密钥。

这是不是意味着，需要保存 2000 个对称密钥呢？通过前面的讨论，我想你已经有了答案。

在即时通信的场景里，还有一种用户不会喜欢，但是厂商会喜欢的加密方式。那就是每一个用户都把数据加密传递到通信的服务器，然后再由服务器分发给数据的接收方。

这种方式最大的优点，就是服务器知道用户发送的数据明文。服务器知道了数据的明文，就可以做很多事情了。有些事情，我们也许会喜欢；有些事情，我们可能不会喜欢。这种方式还有一个不太重要的优点，就是每一个用户只需要一个用于和服务器通信的对称密钥就够了。这无疑降低了系统设计的复杂度。

但是，我们不会喜欢服务器窥视我们的隐私，因为机器的背后站着不可预测的人。我们也不希望保存 2000 个对称密钥，毕竟密钥的管理不是一件轻松的事情。有没有办法，我们可以和成千上万的人通信，每一个通信都使用不同的对称密钥，但是又不需要在本地保存这些对称密钥呢？

当然是有答案的。Kerberos 就是一个仅仅使用对称密钥系统，就可以支撑这种通信方式的协议。更直观的方法，就是使用基于非对称密钥的密钥交换技术。

Kerberos 协议的使用场景，目前还在逐渐萎缩。除了单点登陆的系统之外，至少在我的认知范围内，使用 Kerberos 协议的新系统已经不太常见了。不过，Kerberos 协议是一个设计优雅的协议。在不使用非对称密钥技术的情况下，它依然可以做到支持大规模用户的端到端加密，这是一个很了不起的成就。

对称密钥的规模化是使用对称密钥的一个大麻烦，这也给对称密钥的管理带来了很多挑战，不过也催生了很多成熟的解决方案。下一次，我们讨论对称密钥的另外一个麻烦，尤其是量子计算时代来临的时候，这个麻烦可能更要命。

Take Away（今日收获）

今天，我们讨论了生成对称密钥的时机，介绍了两种不同生存周期的对称密钥，也就是，即用即弃的对称密钥和需要留存的对称密钥。即用即弃的对称密钥是我们推荐使用的方式。如果对称密钥需要留存，一定要把它当做超级敏感的信息来处理。

另外，我们还讨论了对称密钥在规模化通信场景下的麻烦。使用场景不同，解决这些麻烦的办法也是不同的。更通用的解决方案，需要了解更高级的协议，或者非对称密钥系统。如果还有机会，我们以后再来讨论这些解决方案。

通过今天的讨论，我们要：

有意识优先使用即用即弃的对称密钥；

有意识去保护好需要留存的对称密钥；

知道对称密钥在规模化通信场景下的麻烦，能够有意识地去寻找、学习相应的解决方案。

思考题

好的，又到了留思考题的时间了。

这一次的思考题，我们再加大一点难度，留一个延伸阅读题。我们前面说过，Kerberos 协议是一个设计优雅的协议，能够用来解决对称密钥在规模化通信场景下的问题。

今天的思考题，就是去阅读 Kerberos 协议，去了解这个协议是怎么工作的，是怎么解决对称密钥的规模化通信问题的。如果让你使用 Kerberos 协议去设计一个即时通信软件的数据加密框架，你觉得会有哪些优点，会有哪些缺点？

欢迎在留言区留言，分享你的阅读体验和见解。

好的，今天就这样，我们下次再聊。

提建议

上一篇 17 | 加密密钥是怎么来的？

下一篇 19 | 量子时代，你准备好了吗？

精选留言 (1)

写留言

Ender0224

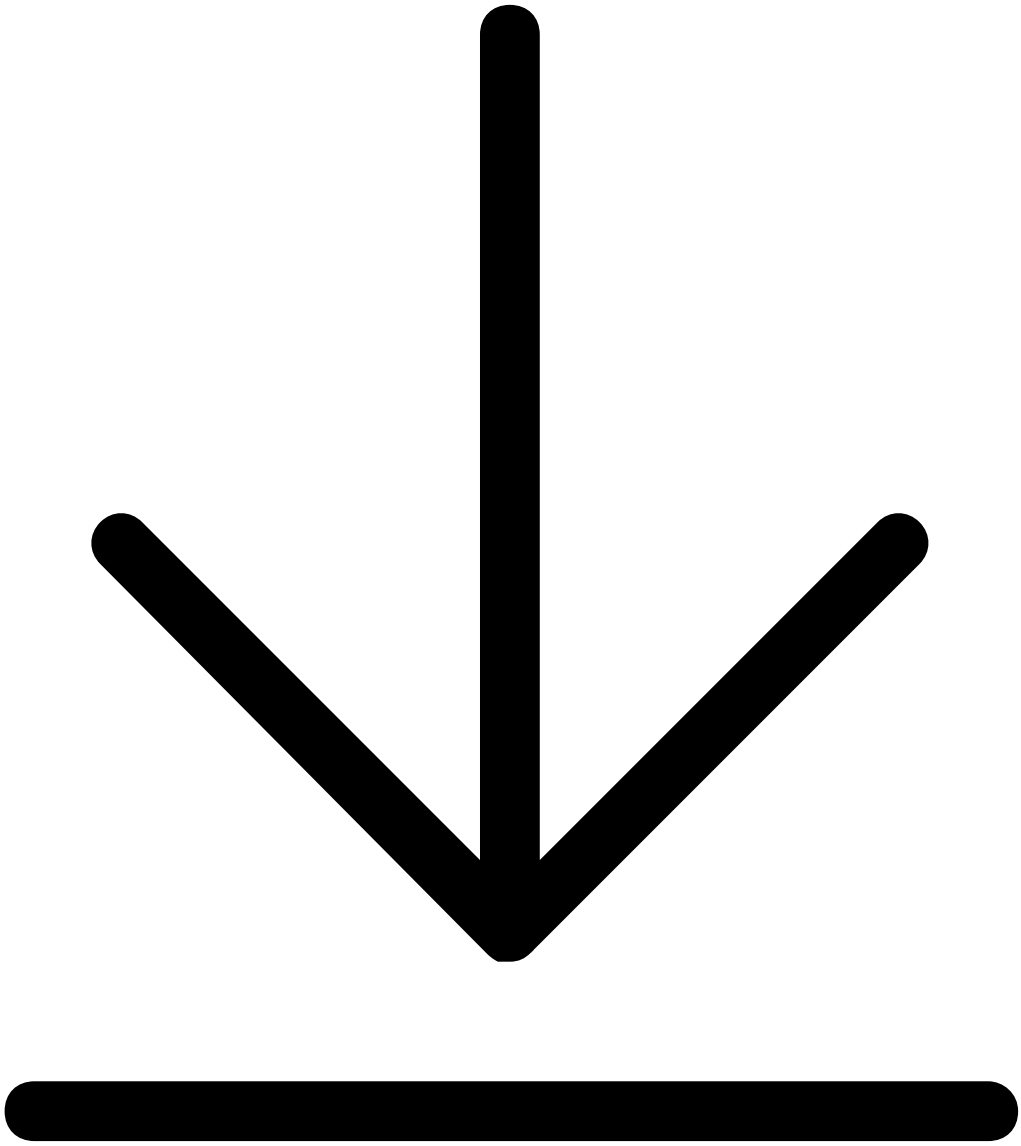
2021-01-05

请教，一般kerberos 的使用场景不是集中认证吗？和通信加密是不是没有关系。

我认为，kerberos解决了在不可信集群内提供了各组件的互相认证（仿冒）问题，但是现在云化下，厂商有更靠谱简单的方案构建一个可信环境，攻击者都攻破信任域到内部了，能干的就太多了，kerberos也防不住啥破坏了。加上引入了管理复杂性和降低性能，用...

展开

作者回复: 换种说法，kerberos解决的问题是怎么通过集中认证的手段，解决对称密钥规模化分发的问题。集中认证只是手段，解决的问题是怎么在两个实体之间协商出对称密钥。



拖拽到此处
图片将完成下载