41 真正理解 HTTPS

更新时间：2020-08-10 14:47:44

> 我们活着不能与草木同腐，不能醉生梦死，枉度人生，要有所作为。——方志敏

在前端面试中，总有一些话题，候选人说上半天也未必能说清楚、最后反而容易把自己绕到沟里去，HTTPS 就是这类问题的典型代表。

一方面，它具备一定的理论性，乍一看比较枯燥，这导致很多人在学习 HTTP 协议的时候都选择性地忽略了还有 HTTPS 这个东西；另一方面，它又确实挺绕的，就算你今天记住了，明天还是可能会忘记。由此，HTTPS 相关的考题具备了令面试官欣喜若狂的区分度——你越怕，他就越考。

一般来说，我们如果能对一样东西形成牢固的记忆，无非是两个原因：1.大量重复 2.印象深刻。

这就好比，楼下的看门大叔，他虽然长着一张让人记不住的大众脸，但你每天从他身边经过、每天要被他量一次体温，这样持续上几个星期，你想不记住他长啥样也难——此之谓“大量重复”；而你们公司神似高圆圆的产品经理韩梅梅，美得惊心动魄，见过她的人想必一辈子也忘不掉自己人生中出现过这样一个仙女——这就是“印象深刻”。

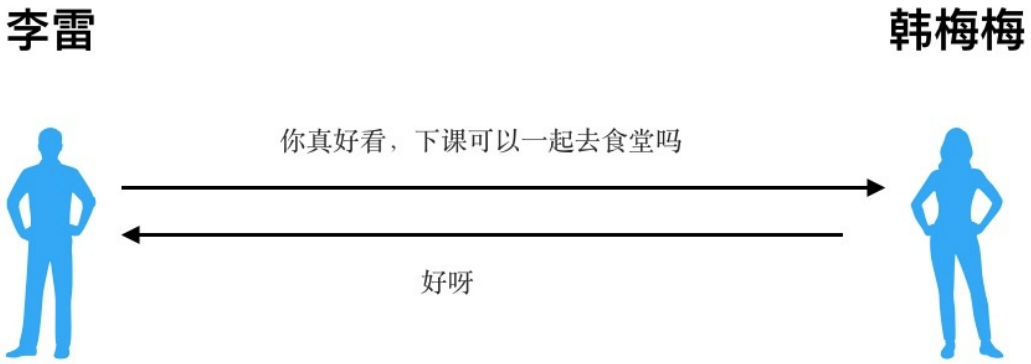
面向面试来说，各位当中恐怕极少有同学具备进行“大量重复”的时间资本。本着“效率为王”的原则，本节我们追求的就是一个“印象深刻”。因此咱们不直接罗列 HTTPS 相关的术语和理论知识，而是先来看一个贴近生活的小故事，从故事里去认知 HTTPS 中关键的知识点。

李雷和韩梅梅的故事

顺利的第一次约会

18岁的蠢萌少年李雷，暗恋着班上神似高圆圆的学霸韩梅梅。这一天，他终于鼓起勇气写下了“你真好看，下课可以一起去食堂吗”的邀请邮件，发送给了韩梅梅。

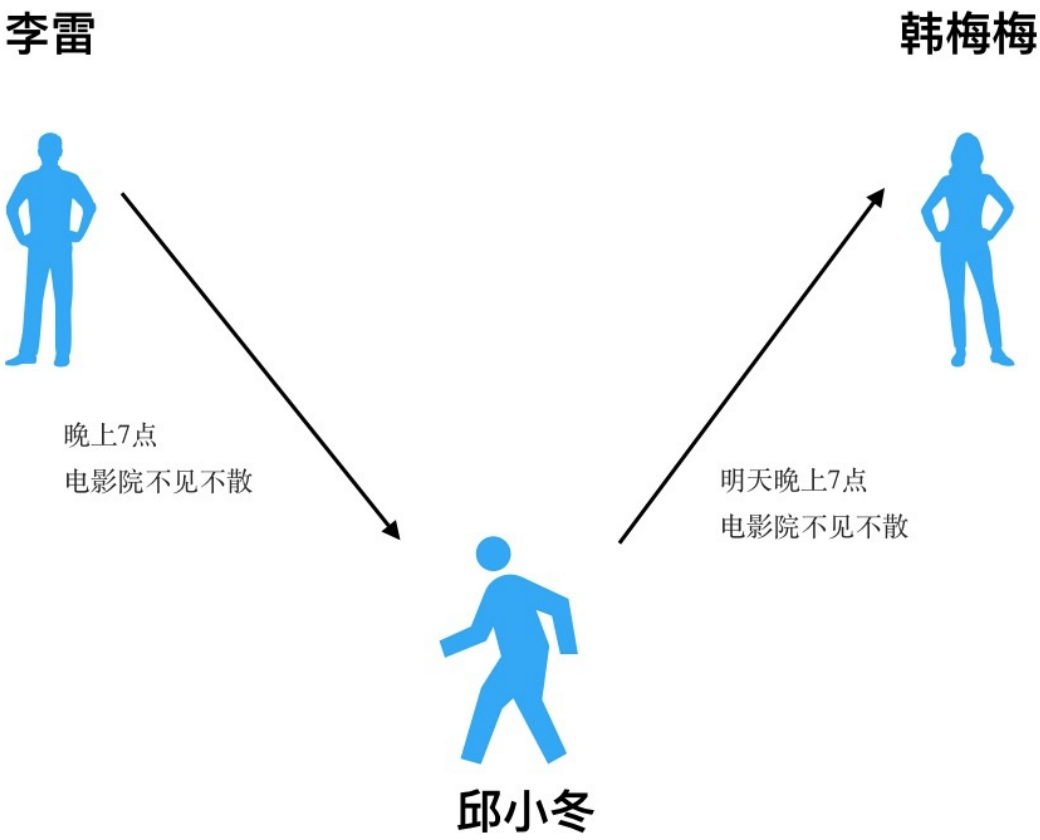
韩梅梅收到邮件一看，开心得不行，赶紧回复“好呀”。这是两人的第一次线上沟通，过程很顺利：



被邱小冬破坏的第二次约会

有了这一次对接成功的经历。第二天，李雷故技重施，又想约韩梅梅出去玩。于是他又写了一封邮件，满怀期待点了发送。没想到，班上喜欢韩梅梅的不止他一个人，还有住在他隔壁宿舍的邱小冬。邱小冬昨天眼睁睁看着李雷和韩梅梅开开心心一起去了食堂，心里很不爽。今天他在李雷宿舍的路由器上动了手脚，拦截李雷的网络请求。

邱小冬打开拦截到的邮件一看，上面写着“晚上7点电影院不见不散”。他眼珠一转就是一个馊主意，把“晚上7点”前面加了俩字，改成了“明天晚上7点”。然后，他把这封错误的邮件转发给了韩梅梅。于是这个沟通过程变成了下面这样：



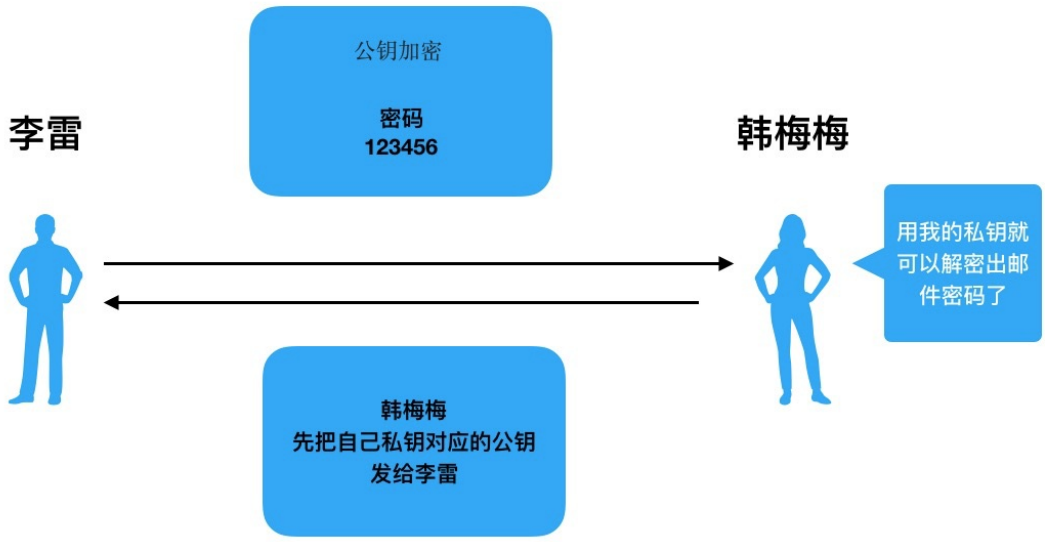
用密码来保护的第三次约会

结果这天晚上李雷等到电影散场也没等到韩梅梅来，第二天见了韩梅梅，才知道事情原委。两人决定用一种新的方式来沟通——对邮件内容进行加密：由韩梅梅把密码告知李雷，随后两人每次发送邮件都用这个密码来做加密和解密。假设密码内容是123456，那么沟通过程就变成了下面这样：

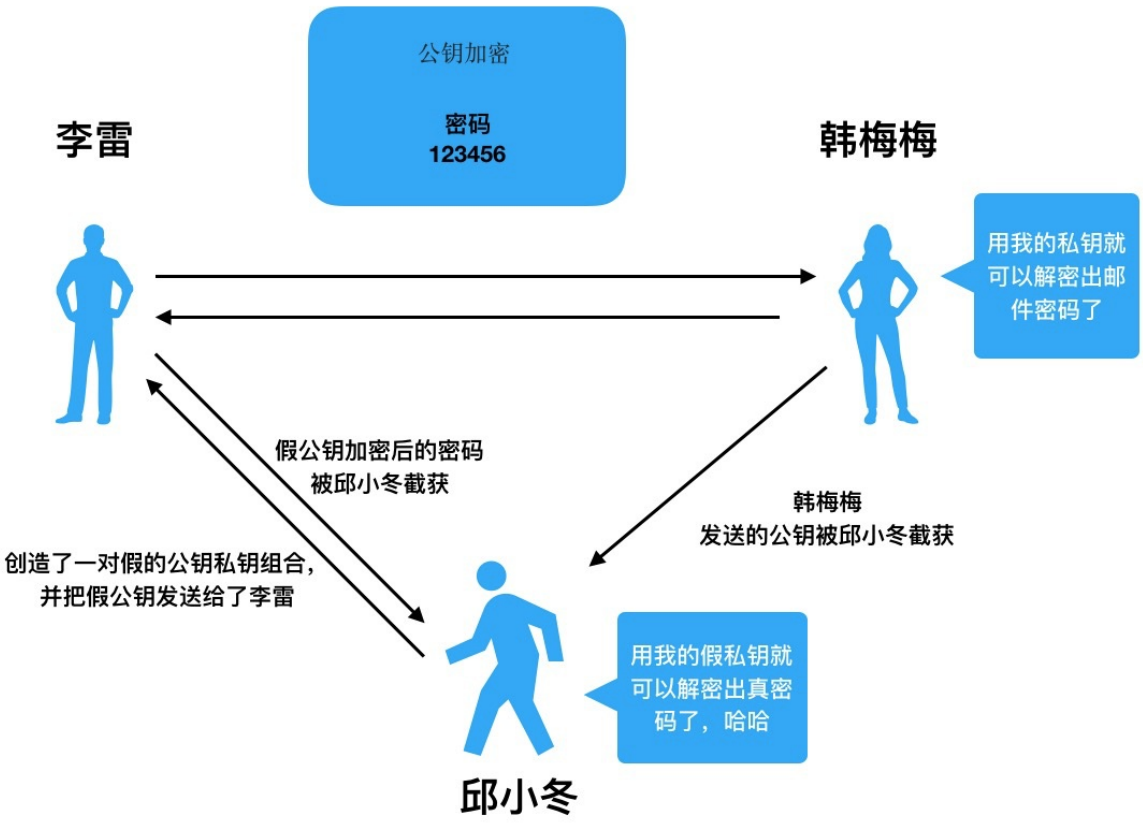
但是仔细想想，邱小冬既然能拦截邮件，自然就能拦截邮件密码。这还是不够安全。李雷琢磨了一下，觉得当下主要需要解决的问题是要保证邮件的密码安全。如何保证呢？李雷提出：我们来给邮件密码做加密！

邱小冬再也没辙的第四次约会

李雷想到，计算机课上，老师讲过一种组合加密的方式：公钥+私钥组合。公钥和私钥是多对一的关系。公钥加密过的信息，只有私钥能解开；私钥加密过的信息，所有和它有关系的公钥都能解开。完全可以让韩梅梅持有一个私钥，然后我用公钥把信息加密，这样被加密的信息就只有持有私钥的韩梅梅能解开了——就算邱小冬能拦截我的公钥，他手里没有私钥，到头来还是解不开。用这种方式，我们就可以保证邮件密码的安全了。这个过程如下：



看似没有问题，不过李雷可低估了邱小冬的阴险程度。邱小冬手里虽然没有真正的私钥，但是他的截获能力还在。现在他截获了韩梅梅发来的公钥之后，把自己编造的一个假的公钥发给了李雷。李雷傻乎乎地用邱小冬的假公钥给邮件密码加密后，邱小冬就能用自己的假私钥去解密这个邮件密码了。到头来，邱小冬还是拿到了邮件密码，可以对两人的邮件往来为所欲为：

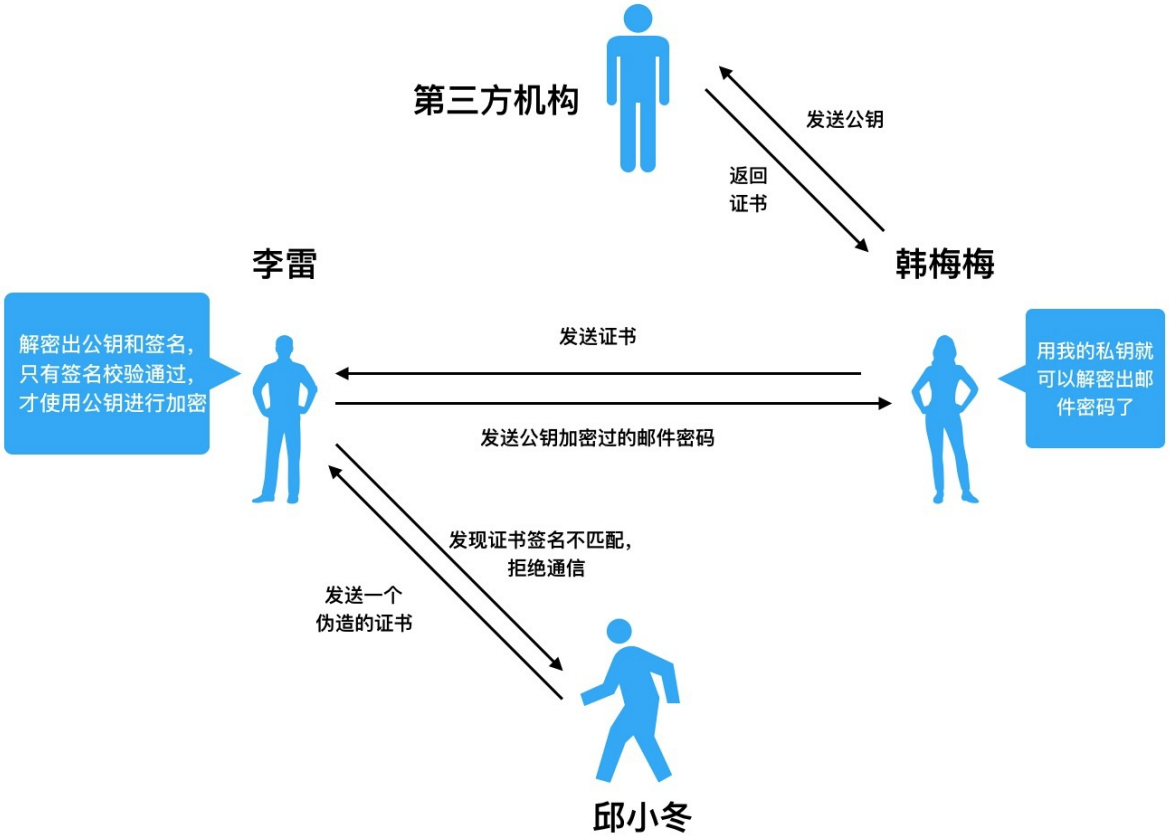


李雷彻底被邱小冬折磨得没脾气了。这时学霸韩梅梅说话了：“邱小冬之所以能继续得手，是因为你感知不到他的身份。你要早知道那是邱小冬发的公钥，而不是我发的公钥，那你就不会上他的当了。要我说，咱们就差一个身份认证机制了。而且这个身份认证必须得由权威的人来做，这样才能完全地规避掉冒充的行为。”

说话间，韩梅梅已经找来了一个可靠的第三方机构。在通信开始之前，韩梅梅首先会先把邮件密码发送给这个第三方。这个第三方会帮韩梅梅申请一个证书，然后用第三方自己的私钥来加密韩梅梅的公钥、再基于韩梅梅的服务端地址这些特殊的标识信息来为韩梅梅生成一个独特的签名，这些信息（包括第三方机构的名称、韩梅梅服务器的地址等信息）都会被写入到证书里、返回给韩梅梅。

接下来，韩梅梅不再直接发送公钥，而是把这个证书发送给李雷。李雷这边呢，会提前维护好所有正规第三方机构的公钥。李雷首先会根据证书里第三方机构的名称，定位到正确的公钥，然后用这个公钥去解密出韩梅梅发来的公钥、同时解密韩梅梅发来的那个独特签名。

注意，重点来了：**李雷再也不会轻信任何人发来的公钥了，他现在还要验证对方的身份**。如何验证呢？基于当前证书的签名生成算法，结合韩梅梅的服务器地址等信息，重新生成一遍证书签名，然后比对这个证书签名和韩梅梅发来的证书里的独特签名是不是同一个。如果是，那么万事大吉，李雷可以相信当前解密出的公钥就是韩梅梅发的公钥了。否则，那就很有可能是邱小冬伪造的证书，李雷直接拒绝通信即可。



题眼点拨

李雷和韩梅梅的故事中，“李雷”替换成“客户端”，“韩梅梅”替换成“服务端”，然后再把这个故事讲给面试官，其实就可以作为“谈谈你对 HTTPS 的理解”这个问题的完整答案了。

不过在这个故事中，有几个过程是需要大家重点理解和记忆的，在面试中面试官可能会着重考察这些过程对应的计算机定义。这里我用计算机的语言来帮大家对号入座一下：

明文传输

我们可以看到在第一次约会中，李雷和韩梅梅直接互通邮件，没有采取任何的加密措施。这就是所谓的“明文传输”。

对应到网络传输中，就是客户端发送请求，服务端发送响应，双方都不对自己的请求/响应内容做加密的情况。这种情况下，请求/响应一旦被中间人（比如邱小冬）拦截，就可以对其中的内容一览无余、为所欲为。

对称加密

第二次约会，李雷和韩梅梅约定了邮件密码。加密用这个密码，解密还是用这个密码。这就是所谓的“对称加密”。

对应到网络传输中，就是客户端和服务器约定一个共同的“公钥”，加密和解密都依赖这一个公钥这种情况。这种情况下，一旦公钥失窃了，那么双方传输的密文信息就会再次进入裸奔状态，仍然无法规避中间人的攻击。

非对称加密

第三次约会，韩梅梅保留了一个私钥，并且把和这个私钥关联的公钥发送给了李雷。李雷用这个公钥对邮件密码做加密，韩梅梅收到邮件密码后再用私钥来解密，这个过程就是“非对称加密”。

非对称加密，在这里指的是公钥+私钥配合加密这种手段。公钥和私钥是多对一的关系，公钥加密的内容，只有私钥可以解开，私钥加密的内容，所有的公钥都可以解开。

这样一来，就算中间人截获了公钥，但由于手里没有私钥，仍然没法正确地对数据进行解密。

注意，第三次约会时，两人只有第一次通信传输邮件密码时使用了非对称加密，后续仍然是通过一个邮件密码来进行加密&解密的。这个过程，就是对称加密和非对称加密结合的一种用法——先用非对称加密确保公钥（邮件密码）的安全，再用安全的公钥（邮件密码）来进行对称加密。

不过即便如此，仍然没办法规避中间人伪造公钥的这种场景，所以我们还需要第三方认证。

第三方认证

第四次约会，韩梅梅去找了第三方机构。这个第三方机构在现实中其实是真实存在的，叫做CA（Catificate Authority）。它的作用就是提供证书，证书中包含的主要信息有：

- 域名
- 公司信息
- 序列号
- 签名信息

CA 机构有很多，客户端里会维护一套所有权威的 CA 机构的公钥用于解密。客户端获取到证书里的机构信息之后，就会取出对应机构的公钥来解析证书里的签名和服务器发来的公钥信息。用签名来校验对方的身份，若校验通过，就可以顺利地使用当前解读出的公钥进行通信了。这个过程，就是“第三方认证”。

}

40 从 HTTP 到 HTTP2.0

42 跨域解决方案