03 | 密码学基础：如何让你的密码变得“不可见”？

2019-12-09 何为舟

讲述：何为舟
时长 17:52 大小 16.37M

你好，我是何为舟。

上一讲，我们学习了黄金法则的三部分核心内容：认证、授权、审计。它们描述了用户在使用应用的各个环节，我们需要采取的安全策略。

在掌握了黄金法则之后，你就能以在安全发展规划上的宏观能力，赢得面试官的认可。接下来，他想考验一下你对安全具体知识的理解，以此来判断你能否将安全发展落地。于是，他问了一个非常基础的问题：你懂加解密吗？

可以说，密码学是“黄金法则”的基础技术支撑。失去了密码学的保护，任何认证、授权、审计机制都是“可笑”的鸡肋。

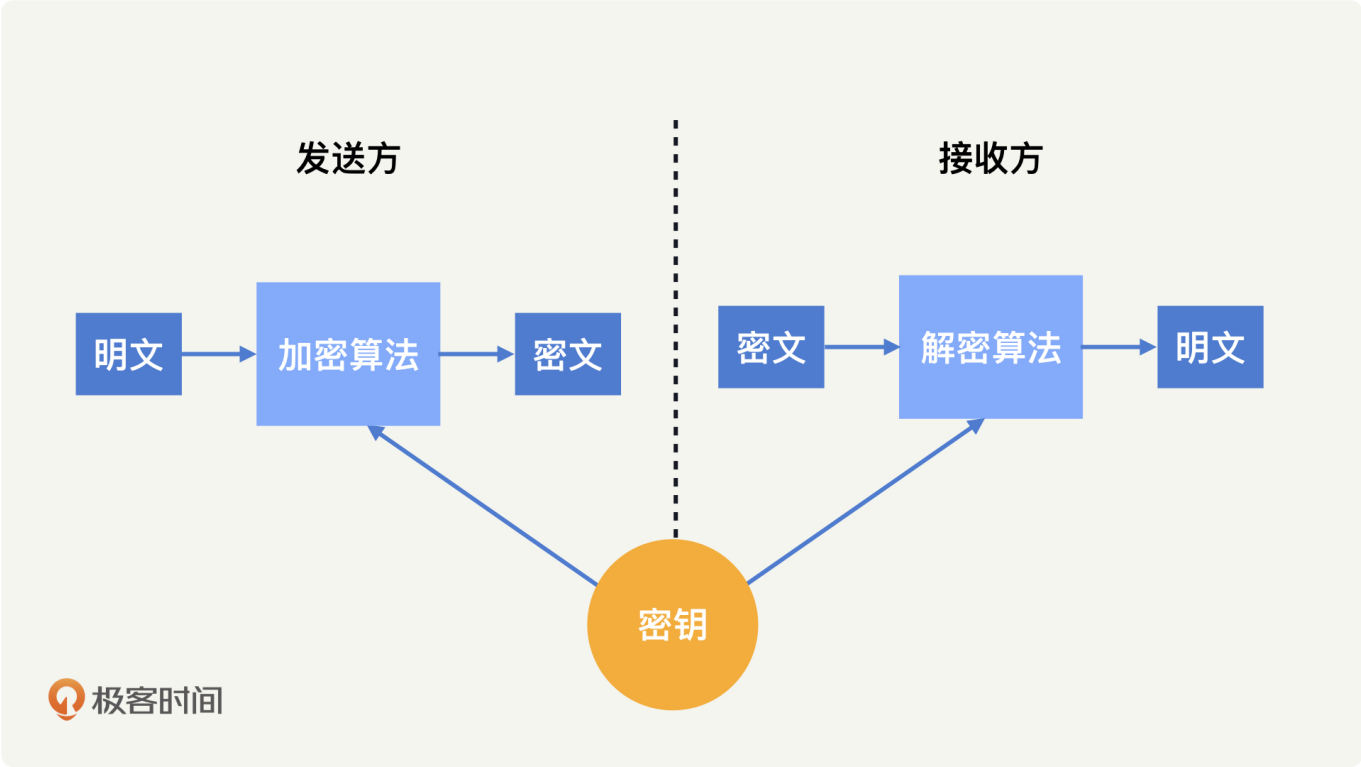
在实际的生活工作中经常会有这样的场景发生：多个用户共用一个 Wi-Fi 来上网、共用一个服务器来跑任务；多个进程共用一个数据库来完成数据存储。在这些场景中，多方交互都通过一个共同的通道来进行，那我们该如何保障其中内容的 CIA 呢？这就需要用到各种加密技术了。今天，我们就一起来学习密码学相关的知识。

首先，我先来普及一个语文知识。密钥中的钥，发音为 yuè，不是 yào。虽然通常情况下，你按正常发音读的话，别人都会听成“蜜月”。但是，我们还是要用正确、专业的发音。

接下来，我来介绍一些经典的密码学算法：对称加密算法、非对称加密算法和散列算法。这些算法的具体实现不是咱们课程的重点，而且本身的过程也非常复杂。在安全这块内容里，你只需要明确了解这些算法的概念及其优缺点，就足够你去选取合适的加密算法了。

对称加密算法

首先，我们来看对称加密算法。所谓对称加密，代表加密和解密使用的是同一个密钥。概念很简单，但是也很不具体、直观。为了帮助你理解，我把具体的加解密过程，画了一张图，你可以看一下。



下面我来具体讲讲这个过程，如果我想给你发一段消息，又不想被其他人知道。那么我作为发送方，会使用加密算法和密钥，生成消息对应的密文；而你作为接收方，想要阅读消息，

就需要使用解密算法和一个同样的密钥，来获得明文。

我们常见的经典对称加密算法有 DES、IDEA、AES、国密 SM1 和 SM4。下面我们一起来具体看看。

第一种对称加密算法是 **DES**（数据加密标准，Data Encryption Standard）。

DES 应该是最早的现代密码学算法之一。它由美国政府提出，密钥长度为 56 位。目前，它暴力破解 56 位密码的时间，已经能控制在 24 小时内了。

DES 实际上是一个过时的密码学算法，目前已经不推荐使用了。关于 DES，还有一点特别有意思。DES 包含一个关键模块：S 盒，其设计的原理一直没有公开。因此，很多人都相信，这个 S 盒中存在后门，只要美国政府需要，就能够解密任何 DES 密文。

第二种对称加密算法是 **IDEA**（国际数据加密算法，International Data Encryption Algorithm）。

IDEA 由瑞士研究人员设计，密钥长度为 128 位。对比于其他的密码学算法，**IDEA 的优势在于没有专利的限制**。相比于 DES 和 AES 的使用受到美国政府的控制，IDEA 的设计人员并没有对其设置太多的限制，这让 IDEA 在全世界范围内得到了广泛地使用和研究。

第三种需要了解的对称加密算法是 **AES**（高级加密标准，Advanced Encryption Standard）。

在 DES 被破解后，美国政府推出了 AES 算法，提供了 128 位、192 位和 256 位三种密钥长度。通常情况下，我们会使用 128 位的密钥，来获得足够的加密强度，同时保证性能不受影响。目前，**AES 是国际上最认可的密码学算法**。在算力没有突破性进展的前提下，AES 在可预期的未来都是安全的。

最后一种是国密 **SM1**（SM1 Cryptographic Algorithm）和 **SM4**（SM4 Cryptographic Algorithm）。

我们知道，密码学作为安全的基础学科，如果全部依靠国外的技术，对于国家安全可能产生不利影响。因此，中国政府提出了一系列加密算法。其中，国密算法 SM1 和 SM4 都属于

对称加密的范畴。SM1 算法不公开，属于国家机密，只能通过相关安全产品进行使用。而 SM4 属于国家标准，算法公开，可自行实现使用。国密算法的优点显而易见：**受到国家的支持和认可。**

借助下面的对比情况表，相信你会对这几种对称加密算法有更清晰的认识。

	密钥长度	加密强度	性能	版权
DES	56	弱	快	美国
3DES	168	中	慢	美国
IDEA	128	强	中	瑞士
AES	128、192、256	强	快	美国
SM1	128	强	未测试	中国（算法保密）
SM4	128	强	未测试	中国（算法公开）

极客时间

现在你应该对几种经典的对称加密算法有了初步地了解。接下来，我们来看一看它们是如何应用的。

在加密通信中（如 HTTPS、VPN、SSH 等），通信双方会协商出一个加密算法和密钥，对传输的数据进行加密，从而防止第三方窃取。在类似数据库加密这种存储加密技术中，通信双方也是将存储空间中的数据进行加密，这样即使硬盘被物理窃取，也不会导致信息丢失。在公司内部，为了避免用户的 Cookie 和隐私信息发生泄漏，也需要对它们进行加密存储。

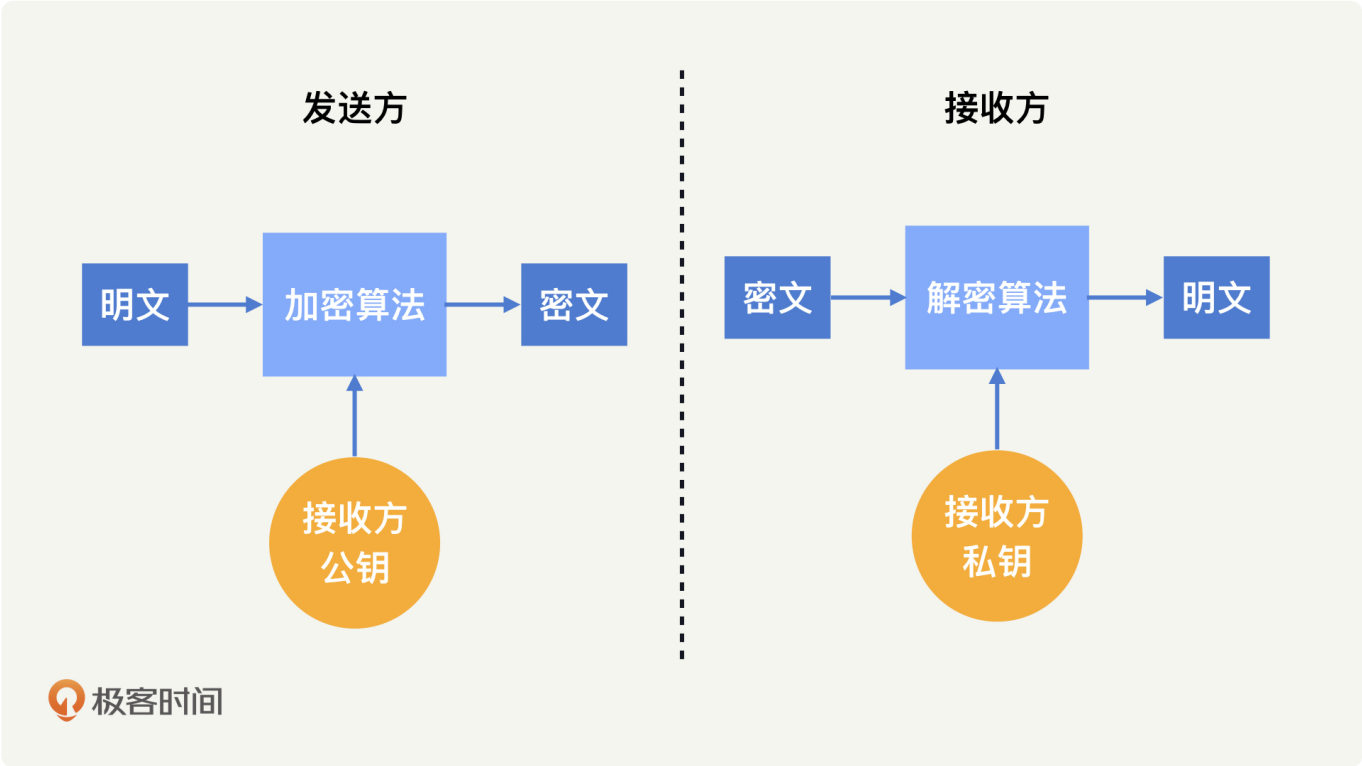
对于大部分公司来说，选取 AES128 进行加解密运算，就能获得较高的安全性和性能。如果是金融或政府行业，在涉及国家层面的对抗上，有一定的合规需求，则需要应用国密算法。

另外，在选取加密算法的时候，存在不同的分组计算模式：ECB/CBC/CFB/OFB/CTR。这些模式的具体细节不是我们学习的重点，在这里就不展开了。你需要知道的是：选取 CBC

和 CTR 这两种推荐使用的模式就可以满足大部分需求了，它们在性能和安全性上都有较好的保证。

非对称加密算法

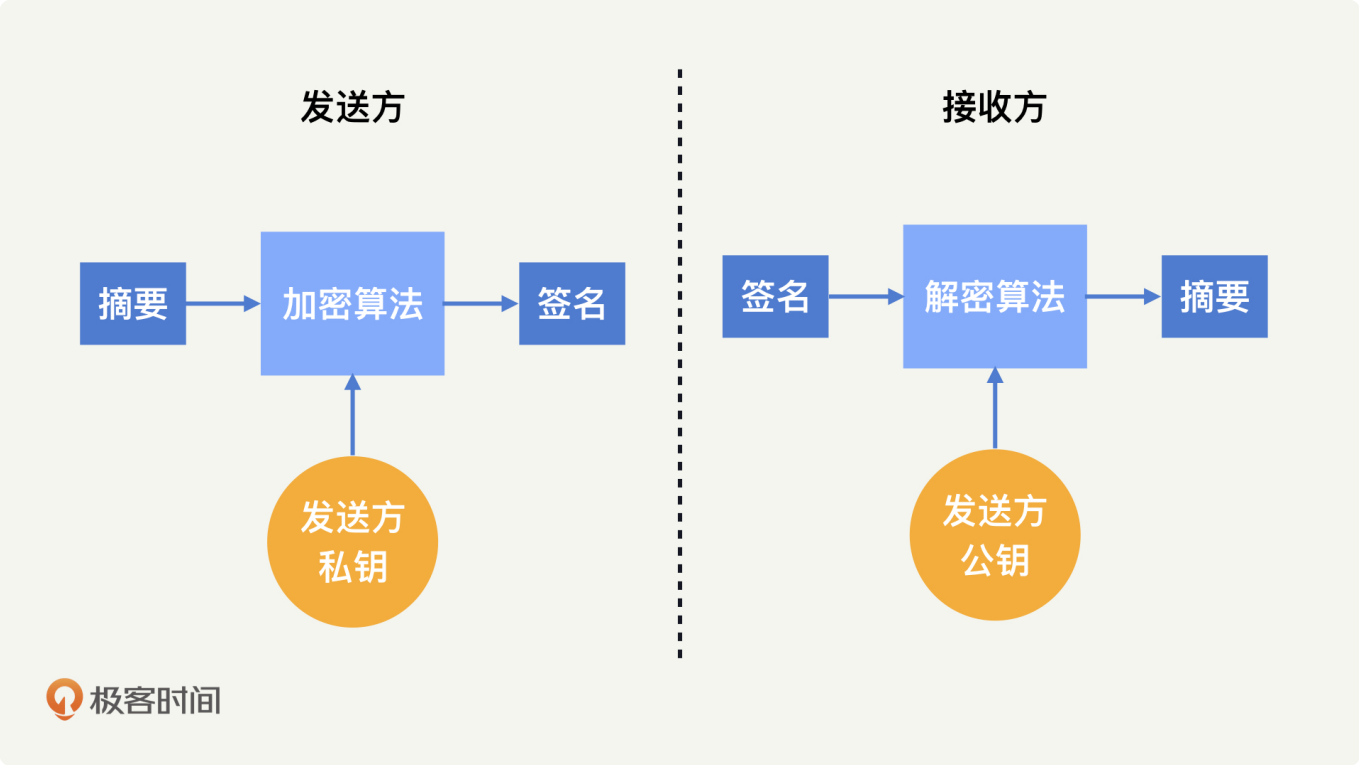
有对称就一定会有非对称。非对称加密代表加密和解密使用不同的密钥。具体的加解密过程就是，发送方使用公钥对信息进行加密，接收方收到密文后，使用私钥进行解密。具体我也画了一张图，你可以和上面的对称加密算法的图一起对照着看一下。可以看到，非对称加密和对称加密算法的最大区别就是，加密和解密使用的密钥是不同的。



当使用对称加密算法的时候，你不仅要跟每一个通信方协定一个密钥，还要担心协商过程中密钥泄漏的可能性。比如，我当面告诉了你一个密码，怎么保证不被偷听呢？而在非对称加密算法中，公钥是公开信息，不需要保密，我们可以简单地将一个公钥分发给全部的通信方。也就是说，我现在就可以告诉你一个公钥密码，即使这意味着所有阅读这篇文章的人都知道了这个密码，那也没关系。因此，非对称密钥其实主要解决了密钥分发的难题。

除了加密功能外，大部分的非对称算法还提供签名的功能。这也就是说，我们可以使用私钥加密，公钥解密。一旦接收方通过公钥成功解密，我们就能够证明发送方拥有对应的私钥，也就能证实发送方的身份，也就是说，私钥加密就是我们说的签名。

你还可以这样理解，比如我现在和你说话，内容经过了我的私钥加密，你用公钥解得了明文。因为私钥只有我拥有，所以只有我能够发出这段话来，别人都不可能。这也就是说，我不可能狡辩称这段话不是我说的。



所有的非对称加密算法，都是基于各种数学难题来设计的，这些数学难题的特点是：正向计算很容易，反向推倒则无解。经典的非对称加密算法包括：**RSA、ECC 和国密 SM2**。接下来，我们一个个来看。

我们先看第一种非对称加密算法 **RSA**（RSA 加密算法，RSA Algorithm）。

RSA 的数学难题是：两个大质数 p、q 相乘的结果 n 很容易计算，但是根据 n 去做质因数分解得到 p、q，则需要很大的计算量。RSA 是比较经典的非对称加密算法，它的主要优势就是**性能比较快**，但想获得较高的加密强度，需要使用很长的密钥。

我们再来看第二种 **ECC**（椭圆加密算法，Elliptic Curve Cryptography）。

ECC 是基于椭圆曲线的一个数学难题设计的。目前学术界普遍认为，椭圆曲线的难度高于大质数难题，160 位密钥的 ECC 加密强度，相当于 1088 位密钥的 RSA。因此，**ECC 是目前国际上加密强度最高的非对称加密算法。**

最后一种是国密 **SM2**（SM2 Cryptographic Algorithm）。

国密算法 SM2 也是基于椭圆曲线问题设计的，属于国家标准，算法公开，加密强度和国际标准的 ECC 相当。而**国密的优势在于国家的支持和认可**。

好了，这 3 种非对称加密算法的优缺点我也总结成了一张表格，你可以看一看。

对比前提：同等密钥长度、加密强度

	加密强度	密钥生成性能	加解密性能	版权/专利
RSA	弱	慢	快	RSA公司
ECC	强	快	慢	争议中
SM2	强	快	慢	中国

极客时间

我们前面说了，对比于对称加密算法，非对称加密算法最大的优势就是解决密钥分发的问题。因此，现在大部分的认证和签名场景，其实使用的都是非对称加密算法。比如，在 SSH 登录、Git 上传等场景中，我们都可以将自己的公钥上传到服务端，然后由客户端保存私钥。

那么，如果你遇到需要使用非对称加密的场景（比如多对一认证），我推荐你使用 ECC 算法。

散列算法

散列算法应该是最常见到的密码学算法了。大量的应用都在使用 MD5 或者 SHA 算法计算一个唯一的 id。比如 Git 中的提交记录、文件的完整性校验、各种语言中字典或者 Map 的实现等等。很多场景下，我们使用散列算法并不是为了满足什么加密需求，而是利用它可以对任意长度的输入，计算出一个定长的 id。

作为密码学的算法，散列算法除了提供唯一的 id，其更大的利用价值还在于它的不可逆性。当用户注册，提交账号密码时，作为一个安全的应用，是绝对不能够存储明文密码的。因此，我们对用户的密码通过散列算法进行计算，存储最终的散列值。

在后续登录的过程中，我们如果计算出的用户提交的密码的散列值和你存储的散列值一致，就可以通过验证了。这样一来，任何人（即使是内部员工）都不知道用户真实的密码是什么，而用户也能够完成密码的校验。

除了刚才说的不可逆性，在密码学上，我们对散列算法的要求还有：鲁棒性（同样的消息生成同样的摘要）、唯一性（不存在两个不同的消息，能生成同样的摘要）。

经典的散列算法包括 MD5、SHA、国密 SM3。下面我们逐一来看。

我们先来看第 1 种，**MD5**（消息摘要算法，Message-Digest Algorithm 5）。

MD5 可以用来生成一个 128 位的消息摘要，它是目前应用比较普遍的散列算法，具体的应用场景你可以自行参阅。虽然，因为算法的缺陷，它的唯一性已经被破解了，但是大部分场景下，这并不会构成安全问题。但是，如果不是长度受限（32 个字符），我还是不推荐你继续使用 MD5 的。

第 2 种是 **SHA**（安全散列算法，Secure Hash Algorithm）。

SHA 是美国开发的政府标准散列算法，分为 SHA-1 和 SHA-2 两个版本，SHA-2 细分的版本我们就不介绍了。和 MD5 相同，虽然 SHA 的唯一性也被破解了，但是这也不会构成大的安全问题。目前，SHA-256 普遍被认为是相对安全的散列算法，也是我最推荐你使用的散列算法。

第 3 种是国密 **SM3**（SM3 Cryptographic Algorithm）。

国密算法 SM3 是一种散列算法。其属于国家标准，算法公开，加密强度和国际标准的 SHA-256 相当。和国密 SM2 一样，它的优势也在于国家的支持和认可。

上述算法的相关对比情况，我也总结了一下，如下表所示：

	长度	冲突概率	安全性	性能
MD5	128	中	中	中
SHA	160、256	低	高	慢
SM3	256	低	高	未测试

极客时间

另外，我们在使用散列算法的时候，有一点需要注意一下，一定要注意加“盐”。所谓“盐”，就是一串随机的字符，是可以公开的。将用户的密码“盐”进行拼接后，再进行散列计算，这样，即使两个用户设置了相同的密码，也会拥有不同的散列值。同时，黑客往往会提前计算一个彩虹表来提升暴力破解散列值的效率，而我们能够通过加“盐”进行对抗。“盐”值越长，安全性就越高。

总结

好了，我们来总结一下这一节，你需要掌握的重点内容。

在这节课中，我对各种加密算法和应用场景进行了全面的介绍。密码学是一门深奥的学科，而作为密码学的使用者，你只需要正确地理解各类算法的特性和功能，就可以满足日常的应用需求了。

总的来说，在使用的时候，你要记住下面这些内容：对称加密具备较高的安全性和性能，要优先考虑。在一对多的场景中（如多人登录服务器），存在密钥分发难题的时候，我们要使用非对称加密；不需要可逆计算的时候（如存储密码），我们就使用散列算法。

在具体算法的选取上，你只需要记住：对称加密用 AES-CTR、非对称加密用 ECC、散列算法用 SHA256 加盐。这些算法就能够满足大部分的使用场景了，并且在未来很长一段时间内，都可以保持一个较高的安全强度。

思考题

通过今天的学习，相信你已经了解了密码学的各种概念和知识。对于这些加密算法，哪些你比较了解或者使用过呢？可以谈谈你的想法。

欢迎留言和我分享你的思考和疑惑，也欢迎你把文章分享给你的朋友。我们下一讲再见！

上一篇　02 | 安全原则：我们应该如何上手解决安全问题？

下一篇　04 | 身份认证：除了账号密码，我们还能怎么做身份认证？

精选留言 (25)

写留言

Geek_f7f72f
2019-12-10

sha3，blake2不介绍下吗，相比所谓的国密，应用范围更广吧

作者回复: 现有技术没有出现明显的问题，所以大家还是习惯性使用传统的算法，这些新算法的替代性和普及性不会那么高。
另外，不要低估国密，最近和一些数据安全的人聊，他们表示如果只做国内业务的话，最好都用国密。因为说不定哪天等保或者国内的数据安全法，就强制要求国密了。

6

rocedu
2019-12-10

密钥的发音问题，依据在哪？

展开

作者回复: 你好，感谢你的留言。如果一定要追究依据的话，可以查阅字典，只有钥匙读yao。这其实是北京人的方言发音，钥其实是只有yue这个读法的。
不过嘛，字典也会适应潮流，将错就错，比如‘空穴来风’的意思。所以，科普yue仅仅是我的偏执，不需要认同。

4

Geek_f7f72f
2019-12-10

TLS很早就弃用了IDEA，标记为不安全，是因为理论上的缺陷，还是其他原因？

作者回复: 你好，感谢你的留言。这个问题之前还真没注意到，特地查了一下。看到的原因也就是“due to the availability of faster algorithms”。也就是说，已经有更快更好的AES了，就没有再使用IDEA的意义了。

3

return
2019-12-09

老师 请教一下：

唯一性（不存在两个不同的消息，能生成同样的摘要）。

我理解 散列算法做不到这一点吧， 消息是任意的 是个无限集，但是散列值 是固定长度...

展开

作者回复: 理论是这样没错，但实际使用时，你肯定不会有那么多数据需要去做散列。所以，追求的是在有限数据量下，碰撞概率几乎为0。

1 3

Ender0224
2019-12-12

请问如何保证对称秘钥在非可信环境中的安全传输，是不是只能只能使用非对称算法先加密才可以，有其他方案么

展开

作者回复: 你好，感谢你的留言。在https中使用了DH密钥交换算法实现的。可以想象成一边出一半的密钥，然后就能够拼成一个完整密钥。因篇幅限制，课程中没有具体讲。

1

eason2017
2019-12-10

老师好，请问两段用相同密钥做计算，然后，后端做计算后的值比较是否相同。这种算作是对称还是非对称加密呀

展开

作者回复: 只是加密后密文的比对？那似乎没有加密的必要，散列算法就可以了。

1 1

Vokey
2019-12-19

《现代汉语词典（第7版）》：

【密钥】mìyuè （口语中多读mìyào）

展开

鸵鸟
2019-12-19

非对称加密推荐ECC的原因是什么呢？当前在手机PC的系统安全这些业务上，苹果高通都是使用的RSA，而国内一线手机厂商开始在布局国密SM2同时支持RSA，请问ECC的优势在哪里呢？

小美
2019-12-16

老师好，我们前端需要加密传输数据到后端，怎么防止密钥泄露呢。攻击者通过反编译能拿到前端加密用的密钥，

展开

作者回复: 非对称加密，公玥不需要保密，存储到前端即可。

1

Geek_f7f72f
2019-12-16

评论里有提到Bcrypt, 它目前的安全性如何，有没有更好的替代?

作者回复: 我理解的Bcrypt其实就是散列+盐的封装实现，作为一种最佳实践的封装，安全性上应该不会出现太大问题。

活明白
2019-12-15

2010年1月1日《密码法》正式实施，商用密码应用有相应的规定，特别是关键信息基础设施商用密码的使用。未使用或者未按照要求开着商用密码应用安全性评估的，可能被处罚，情节严重的处十万元以上一百万以下罚款，对直接负责的主管人员处一万元以上十万元以下罚款（可参考第三十七条）。

展开

1

Geek_98dc22
2019-12-14

国密算法的价值在哪里呢？ 从描述来看还是参考ecc aes等算法，但性能还不确定。既然已经有公开的高可用算法，为什么不用或者参与该算法的研究中，进一步提升这类知名加密算法呢

作者回复: 这就是国家层面的考量了。一方面，是对加密算法的安全性考量，比如DES中可能的后门。另一方面，也是对专利版权的保护，毕竟自己的专利，自己才有可控性。这就和芯片一样，现在用外国的没啥问题，哪天它不让你用了呢?

旺旺
2019-12-14

非对称效率高，对称加密效率太低，所以一般都是非对称加密传递对承加密密钥，用对称加密传输数据，然后有效期后更换对称加密的秘钥。

展开

作者回复: 第一句说反了。对称效率高，非对称效率低。

张望
2019-12-14

建议程序员们也多关注国家法律法规要求，我国密码法马上正式实施了，其实对于加密算法选用就有了明确的法律要求，没有影响力的小软件还好说，做大了的软件一定会被监管到的，到时再更换全套加密算法成本也会很高的。

展开

作者回复: 嗯，最近跟同行聊的时候。他们也表示公司内部开始推进国密算法的应用了，为了避免以后又要改。不过吧，这个改动还是有难度的，毕竟国际算法都用了这么多年了，老旧代码改动成本不容小觑。

p4ny
2019-12-12

蜜月，秘药，公约，公药.....感觉都挺顺嘴的怎么办.....

展开

作者回复: 哈哈，小细节，不重要～

Cy23
2019-12-12

嗯，最后记住这几个就可以了，对称加密用 AES-CTR、非对称加密用 ECC、散列算法用 SHA256 加盐

攻城狮Ra
2019-12-11

我们常见的破解软件用的密钥，是不是SHA256加盐，盗版系统又是怎么一回事呢

作者回复: 我知道的主要两种方式吧。一种是序列码存在一定的规则，软件在本地只是验证序列码是否符合这个规则。常见的带序列码生成器的，应该都是这种模式。另一种破解就是直接篡改软件的验证流程，跳过验证这个步骤，破解补丁就是用来干这个事情的。

张智凯
2019-12-11

问下保存盐后密码的散列值的时候，用户的盐也要跟用户的标识关联起来保存吧

作者回复: 你好，感谢你的留言。是的，肯定要关联起来，只是不需要保密而已。

张诚
2019-12-10

我们用的是RSA的非对称加密对于请求参数中的敏感参数进行的加密，用MD5+盐，对整个请求参数的json串进行散列计算签名。以前也用到过DES和AES对称加密。

王蒙
2019-12-10

请问如何防止session劫持？项目通过session保存用户信息，sessionid存在客户端浏览器cookie里，通过抓包很容易抓到sessionid，拿到sessiondid，就可以随意操作受害用户，有没有比较好的解决方案

作者回复: https。本地你自己抓自己的包肯定能抓到。但劫持是网络层次的事情，通过https，黑客是没办法在网络中抓用户的包的。