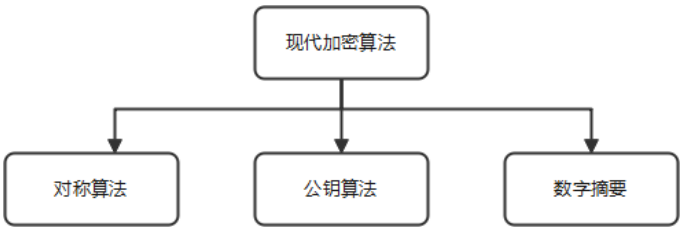
加密算法介绍

简介

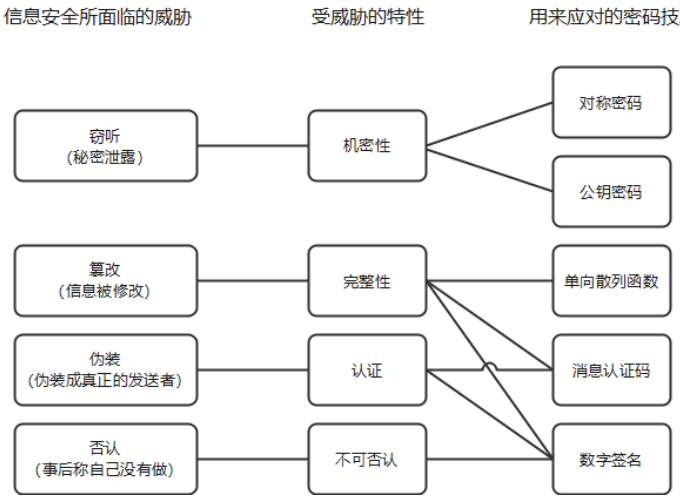
在现代信息社会信息中，快捷的信息传递也带来了更多的信息安全问题。信息安全问题是牵涉了信息的发送者、信息、信息接收者以及第三方的系统问题。其中的第三方是广义的，除了少量有恶意获取中间信息意图的第三方外，任何不属于发送者及接收者系统内的人员及设备都应看做第三方。因此便产生了信息加密等一系列需求。

密码技术由来已久，可以简单分为经典密码技术以及现代密码技术。现代密码技术基于**计算机加密**，应用上不仅仅是提供基于密码的机密性，还用于检验消息是否被篡改的完整性、以及用于确认对方是否是本人的认证等都是密码技术的重要部分。

文章中所介绍的加密算法都是现代加密法的成果。主要有如下三大类:



除此之外，还有其他应用在加密工作中的工具如：消息认证码、数字签名、伪随机数生成器。相关技术的应用场景如下：


信息安全所面临的威胁
受威胁的特性
用来应对的密码技术
窃听
（秘密泄露）
机密性
对称密码
公钥密码
篡改
（信息被修改）
完整性
单向散列函数
伪装
（伪装成真正的发送者）
认证
消息认证码
否认
（事后称自己没有做）
不可否认
数字签名

密钥

日常生活中用户经常会使用各式 “密码”，如银行卡以及电子账号。但实际上那些在密码算法中并不是严格意义上的密码（或者称为密钥）。对于系统来说，我们手中的“密码”实际上应该是系统认证的**口令。**系统内部往往会根据口令使用单向散列算法等转换为系统真正的密码（密钥）。

密码算法中需要密钥去 “关闭”与“打开”数据信息的大门。现实世界中的钥匙是形状复杂的金属制品。而密码算法中的密钥则是一串数字。密钥的**数字大小本身并不重要**，重要的是**密钥空间的大小，**也就是可能出现的密钥的总数量，因为密钥空间越大，进行暴力破解就越困难（这是一个概率问题）。密钥空间的大小是由**密钥长度**决定的。

如下文的对称密码 DES 的密钥**有效位数**是 56 位（7 字节），因为是在计算机系统中，因此使用二进制举例表示如下：

01011001 10001010 10001100 10010011 00110100 10100101 10100111

密钥和明文是等价的。信息的机密性不应该依赖于密码算法本身，而是应该依赖于妥善保管的密钥。这是密码世界的常识之一。

对于加密算法的无法破解，是指其在空间和时间上不具备实现的条件。如某个加密算法，采用暴力破解，在现有的计算资源条件下，需要花费 50 年时间，那么就可以认为其无法破解（机密信息往往具有很强的时效要求）。随着计算机运算能力的提升，采用暴力破解的方式破解密码的时间也会缩短。

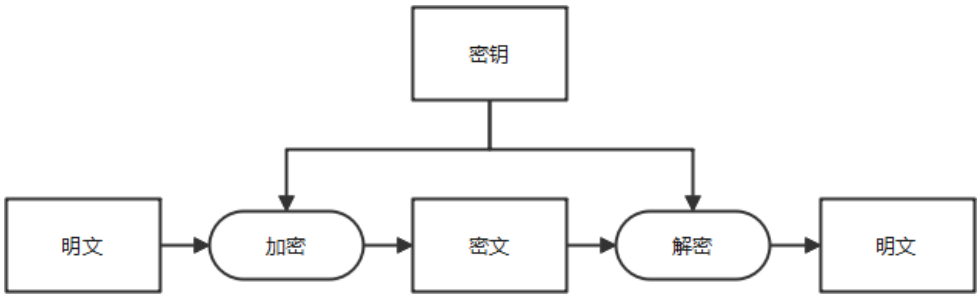
生成密钥的最好方法就是使用随机数，因为密钥需要具备不易被他人推测的性质。在可能的情况下最好使用能够生成密码学上的随机数的硬件设备，过去的设计中会通过伪随机数生成器进行随机数生成，现在则有更为安全的**真随机数生成器（TRNG）**，往往借助物理噪声生成随机数。

对称加密算法

对称加密算法也叫共享密钥加密算法、单密钥加密算法。采用单密钥的加密方法，密钥同时用作信息的加密和解密，即解密算法为加密算法的逆算法。因此在知道了加密算法后也就知道了解密算法。

对称加密算法顾名思义，其解密过程是加密过程的逆过程，在加密及解密过程中使用的是同一个密钥。

对称加密的流程如下图



衡量对称加密算法优劣的取决于其密钥的长度。密钥越长，破解需要测试的密钥就越多，破解这种算法的难度就越大。其安全性取决于是否有未经授权的人获得了对称密钥。

常用对称加密算法主要有:

- DES：DES（Data Encryption Standard）是 1977 年美国联邦信息处理标准（FIPS）中所采用的一种对称密码（FIPS 46-3）。DES 一直以来被美国以及其他国家的政府和银行等广泛使用。DES 使用一个 64 位的密钥来加密每个块长度为 64 位的明文，并生成每个块长度为 64 位的密文。DES 是一个包含 16 个阶段的替换-置换加密方法。然而，随着计算机的进步，现在 DES 已经能够被暴力破解，强度大不如前了。RSA 公司举办的破译 DES 密钥的比赛，已经可以在 24 小时内对密钥进行破解。DES 是以 64 比特的明文（比特序列）为一个单位来进行加密的，这个 64 比特的单位称为分组。DES 每次只能加密 64 比特的数据，如果要加密的明文比较长，就需要对 DES 加密进行迭代（反复），而迭代的具体方式就称为模式（mode）。DES 的密钥长度是 64 比特，但是由于每隔 7 个比特会设置一个用于错误检查的比特，因此实质上其密钥长度是 56 比特。

- 3DES：3DES 是为了增加 DES 的强度，将 DES 重复 3 次所得到的的一种密码算法。明文经过三次 DES 处理才能变成最后的密文，由于 DES 密钥的长度实质上是 56 比特，因此三重 DES 的密钥长度就是 56*3=168 比特。

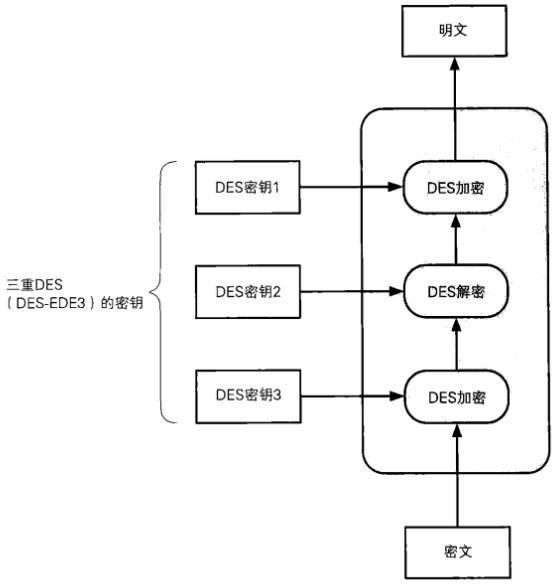


图 3-7　三重 DES 的加密

在 3DES 加密过程中加入解密过程是为了兼容 DES，只要三次操作的密钥相同，那么 3DES 加密过程也就相当于 DES 加密过程。尽管 3DES 目前还被银行机构使用，但其处理速度不高，而且在安全性方面也逐渐显现出一些问题。

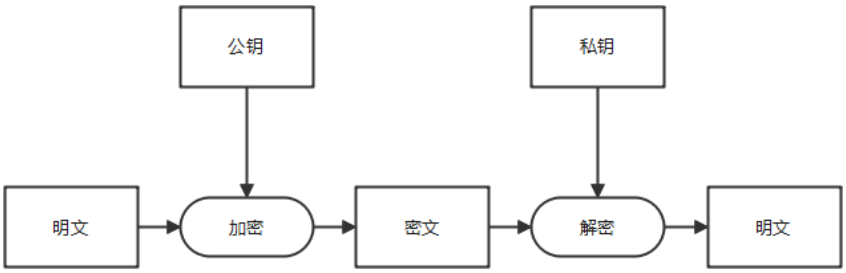
- AES:AES（Advanced Encryption Standard）是取代其前任标准（DES）而成为新标准的一种对称密码算法。现在的 AES 实际上是在众多密码算法中筛选出来的，其原有名称叫做 Rijndael。Rijndael 的分组长度为 128 比特，密钥长度可以以 32 比特为单位在 128 比特到 256 比特的范围内进行选择（不过在 AES 的规格中，密钥的长度只有 128、192 和 256 比特三种）。

在上述三者选择上，DES 已经不建议用于新的用途，因为使用暴力破解法已经能够在现实时间内对其破译。由于兼容性的因素 3DES 在今后还会使用一段时间，但会逐渐被 AES 所取代。其他还有一些加密算法，此处不进行介绍。

非对称加密算法

在加密消息传递中存在一方提前将密钥传递给另一方的过程。对称加密算法中的加密和解密使用的是同一个密钥。如果密钥在传递过程中被窃取，那么加密信息将不再安全。由于**密钥配送问题**的存在，衍生出了一些解决此问题的方案。而非对称密码（也称公钥密码）就是解决密钥配送问题的方案之一。在对称密码中，加密密钥和解密密钥是相同的，但非对称密码中，加密密钥和解密密钥却是不同的。只要拥有加密密钥，任何人都可以加密，但没有解密密钥时无法解密的。因此，公钥密码的一个重要性质，就是拥有解密密钥的人才能够进行解密。

接收者事先将加密密钥发送给发送者，这个加密密钥即使被窃取或者也没有问题。发送者使用加密密钥对通信内容进行加密并发送给接收者，而只有拥有解密密钥的人才能够进行解密（即接收者本身就拥有解密密钥）。这样一来，就不需要将解密密钥配送给接受者了。



公钥加密大多数是基于求解难题的。也就是说，是很难解决的问题。人们往往把大数字的因子分解或找出一个数的对数之类的问题作为公钥系统的基础。

公钥算法主要有：

- RSA 算法：RSA 是一种公钥密码算法，它的名字是由它的三位发明者的姓氏首字母组成（Rivest-Shamir-Leonard）。对极大整数做因数分解的难度决定了 RSA 算法的可靠性，到目前为止，世界上还没有任何可靠的攻击 RSA 算法的方式。只要其钥匙的长度足够长，用 RSA 加密的信息实际上是不能被解破的。RSA 可以被用于公钥密码和数字签名。
- Rabin 算法：基于费马小定理的大素数检测算法。
- El Gamal 算法：是一个基于迪菲-赫尔曼密钥交换的非对称加密算法。其安全性依赖于计算有限域上离散对数这一难题
- 椭圆曲线算法（ECC）：是一种基于椭圆曲线数学的公开密钥加密算法。椭圆曲线算法是代替 RSA 的强有力的竞争者，有如下特点：安全性能更高，如 160 位 ECC 与 1024 位 RSA、DSA 有相同的安全强度；计算量小，处理速度快，在私钥的处理速度上远比 RSA 快得多；存储空间占用小，ECC 的密钥尺寸和系统参数与 RSA 相比要小得多， 所以占用的存储空间小得多；带宽要求低，使得 ECC 具有广泛的应用前景。

在非对称加密算法中通过公钥加密的数据，只能通过私钥解开。通过私钥加密的数据，只能通过公钥解开。其主要局限在于，这种加密形式的速度相对较低，通常仅在关键时刻才使用该算法。

混合密码

公钥密码的处理速度只有对称密码的几百分之一。不适用于处理大量数据的加密。但对称密码又存在着**密钥配送问题。**为了系统的性能及安全性。往往会在系统的不同部分及阶段会将对称密码及公钥密码进行混合使用。即用对称密码提高处理速度，用公钥密码解决密钥配送问题。这样的方式即为混合密码

系统。

著名的密码软件 PGP 以及网上的密码通信所使用的的 SSL/TLS 都运用了混合密码系统。

数字摘要算法

又称哈希算法、散列算法，是一种单向算法，它通过对数据内容进行散列得到一个固定长度的密文信息（信息是任意长度，而摘要是定长）。即用户可以通过哈希算法对目标信息生成一段特定长度的唯一的 Hash 值，却不能通过这个 Hash 值重新获得目标信息。该算法不可逆。所以理解是与上述的加密方式均有差别，可以看作是**无法从密文（Hash 值）再去获得明文的。**

数字摘要算法有一个输入和一个输出，其中输入称为消息，输出称为散列值。单向散列函数可以根据消息的内容计算出散列值，而散列值就可以被用来检查消息的完整性。

散列值的长度和消息的长度无关。无论消息是 1 比特，还是 100MB，甚至是 100GB，单向散列函数都会计算出固定长度的散列值。以 SHA-1 单向散列函数为例，它所计算出的散列值的长度永远都是 160 比特。

理论上一个 Hash 值是可能对应无数多个原文的，算法把无限的映射成有限，因此可能会有碰撞（两个不同的信息，算出的摘要相同）。但是在现实中，由于原文的长度有限，所以想要出现两段原文对应同一个 Hash 值的是及其困难的，即几乎不会出现碰撞的情况，这就使得该算法得以在现实中可以应用。

单向散列函数的一些应用有：1.检测软件是否被篡改 2.基于口令的加密 3.消息认证码 4.数字签名 5.伪随机数生成器 6.一次性口令

数字摘要算法主要有

- MD5：一种被广泛使用的密码散列函数，可以产生出一个 128 位的散列值，用于确保信息传输完整一致。MD5 的强抗碰撞性已经被攻破，现在已经能够产生具备相同散列值的两条不同消息，对于需要高度安全性的数据，专家一般建议改用其他算法，如 SHA-2。
- SHA-1：SHA-1 可以生成一个被称为消息摘要的 160 位散列值，散列值通常的呈现形式为 40 个十六进制数。SHA-1 已经不再视为可抵御有充足资金、充足计算资源的攻击者。自 2010 年以来，许多组织建议用来 SHA-2 或 SHA-3 替换 SHA-1。

- SHA-2：是一种密码散列函数算法标准，其输出长度可取 224 位、256 位、384 位、512 位，分别对应 SHA-224、SHA-256、SHA-384、SHA-512。目前 SHA-2 算是安全的加密算法。
- SHA-3：基于 Keccak 算法，是与 SHA-2 不同的设计方式，可避免已有的攻击方式，而且能够提供 SHA-2 不具备的一些性能。

应用场景

对称加密算法

- 无需进行密钥交换的场景，如内部系统，事先就可以直接确定密钥
- 防止明文传输数据被窃取的
- 加解密速度快，适合数据内容比较大的加密场景

公钥算法

- 适用于需要密钥交换的场景，如互联网应用，无法事先约定密钥
- 与对称加密算法结合。利用非对称加密算法安全性较好的特点，传递对称加密算法的密钥。利用对称加密算法加解密速度快的特点，进行数据内容比较大的加密场景的加密。如 HTTPS。

数字摘要算法

- 下载文件时，文件的完整性校验
- 接口交互时，交互数据的完整性校验
- 数字证书的指纹生成算法
- 密码的正确性校验，即只需要验证密码的摘要是否相同即可确认密码是否相同，同时也保证让密码以密文保存，无法被可逆破解。

加密算法的硬件实现

加密算法可以使用软件或者硬件去实现。传统的加密是通过在主机上运行加密软件来实现的，但加密操作能实现的安全性能应从系统层面来看，软件实现的加密功能存在天然缺陷。同时也会占用系统的运算资源。现有的加密算法通常是由系统中的安全加密芯片或者内嵌在 SOC 中实现，而在芯片内部，则通过 IP 核的形式来从硬件上实现加密算法。

加密算法的 IP 核在芯片中，除了完成既定功能以外，还需要考虑针对各式外部攻击的抗攻击设计。

总结

密码是信息安全的一部分，安全加密芯片是其实现形式也是目前理想的实现形式之一，同时也只是安全系统的一部分。

参考文献

1.结城浩(日).图解密码技术

2.斯皮尔曼(美).经典密码学与现代密码学

3. RSA 加密算法 IP 核的设计与实现

https://www.doc88.com/p-9552703845736.html

4. AES 加密算法 IP 核的设计与实现

https://www.doc88.com/p-5374126793849.html?s=rel&id=5