|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **记录名称** | YSM仪器概要设计 | | **记录编码** |  | |
| **密 级** |  | | | | |
| **起草/日期** |  | **审核/日期** |  | **批准/日期** |  |

**记录修改履历**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **修订前版本** | **现行**  **版本** | **修订内容概述** | **起草** | **修订日期** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |

**目 录**

[1 常见加密算法 3](#_Toc4412618)

[1.1 对称加密 3](#_Toc4412619)

[1.2 非对称加密 3](#_Toc4412620)

[1.3 Hash算法 4](#_Toc4412621)

[2 加密算法的选择 4](#_Toc4412622)

[3 其他常见加密算法 5](#_Toc4412623)

[3.1 DES加密算法 5](#_Toc4412624)

[3.2 AES加密算法 5](#_Toc4412625)

[3.3 RSA加密算法 6](#_Toc4412626)

[3.4 Base64加密算法 6](#_Toc4412627)

[3.5 MD5加密算法 6](#_Toc4412628)

[3.6 SHA1加密算法 7](#_Toc4412629)

# 常见加密算法

参考：

https://www.cnblogs.com/colife/p/5566789.html

https://blog.csdn.net/ddffr/article/details/77153127

常见加密算法常见的加密[算法](http://lib.csdn.net/base/31)可以分成三类，对称加密算法，非对称加密算法和Hash算法。

## 对称加密

    指加密和解密使用相同密钥的加密算法。对称加密算法的优点在于加解密的高速度和使用长密钥时的难破解性。假设两个用户需要使用对称加密方法加密然后交换数据，则用户最少需要2个密钥并交换使用，如果企业内用户有n个，则整个企业共需要n×(n-1) 个密钥，密钥的生成和分发将成为企业信息部门的恶梦。对称加密算法的安全性取决于加密密钥的保存情况，但要求企业中每一个持有密钥的人都保守秘密是不可能的，他们通常会有意无意的把密钥泄漏出去——如果一个用户使用的密钥被入侵者所获得，入侵者便可以读取该用户密钥加密的所有文档，如果整个企业共用一个加密密钥，那整个企业文档的保密性便无从谈起。

**常见的对称加密算法：**DES、3DES、DESX、Blowfish、IDEA、RC4、RC5、RC6和AES

## 非对称加密

    指加密和解密使用不同密钥的加密算法，也称为公私钥加密。假设两个用户要加密交换数据，双方交换公钥，使用时一方用对方的公钥加密，另一方即可用自己的私钥解密。如果企业中有n个用户，企业需要生成n对密钥，并分发n个公钥。由于公钥是可以公开的，用户只要保管好自己的私钥即可，因此加密密钥的分发将变得十分简单。同时，由于每个用户的私钥是唯一的，其他用户除了可以可以通过信息发送者的公钥来验证信息的来源是否真实，还可以确保发送者无法否认曾发送过该信息。非对称加密的缺点是加解密速度要远远慢于对称加密，在某些极端情况下，甚至能比非对称加密慢上1000倍。

**常见的非对称加密算法：**RSA、ECC（移动设备用）、Diffie-Hellman、El Gamal、DSA（数字签名用）

## Hash算法

    Hash算法特别的地方在于它是一种单向算法，用户可以通过Hash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。因此Hash算法常用在不可还原的密码存储、信息完整性校验等。

**常见的Hash算法：**MD2、MD4、MD5、HAVAL、SHA、SHA-1、HMAC、HMAC-MD5、HMAC-SHA1

    加密算法的效能通常可以按照算法本身的复杂程度、密钥长度（密钥越长越安全）、加解密速度等来衡量。上述的算法中，除了DES密钥长度不够、MD2速度较慢已逐渐被淘汰外，其他算法仍在目前的加密系统产品中使用。

# 加密算法的选择

    前面的章节已经介绍了对称解密算法和非对称加密算法，有很多人疑惑：那我们在实际使用的过程中究竟该使用哪一种比较好呢？

    我们应该根据自己的使用特点来确定，由于非对称加密算法的运行速度比对称加密算法的速度慢很多，当我们需要加密大量的数据时，建议采用对称加密算法，提高加解密速度。

对称加密算法不能实现签名，因此签名只能非对称算法。

由于对称加密算法的密钥管理是一个复杂的过程，密钥的管理直接决定着他的安全性，因此当数据量很小时，我们可以考虑采用非对称加密算法。

    在实际的操作过程中，我们通常采用的方式是：采用非对称加密算法管理对称算法的密钥，然后用对称加密算法加密数据，这样我们就集成了两类加密算法的优点，既实现了加密速度快的优点，又实现了安全方便管理密钥的优点。

    如果在选定了加密算法后，那采用多少位的密钥呢？一般来说，密钥越长，运行的速度就越慢，应该根据的我们实际需要的安全级别来选择，一般来说，RSA建议采用1024位的数字，ECC建议采用160位，AES采用128为即可。

# 其他常见加密算法

由于计算机软件的非法复制，通信的泄密、数据安全受到威胁，解密及盗版问题日益严重，甚至引发国际争端，所以在信息安全技术中，[加密](http://www.cksis.com/)技术占有不可替代的位置，因此对信息加密技术和加密手段的研究与开发，受到各国计算机界的重视，发展[日新月异](https://www.baidu.com/s?wd=%E6%97%A5%E6%96%B0%E6%9C%88%E5%BC%82&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd)。现在我们就几种常用的加密[**算法**](http://lib.csdn.net/base/datastructure)给大家比较一下。

## DES加密算法

DES加密算法是一种分组密码，以64位为分组对数据加密，它的密钥长度是56位，加密解密用同一算法。DES加密算法是对密钥进行保密，而公开算法，包括加密和解密算法。这样，只有掌握了和发送方相同密钥的人才能解读由DES[加密算法](http://www.cksis.com/blog/category/jiamisuanfa)加密的密文数据。因此，破译DES加密算法实际上就是搜索密钥的编码。对于56位长度的密钥来说，如果用穷举法来进行搜索的话，其运算次数为256。

随着计算机系统能力的不断发展，DES的安全性比它刚出现时会弱得多，然而从非关键性质的实际出发，仍可以认为它是足够的。不过，DES现在仅用于旧系统的鉴定，而更多地选择新的加密标准。

## AES加密算法

AES加密算法是密码学中的高级加密标准，该加密算法采用对称分组密码体制，密钥长度的最少支持为128、192、256，分组长度128位，算法应易于各种硬件和软件实现。这种加密算法是[美国联邦政府](https://www.baidu.com/s?wd=%E7%BE%8E%E5%9B%BD%E8%81%94%E9%82%A6%E6%94%BF%E5%BA%9C&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd)采用的区块加密标准，这个标准用来替代原先的DES，已经被多方分析且广为全世界所使用。

[AES加密算法](http://www.cksis.com/blog/858-aesjiamisuanfa.html)被设计为支持128／192／256位（/32=nb)数据块大小（即分组长度）；支持128／192／256位（/32=nk)密码长度，，在10进制里，对应34×1038、62×1057、1.1×1077个密钥。

## RSA加密算法

RSA加密算法是目前最有影响力的[公钥加密](http://www.cksis.com/blog/892-gongyuejiamisuanfa.html)算法，并且被普遍认为是目前最优秀的公钥方案之一。RSA是第一个能同时用于加密和数宇签名的算法，它能够抵抗到目前为止已知的所有密码攻击，已被ISO推荐为公钥数据加密标准。RSA加密算法基于一个十分简单的数论事实：将两个大素数相乘十分容易，但那时想要，但那时想要对其乘积进行因式分解却极其困难，因此可以将乘积公开作为加密密钥。

## Base64加密算法

Base64加密算法是网络上最常见的用于传输8bit字节代码的编码方式之一，Base64编码可用于在HTTP环境下传递较长的标识信息。例如，在JAVAPERSISTENCE系统HIBEMATE中，采用了Base64来将一个较长的唯一标识符编码为一个字符串，用作HTTP表单和HTTPGETURL中的参数。在其他应用程序中，也常常需要把二进制数据编码为适合放在URL（包括隐藏表单域）中的形式。此时，采用Base64编码不仅比较简短，同时也具有不可读性，即所编码的数据不会被人用肉眼所直接看到。

## MD5加密算法

MD5为计算机安全领域广泛使用的一种散列函数，用以提供消息的完整性保护。对[MD5加密算法](http://www.cksis.com/blog/316-md5jiamiyingyong.html)简要的叙述可以为：MD5以512位分组来处理输入的信息，且每一分组又被划分为16个32位子分组，经过了一系列的处理后，算法的输出由四个32位分组组成，将这四个32位分组级联后将生成—个128位散列值。

MD5被广泛用于各种软件的密码认证和钥匙识别上。MD5用的是哈希函数，它的典型应用是对一段信息产生信息摘要，以防止被篡改。MD5的典型应用是对一段Message产生fingerprin指纹，以防止被“篡改”。如果再有—个第三方的认证机构，用MD5还可以防止文件作者的“抵赖”，这就是所谓的[数字签名](http://www.cksis.com/blog/869-jiamiyushuziqianming.html)应用。MD5还广泛用于[**操作系统**](http://lib.csdn.net/base/operatingsystem)的登陆认证上，如UNIX、各类BSD系统登录密码、数字签名等诸多方。

## SHA1加密算法

SHA1是和MD5一样流行的消息摘要算法。SHA加密算法模仿MD4加密算法。SHA1设计为和数字签名算法（ＤＳＡ）一起使用。

SHA1主要适用于数字签名标准里面定义的数字签名算法。对于长度小于2“64位的消息，SHA1会产生一个160位的消息摘要。当接收到消息的时候，这个消息摘要可以用来验证数据的完整性。在传输的过程中，数据很可能会发生变化，那么这时候就会产生不同的消息摘要。SHA1不可以从消息摘要中复原信息，而两个不同的消息不会产生同样的消息摘要。这样，SHA1就可以验证数据的完整性，所以说SHA1是为了保证文件完整性的技术。

SHA1加密算法可以采用不超过264位的数据输入，并产生一个160位的摘要。输入被划分为512位的块，并单独处理。160位[缓冲器](https://www.baidu.com/s?wd=%E7%BC%93%E5%86%B2%E5%99%A8&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd)用来保存散列函数的中间和最后结果。缓冲器可以由5个32位寄存器（A、B、C、D和E）来表示。SHA1是一种比MD5的安全性强的算法，理论上，凡是采取“消息摘要”方式的数字验证算法都是有“碰撞”的——也就是两个不同的东西算出的消息摘要相同，互通作弊图就是如此。但是安全性高的算法要找到指定数据的“碰撞”很困难，而利用公式来计算“碰撞”就更困难一目前为止通用安全算法中仅有MD5被破解。

加密算法是[密码](http://www.cksis.com/blog/820-wangluomimaanquanjiqiao.html)技术的核心，以上这些加密算法是常用的加密算法，而这些算法有些已经遭到破译，有些安全度不高，有些强度不明，有些待进—步分析，有些需要深入研究，而神秘的加密算法世界，又会有新的成员加入，期待更安全的算法诞生。