# 加密

## 常用的加密术语

### 基础术语

明文：待加密信息。

密文：经过加密后的明文。

加密：明文转为密文的过程。

加密算法：明文转为密文的转换算法。

加密密钥：通过加密算法进行加密操作用的密钥。

解密：将密文转为明文的过程。

解密算法：密文转为明文的算法。

解密密钥：通过解密算法进行解密操作用的密钥。

### 常用术语

密码分析：截获密文者试图通过分析截获的密文从而推断出原来的明文或密钥的过程。

主动攻击：攻击者非法入侵密码系统，采用伪造、修改、删除等手段向系统注入假消息进行欺骗。（对密文具有破坏作用）

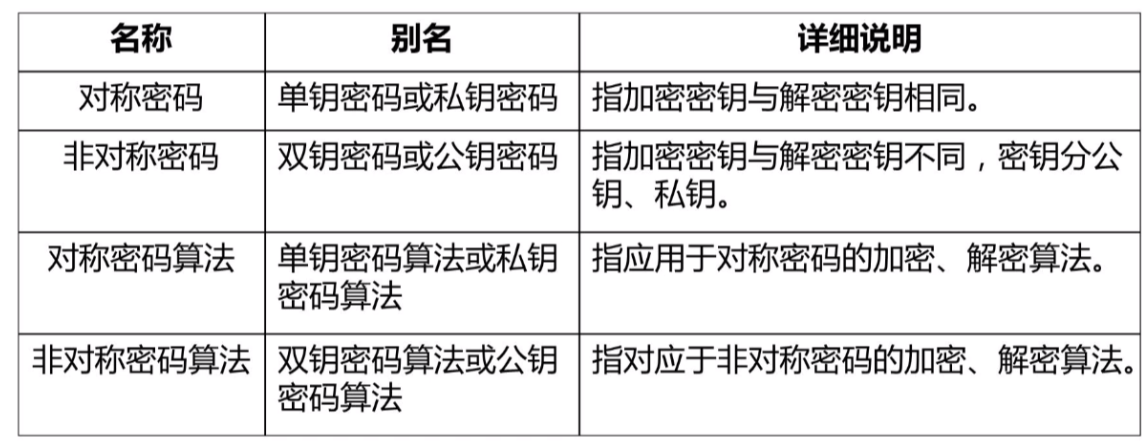
被动攻击：对一个保密系统采取截获密文并对其进行分析和攻击。（对密文没有破坏作用）

密码体制：由明文空间、密文空间、密钥空间、加密算法和解密算法五部分构成。

### 常用的术语

# 密码分类

## 密码体制分类



## 明文的处理方法分类

### 分组密码

分组密码：指加密时将名为分成固定长度的组，用同一密钥和算法对每一块加密，输出也是固定长度的密文。多用于网络加密。

### 流密码

流密码：也称序列密码。指加密时每次加密一位或者一个字节明文。

## 散列函数

散列函数用来验证数据的完整性

特点:长度不受限制，哈希值容易计算

常见的算法：

消息摘要算法MD5等

SHA——安全散列算法

MAC——消息认证码算法

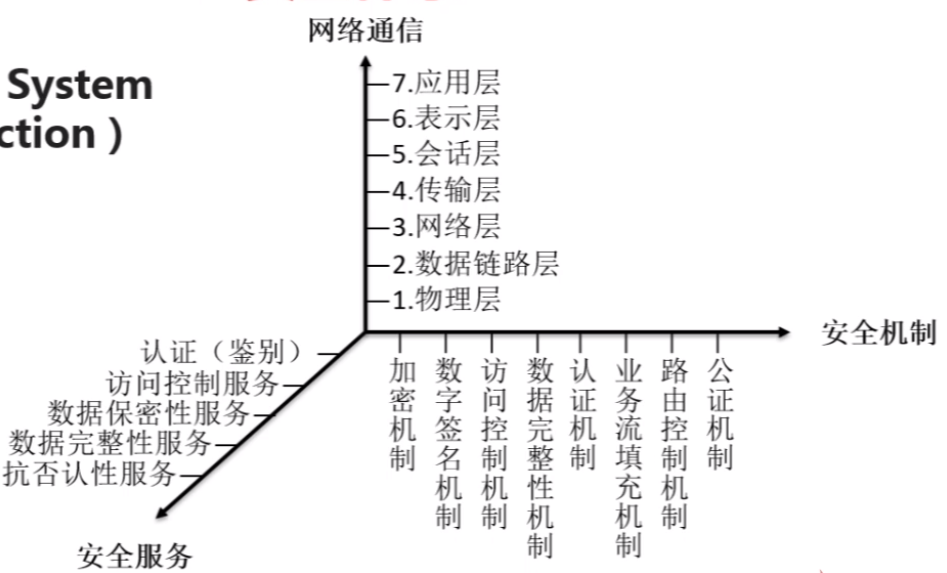
## 数字签名

主要是针对以数字的形式存储的消息进行的处理

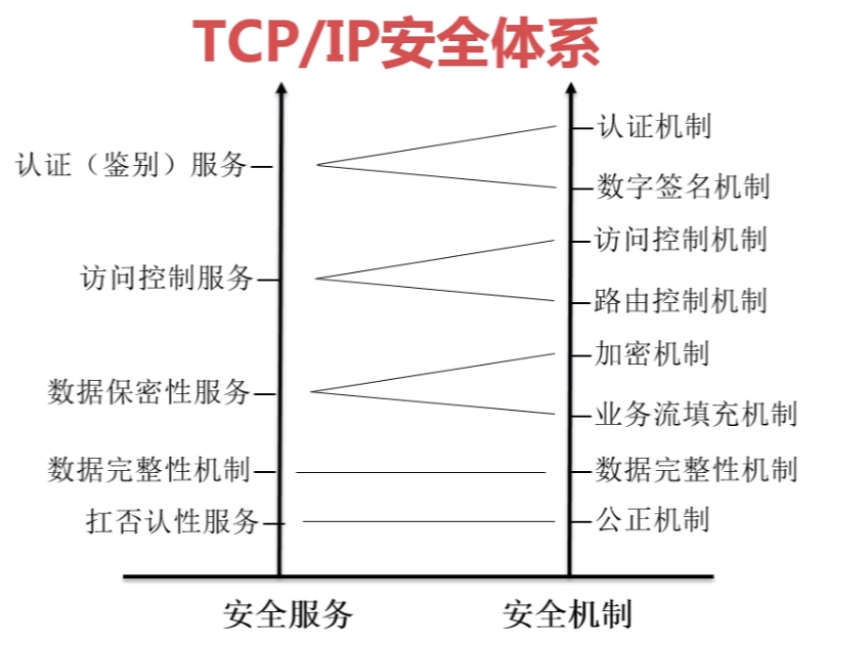
## 对称密码

# 安全体系

## OSI（Open System Interconnection）安全体系



## TCP/IP安全体系



# java安全

## java安全组成

JCA (Java Cryptography Architecture)：java 加密体系结构

JCE(Java Cryptography Extension)：java加密拓展包，

JSSE(Java Secure Socket Extension)：java 安全套接字，JSSE提供基于SSL的加密功能，主要用于网络传输。

JAAS(Java Authentication and Authentication Service)：java鉴别和安全服务

## 相关java包、类

### java.security

-消息摘要

### javax.crypto

-安全消息摘要，消息认证（鉴别）码

### java.net.ssl

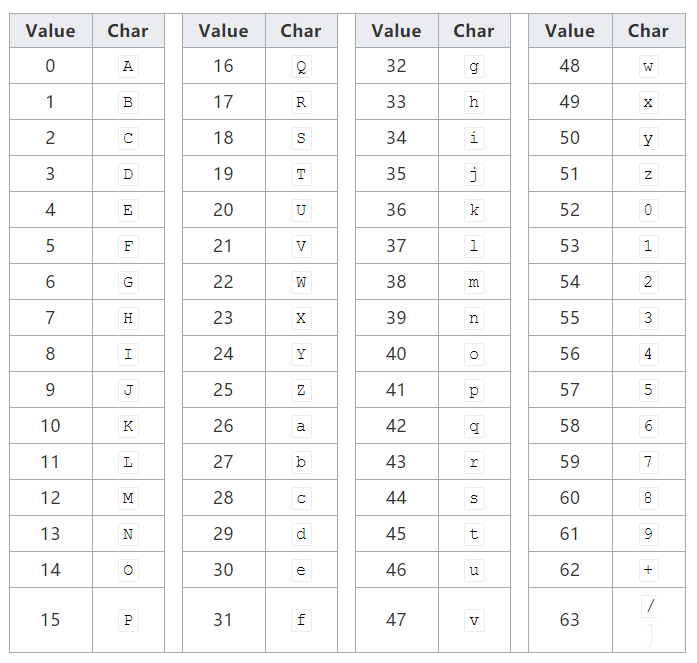
-安全套接字

# 单向加密算法

## Base64

### Base64

B首先，Base64是一种编码算法。为什么叫做Base64呢?因为这种算法只支持64个可打印字符。



### 为什么要采用BASE64

ASCII码的范围是0-127，其中0-31和127这33个字符属于控制字符（Control characters)。剩下32-126这95个字符属于可打印字符（Printable characters)，包含数字、大小写字母、常用符号。

控制字符



可打印字符

早期的一些传输协议，例如传输邮件的SMTP协议，只能传输可打印的ASCII字符。

导致原本8Bit 字节码（范围0-255)超过了可用的范围。

如当邮件传输图片资源的时候，某一个Byte 值是10111011B，对应十进制187不属于ASCII码范围，因此无法被传输。

Base64可以把原本ASCII码的控制字符甚至ASCII码之外的字符都转换成可打印的6bit字符。

既然单一字符的位数有限，我们可以增加字符的数量。8和6的最小公倍数是24，这就意味着我们可以用4个Base64字符来表示3个传统的8bit字符。



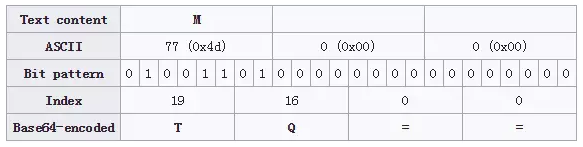
像表格中这样，8Bit X3的字符串可以每6个bit分成一组，每一组bit对应一个+进制的index，每一个index值又对应了Base64的字符。最终，8bit的字符串[Man]编码成了Base64的[TWFu]。

相比8bit的字符编码，Base64编码会多占用三分之一的字节长度，以此为代价实现了更好的兼容性。

那么如果原本的8Bit字符串长度不是3的倍数，比如长度是13，多余的那一个字符怎么编码呢？

很好解决，多余的8bit中，前6bit作一个Base64字符，剩下的2bit作为第二个Base64字符的前两位，后面的四位补0即可。

至于第三第四个Base64字符，没有匹配的8bit字符，则使用[=]字符填充（Padding)。如下图把8bit的[M]编码成Base64的[TQ==]。



我们知道在计算机中的字节共有256个组合，对应就是ascii码，而ascii码的128～255之间的值是不可见字符。而在网络上交换数据时，比如说从A地传到B地，往往要经过多个路由设备，由于不同的设备对字符的处理方式有一些不同，这样那些不可见字符就有可能被处理错误，这是不利于传输的。所以就先把数据先做一个Base64编码，统统变成可见字符，这样出错的可能性就大降低了。

### BASE64解码

解码也是同样道理，6bit X4的字符串可以每8bit分成一组，共3组。每一组转换成一个8位的Byte字节。

### BASE64特点

1.标准base64只有64个字符（英文大小写、数字和+、/）以及用作后缀等号；  
 2.base64是把3个字节变成4个可打印字符，所以base64编码后的字符串一定能被4整除（不算用作后缀的等号）；  
 3.等号一定用作后缀，且数目一定是0个、1个或2个。这是因为如果原文长度不能被3整除，base64要在后面添加\0凑齐3n位。为了正确还原，添加了几个\0就加上几个等号。显然添加等号的数目只能是0、1或2；  
 4.严格来说base64不能算是一种加密，只能说是编码转换。使用base64的初衷。是为了方便把含有不可见字符串的信息用可见字符串表示出来，以便复制粘贴；

### BASE64使用场景

### java实现base64

|  |
| --- |
| @Test **public void** contextLoads() **throws** IOException {  BASE64Encoder base64Encoder = **new** BASE64Encoder();  String password = base64Encoder.encode(**"password"**.getBytes());  System.***out***.println(password);  BASE64Decoder base64Decoder = **new** BASE64Decoder();  password = **new** String(base64Decoder.decodeBuffer(password));  System.***out***.println(password); } |

结果：

cGFzc3dvcmQ=

password

# 对称加密算法

## 对称加密算法

DES

3DEAS

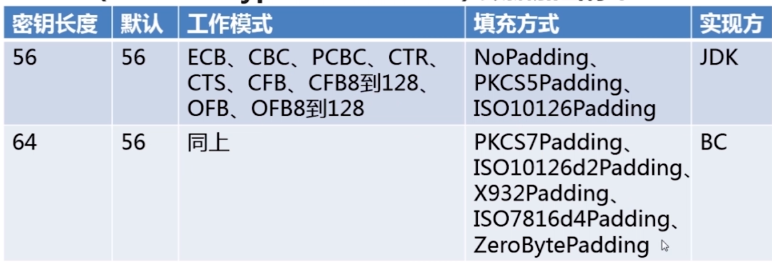
ASE

PBE

IDEA

## DES(Data Encryption Standard)

DES(Data Encryption Standard)数据加密标准



# 信息摘要算法

## 信息摘 要算法

MD(Message Digest)

SHA(Secure Hash Algorithm)

MAC(Message Authentication Code)

验证数据完整性

数字签名核心算法