Crypto 方向学习笔记

密码学基本知识：

密码学是研究如何隐密地传递信息的学科，可分为**古典密码学**和**现代密码学**。

古典密码学主要关注信息的保密书写和传递，以及与其相对应的破译方法。
现代密码学不只关注信息保密问题，还同时涉及信息完整性验证（消息验证码）、信息发布的不可抵赖性（数字签名）、以及在分布式计算中产生的来源于内部和外部的攻击的所有信息安全问题。

古典密码：替换密码 代换密码
现代密码学：对称加密 非对称加密 哈希函数 数字签名

注：密钥 k，明文 m，密文 c
攻击类型的分类：

攻击类型	说明
唯密文攻击	只拥有密文
已知明文攻击	拥有密文与对应的明文
选择明文攻击	拥有加密权限，能够对明文加密后获得相应密文
选择密文攻击	拥有解密权限，能够对密文解密后获得相应明文

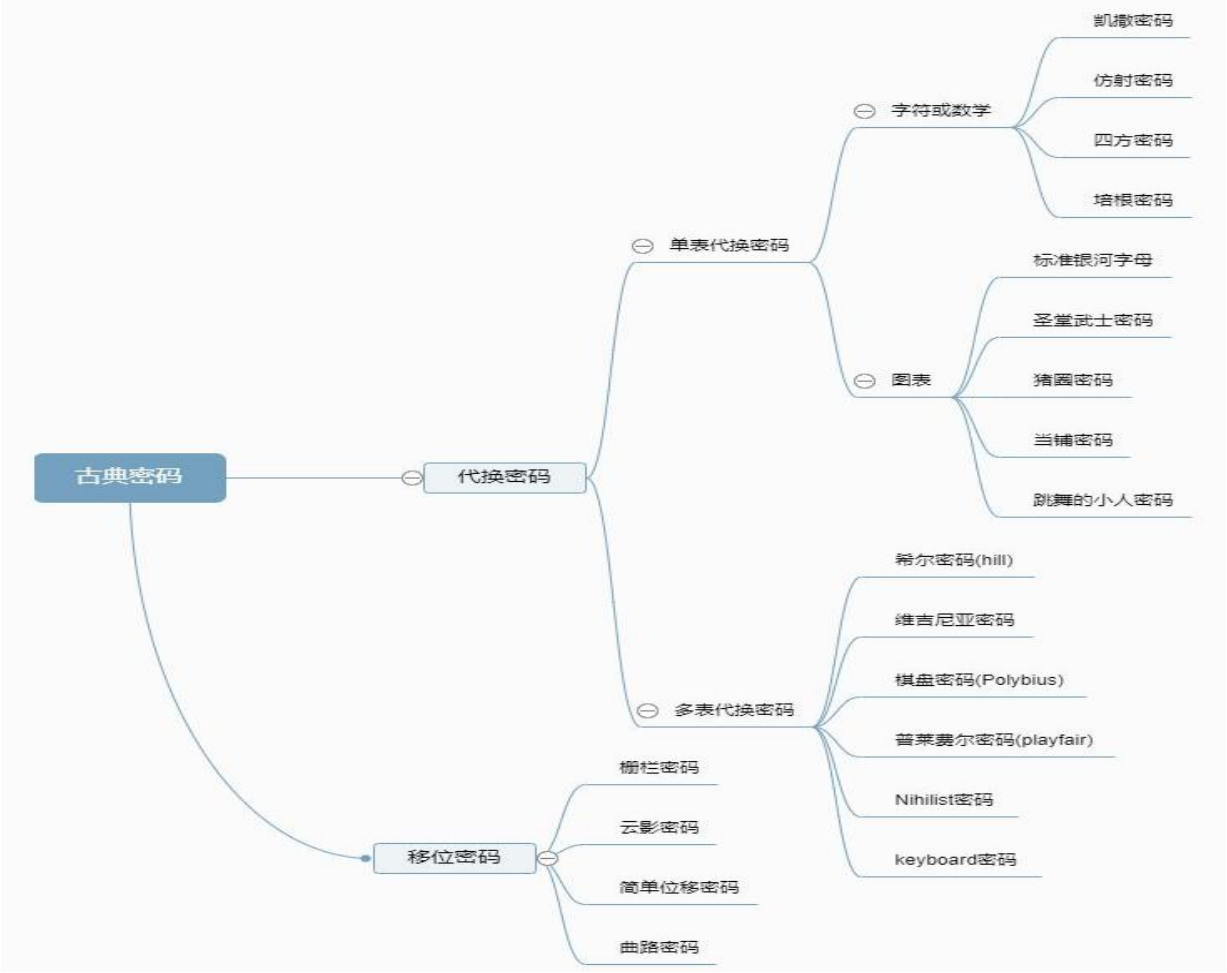
一、古典密码：

古典密码是许多现代密码算法的基石，在形式上分为**换位密码**和**替代密码**两类，其中替代密码又可分为**单表替代**和**多表替代**。

需要具备的能力：
扎实的数学功底，还要有识别、攻击、分析、编程、学习和跨领域能力，能够掌握 Reverse、PWN、Web 等其他领域的基本知识，Crypto 常见的密码学算法及其攻击类型。

以数论+Python 为基础，学习各种密码类型及其实现

古典密码的分类：



换位密码：
根据一定的规则重新排列明文，以便打破明文的结构特性

栅栏密码（换位密码）：
所谓栅栏密码，就是将要加密的明文分为 k 个一组（怎么分？V 型、N 型），然后取每组的第一个字符依次连接，拼接而成的字符串就是密文

替代密码：
分为**单表替代**和**多表替代**
密钥就是其替换表
最有效的攻击方式：词频分析

凯撒密码（单表替换）：
凯撒密码是历史上已知最早的密码之一。
凯撒密码中的每个字母在字母表中“移动”了一定的位置。例如，如果密码为 1，则 A 将替换为 B，B 将替换为 C，依此类推。
相传最早是凯撒大帝用来和将军进行秘密交流时所用的加密方法。

> 举例
>
> 我们使用密钥1来加密 `defend the east wall of the castle`

首先写出密钥为1时明文和密文的对照表：

明文	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
密文	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A

对要加密的明文逐个查表替换：D替换为E、E替换为F、F替换为G、E替换为F... 依次类推，直到整段明文都被加密。

```
plaintext:  defend the east wall of the castle
ciphertext: efgfoe uif fbtu xbmm pg uif dbtumf
```

培根密码（替换密码 隐写密码）：
培根密码，又名倍康尼密码，是由法兰西斯·培根发明的一种隐写术。
它的特殊之处在于：可以通过不明显的特征来隐藏密码信息， 比如大小写、正斜体等，只要两个不同的属性，密码即可隐藏。
培根密码实际上就是一种**替换密码**，一般有两种加密方式，解密时相互对应即可

维吉尼亚密码（多表替换）：
维吉尼亚密码是使用一系列凯撒密码组成密码字母表的加密算法，属于**多表密码**的一种简单形式
原理：
1. **加密过程**:明文字母 p 对应的列和秘钥字母 k 对应的行的交叉点就是加密字母后的密文字母
2. **解密过程**:在密钥字母 k 对应的行找到对应密文字母，则该密文字母 对应的列的字母就是明文字母

维吉尼亚密码盘：

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

波利比奥斯密码（本质上与替换密码相同）：
每个明文字符均被加密为两个密文字符。该密码的安全性较低，当明文长度大于 100 字符时，甚至可以使用手动进行破解。

> **举例**
>
> **使用如下的密钥矩阵加密 `defend the east wall of the castle`**
>
> ```
>    A B C D E
>    - - - - -
> A| p h q g m
> B| e a y l n
> C| o f d x k
> D| r c v s z
> E| w b u t i
> ```

加密过程：

```
明文： d e f e n d  t h e  e a s t  w a l l  o f  t h e  c a s t l e
密文： CCBACBBABECC EDABBA BABBDDED EABBBDBD CACB EDABBA DBBBDDEDBDBA
```

加密过程显而易见，每个明文字母均被替换为当前行首和列首的两个密文字符，如 `d->CC`

> **举例**
>
> **如果像上文一样生成密钥矩阵，则我们需要记住25个字母。为了简化记忆，我们可以选取一个密钥，然后将其余字母按字母表顺序排列。如密钥为 `zebra`,则可以将密钥扩展为：`zebracdfghiklmnopqstuvwxy`**

依次取密钥中五个字符为一行，生成密钥矩阵即可。

希尔密码（有向图密码）：
希尔密码（Hill Cipher）是运用**基本矩阵论原理**的替换密码。
每个字母当作 26 进制数字：A=0，B=1，C=2...一串字母当成 n 维向量，跟一个 n×n 的矩阵相乘，再将得出的结果 MOD26。

其它一些古典密码类型：
换位密码：
密码棒密码：
密码棒是个可使的传递讯息字母顺序改变的工具，由一条加工过、且有夹带讯息的皮革绕在一个木棒所组成。
在古希腊，文书记载着斯巴达人用此于军事上的讯息传递。

曲路密码：

曲路密码是一种置换密码，其中密钥是从明文创建的块中读取密文时要遵循的路径，该密钥需双方事先约定好（曲路路径）。

> 例题
>
> 明文：The quick brown fox jumps over the lazy dog
>
> 填入5行7列表中（事先约定）

首先，我们在明文的合理大小的块中写明文。键的一部分是此网格的大小，因此需要在开始之前决定网格中的多个列或行数（5行7列）。

将明文写入网格后，即可使用已分配的路径读取密文。

T	h	e	q	u	i	c
k	b	r	o	w	n	f
o	x	j	u	m	p	s
o	v	e	r	t	h	e
l	a	z	y	d	o	g

则加密之后的密文：

gesfc inpho dtmwu qoury zejre hbxva lookT

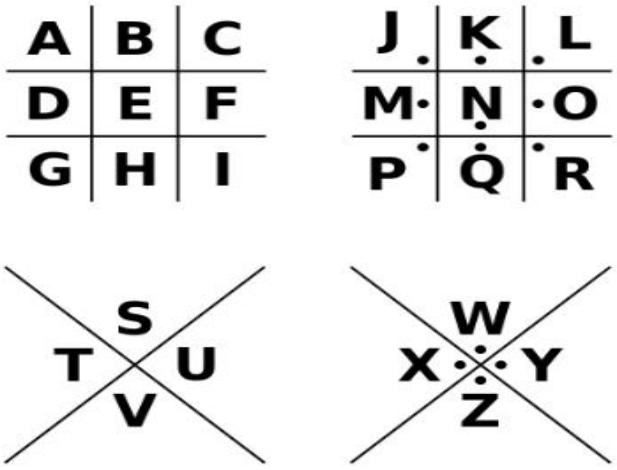
替换密码：

猪圈密码：

猪圈密码，是一种以格子为基础的简单替代式密码。即使使用符号，也不会影响密码分析，亦可用在其它替代式的方法。

优点：简单，方便，容易书写，适合书面上的密码通讯，并且好记。

缺点：″太出名″密码最怕的就是太出名，一但出名它就会毫无秘密可言，知道的人就知道，不知道的人就不知道。



仿射密码(单字母替换密码的特例）：

该密码不如简单替换密码安全，因为它不仅容易受到针对替代密码的所有攻击方法，而且还存在其他缺点。

其**主要缺点**来自以下方面：如果密码分析者可以发现（通过频率分析，暴力破解，猜测或其他方式）两个密文字符对应的明文，则可以通过联立方程组来求解密钥。

分组密码：

分组莫斯密码：

分组摩斯密码首先将明文转换为摩斯编码，然后将固定长度的摩斯码加密回字母。这意味着明文字母和密文字母已经不是一一对应的关系，比常见的替换密码更加安全（但就现在而言仍有方法对其进行破解）。

首先将明文转换为摩斯码，字符之间使用 `x` 分割、单词之间使用 `xx` 分割。

```
明文：defend the east
摩斯：-..x.x..-.x.x-.x-..xx-x....x.xx.x.-x...x-x
```

现在根据密钥生成加密表：

```
R O U N D T A B L E C F G H I J K M P Q S V W X Y Z
. . . . . . . . . - - - - - - - - - x x x x x x x x
. . . - - - x x x . . . - - - x x x . . . - - - x x
. - x . - x . - x . - x . - x . - x . - x . - x . -
```

我们将摩斯码对应到加密表中，如前三个摩斯码为 `-..` 转换到密文为 `E`，接下来的三个摩斯码为 `x.x` 转为到密文为 `S`，依次类推得到所有密文 `ESOAVVLJRSSTRX`。

跨棋盘密码：

跨棋盘密码是一种替换密码，当这种密码在结合其他加密方式，加密效果会更好。

二、常见编码：

编码：用预先规定的方法将对象转换为数码

编码与加密的区别：

1.编码使用公开的方案将数据转换成另一种格式，这样就可以很容易地将其反转；加密将数据转换为另一种格式，这样只有特定的个人才能反转转换。

2.编码是为了维护数据可用性，并使用公开可用的方案；加密是为了维护数据机密性，因此反向转换(密钥)的能力仅限于某些人。

1.Base 家族：

Base64：

①一般情况下密文尾部都会有两个等号，明文很少的时候则没有。

②Base64 编码要求把 3 个 8 位字节（38=24）转化为 4 个 6 位的字节（46=24），之后在 6 位的前面补两个 0，形成 8 位一个字节的形式。如果剩下的字符不足 3 个字节，则用 0 填充，输出字符使用‘=’，因此编码后输出的文本末尾可能会出现 1 或 2 个‘=’。

Base64编码表

码值	字符	码值	字符	码值	字符	码值	字符	码值	字符	码值	字符	码值	字符	码值	字符
0	A	8	I	16	Q	24	Y	32	g	40	o	48	w	56	4
1	B	9	J	17	R	25	Z	33	h	41	p	49	x	57	5
2	C	10	K	18	S	26	a	34	i	42	q	50	y	58	6
3	D	11	L	19	T	27	b	35	j	43	r	51	z	59	7
4	E	12	M	20	U	28	c	36	k	44	s	52	0	60	8
5	F	13	N	21	V	29	d	37	l	45	t	53	1	61	9
6	G	14	O	22	W	30	e	38	m	46	u	54	2	62	+
7	H	15	P	23	X	31	f	39	n	47	v	55	3	63	/

Base64 使用注意问题：

①Base64 和 URL 传参问题

原因：URL 编码器会把标准 Base64 中的“/”和“+”字符变为形如“%XX”的形式，而这些“%”号在存入数据库时还需要再进行转换（ANSI SQL 中已将“%”号用作通配符）

解决方法：采用一种用于 URL 的改进 Base64 编码，它在末尾填充’=’号，并将标准 Base64 中的“+”和“/”分别改成了“-”和“_”（免去了在 URL 编解码和数据库存储时所要作的转换，避免了编码信息长度在此过程中的增加，并统一了数据库、表单等处对象标识符的格式）

②Base64 转换后比原有的字符串长 1/3

Base64 要求把每三个 8Bit 的字节转换为四个 6Bit 的字节（38 = 46 = 24），然后把 6Bit 再添两位高位 0，组成四个 8Bit 的字节，也就是说，转换后的字符串理论上将要比原来的长 1/3。

Base58（和十进制、十六进制一样，但更节省空间）：

Base58 是用于比特币（Bitcoin）中使用的一种独特的编码方式，主要用于产生 Bitcoin 的钱包地址。

相比 Base64，Base58 不使用数字"0"，字母大写"O"，字母大写"I"，和字母小写"l"，以及"+“和”/"符号。

Value	Character	Value	Character	Value	Character	Value	Character
0	1	1	2	2	3	3	4
4	5	5	6	6	7	7	8
8	9	9	A	10	B	11	C
12	D	13	E	14	F	15	G
16	H	17	J	18	K	19	L
20	M	21	N	22	P	23	Q
24	R	25	S	26	T	27	U
28	V	29	W	30	X	31	Y
32	Z	33	a	34	b	35	c
36	d	37	e	38	f	39	g
40	h	41	i	42	j	43	k
44	m	45	n	46	o	47	p
48	q	49	r	50	s	51	t
52	u	53	v	54	w	55	x
56	y	57	z				

Base32：

特点是明文超过十个后面就会有很多等号。

Base32 编码使用 32 个可打印字符(字母 A-Z、数字 2-7)对任意字节数据进行编码的方案，编码后的字符串不区分大小写并排除了容易混淆的字符串。

由于数据的二进制传输是按照 8 比特一组进行的，因此 Base32 按照 5 比特切分的二进制数据，所以数据必须是 40(5 和 8 的最小公倍数)比特的倍数。

Base32编码表

值	符号	值	符号	值	符号	值	符号
0	A	8	I	16	Q	24	Y
1	B	9	J	17	R	25	Z
2	C	10	K	18	S	26	2
3	D	11	L	19	T	27	3
4	E	12	M	20	U	28	4
5	F	13	N	21	V	29	5
6	G	14	O	22	W	30	6
7	H	15	P	23	X	31	7
填充	=						

2.Morse 电码：

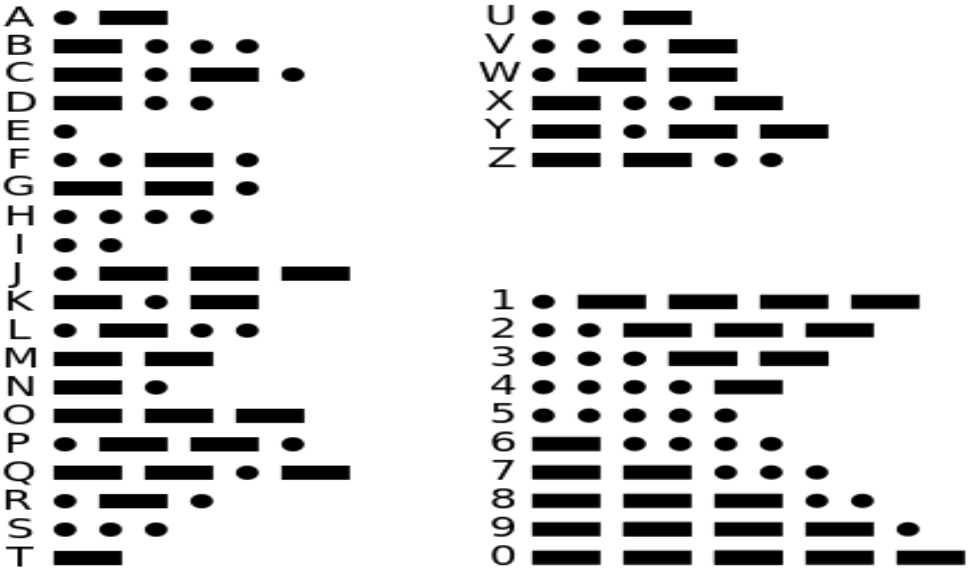
①一点的长度是一个单位

②一划是三个单位

③在一个字母中点划之间的间隔是一点

④两个字母之间的间隔是三点（一划）

⑤两个单词之间的间隔是七点



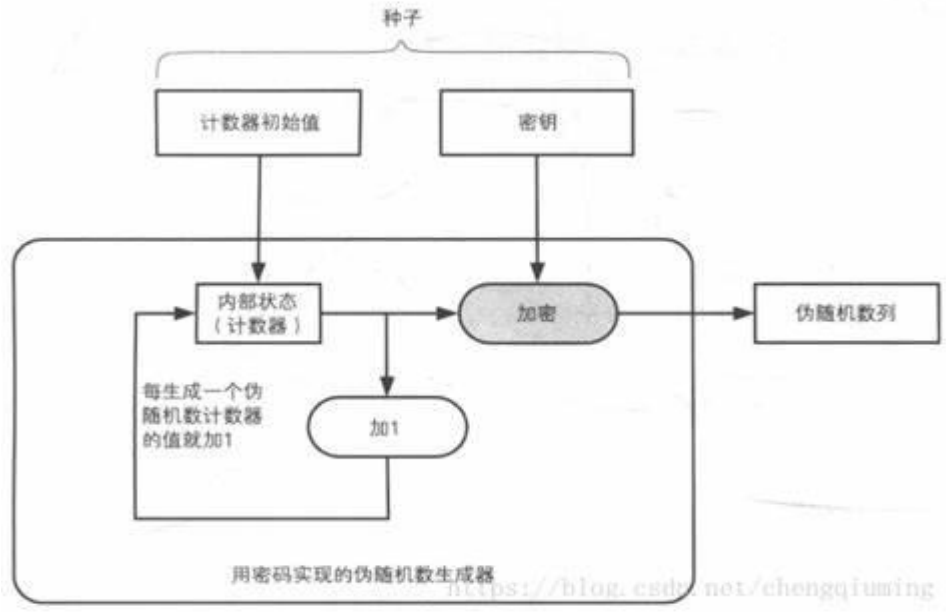
3.与佛论禅
特点：就是看不懂的佛语
4.百家姓暗号
5.在代码混淆中可能用到的 jother 编码和 JSFuck

三、对称密码：

对称密码的分类
1.序列密码（流密码）
流密码是一种对称密钥密码，其中明文数字与伪随机密码数字流相结合。在流密码中，每个明文数字一次用密钥流的相应数字加密，以给出密文流的数字。由于每个数字的加密取决于密码的当前状态，因此也称为状态密码。在实践中，一个数字通常是一个位，组合操作是异或(XOR)。

例如：OTP（one-time pad）一次性密码本
密钥长度与明文长度一样
密钥：10101010101011110
明文：11110101010011101
密文：01011111111000011
流密码特点：
1.流密码一般逐字节或者逐比特处理信息。
2.流密码的密钥长度会与明文的长度相同。
3.流密码的密钥派生自一个较短的密钥，派生算法通常为一个伪随机数生成算法。

伪随机数生成器
一个不符合密码学的伪随机数生成器：
平方取中法：选择一个 m 位数 Ni 作为种子，做平方运算（记为 Ni+ 1 = （Ni * Ni）...），结果若不足 2m 个位，在前补 0。在这个数选中间 m 个位的数作为 Ni+1



2.分组密码

分组密码是一种加/解密算法，将输入的明文分组当做 一个整体处理，输出一个等长的密文分组。

DES 中的一种结构---Feistel 结构

Feistel 的优点：由于它是对称的密码结构，所以对信息的加密和解密的过程就极为相似，甚至完全一样。这就使得在实施的过程中，对编码量和线路传输的要求就减少了几乎一半。

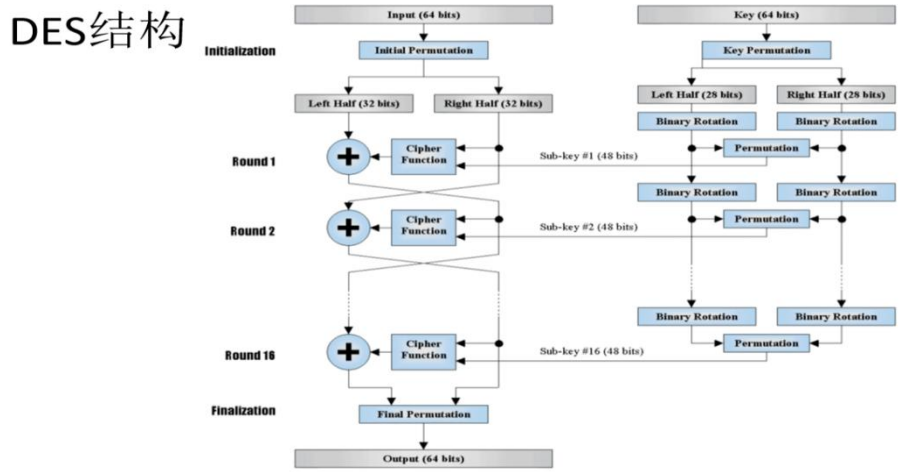
Feistel 密码

Feistel 建议使用乘积密码来增强密码的强度。
Feistel 建议交替使用代换和置换，增强密码的扩散和混淆性能

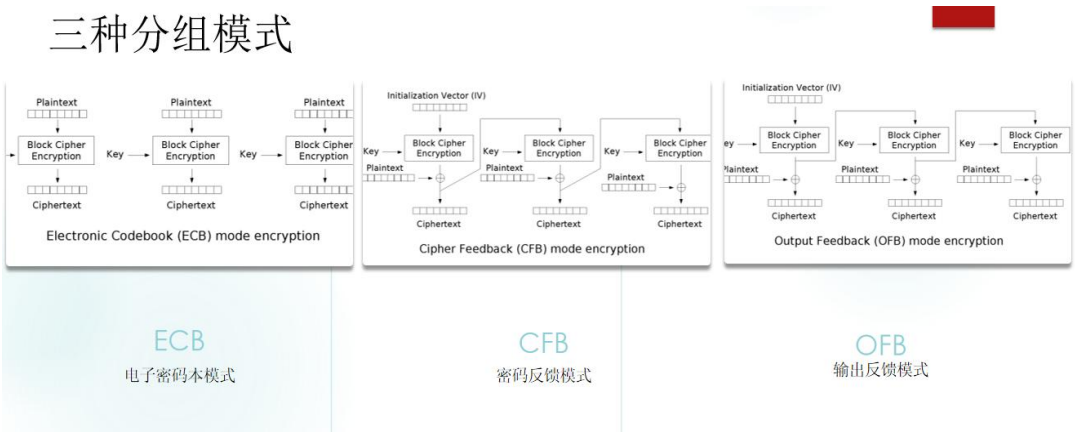
混淆：尽可能使密文和加密密钥之间的关系变得复杂，以阻止攻击者发现密钥。密钥和密文关系为非线性关系。
扩散：让每个明文数字尽可能地影响多个密文数字，使明文的统计特性消散在密文中。明文和密文间关系为非线性关系。

Feistel 密码结构

子密钥生成算法 子密钥生成算法越复杂，密码分析攻击就越困难。
轮函数 轮函数越复杂，抗击密码分析的攻击能力就越强。



3.分组方式



其他类型编码：

1.Base16

Base16 编码使用 16 个 ASCII 可打印字符（数字 0-9 和字母 A-F）对任意字节数据进行编码。

Base16 先获取输入字符串每个字节的二进制值（不足 8 比特在高位补 0），然后将其串联进来，再按照 4 比特一组进行切分，将每组二进制数分别转换成十进制，在下述表格中找到对应的编码串接起来就是 Base16 编码。

可以看到 8 比特数据按照 4 比特切分刚好是两组，所以 Base16 不可能用到填充符号“=”。

Base16 编码后的数据量是原数据的两倍：1000 比特数据需要 250 个字符（即 250*8=2000 比特）。换句话说：Base16 使用两个 ASCII 字符去编码原数据中的一个字节数据。

	a	s	d	f
ASCII	97	115	100	102
hex	0x61	0x73	0x64	0x66
Base16	61	73	64	66

2.URL 编码

url 编码解码,又叫百分号编码，是统一资源定位(URL)编码方式。
URL 地址（常说网址）规定了常用地数字，字母可以直接使用，另外一批作为特殊用户字符也可以直接用（/,:@等），剩下的其它所有字符必须通过%xx 编码处理。

现在已经成为一种规范了,基本所有程序语言都有这种编码,如 js:有 encodeURI、encodeURIComponent，PHP 有 urlencode、urldecode 等。编码方法很简单，在该字节 ascii 码的的 16 进制字符前面加%. 如 空格字符，ascii 码是 32，对应 16 进制是’20’，那么 urlencode 编码结果是:%20

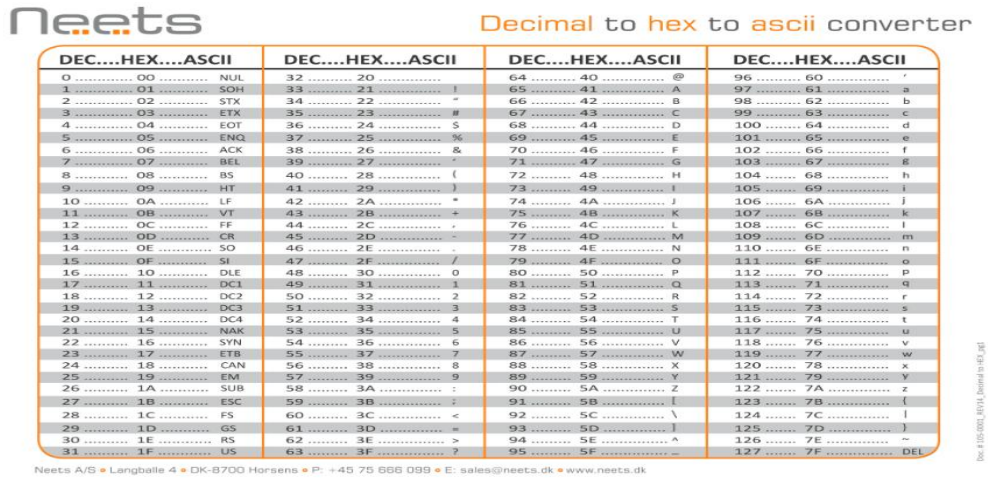
3.ASCII 码

ASCII 编码是美国信息交换标准代码，主要用于显示现代英语和其他西欧语言。它能表示 128 个字符，其中包括英文字符、阿拉伯数字、西文字符以及 32 个控制字符。

因为 1 位二进制数可以表示(21)=2 种状态：0、1；
而 2 位二进制数可以表示（22）=4 种状态：00、01、10、11；
依次类推，7 位二进制数可以表示(27)=128 种状态，每种状态都唯一地编为一个

7 位的二进制码，对应一个字符（或控制码），这些码可以排列成一个十进制序号 0～127。

ASCII字符集



4.Unicode 符号集

Unicode（它的来源是因为 ASCCLL 表示中文不够用）：它包含了世界上所有的符号，并且每一个符号都是独一无二的。**很多人都说 Unicode 编码，但其实 Unicode 是一个符号集（世界上所有符号的符号集），而不是一种新的编码方式。**

出现的问题：

1）出现了 unicode 的多种存储方式，也就是说有许多种不同的二进制格式，可以用来表示 unicode。

2）unicode 在很长一段时间内无法推广，直到互联网的出现。

原始字符	C		a		t	
原始ASCII码（十进制）	67		97		116	
ASCII码（二进制）	0 1 0 0 0 0	1 1	0 1 1 0	0 0 0 1	0 1	1 1 0 1 0 0
新的十进制数值	16	54		5		52
编码后的Xxencode字符	E	q		3		0

5.UTF-8

互联网的普及，强烈要求出现一种统一的编码方式。

UTF-8 就是在互联网上使用最广的一种 unicode 的实现方式。

注：UTF-8 是 Unicode 的实现方式之一

特点：它是一种变长的编码方式。它可以使用 1~4 个字节表示一个符号，根据不同的符号而变化字节长度。

UTF-8 的编码规则很简单，只有两条：

1）对于单字节的符号，字节的第一位设为 0，后面 7 位为这个符号的 unicode 码。因此对于英语字母，UTF-8 编码和 ASCII 码是相同的。

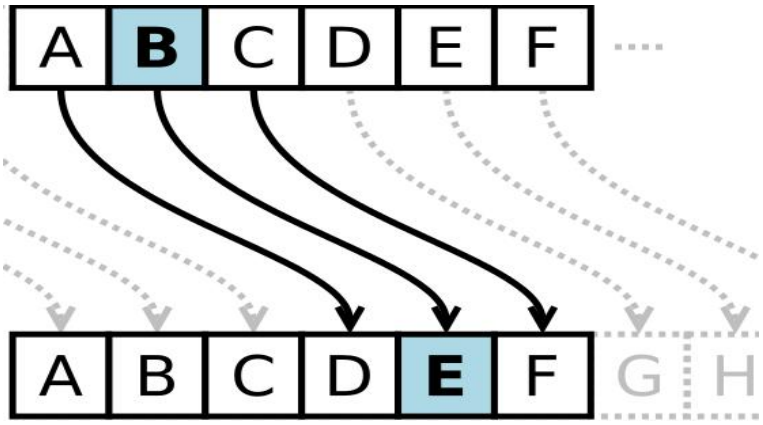
2）对于 n 字节的符号（n>1），第一个字节的前 n 位都设为 1，第 n+1 位设为 0，后面字节的前两位一律设为 10。剩下的没有提及的二进制位，全部为这个符号的 unicode 码。

6.GBK/GB2312/GB18030
GBK 和 GB2312 都是针对简体字的编码，只是 GB2312 只支持六千多个汉字的编码，而 GBK 支持 1 万多个汉字编码。
GB18030 是用于繁体字的编码。
汉字存储时都使用两个字节来储存。

注意：对于一个汉字占用几个字节的问题，用不同的编码占用的字节是不一样的。

四、古典密码、编码习题知识点总结：

1.题目后面有=就先猜测 base64 编码，用 base64 解码解密即可得到 flag
2.**恺撒密码特征**
明文中的所有字母都在字母表上向后（或向前）按照一个固定数目进行偏移后被替换成密文。例如，当偏移量是 3 的时候，所有的字母 A 将被替换成 D，B 变成 E，以此类推。

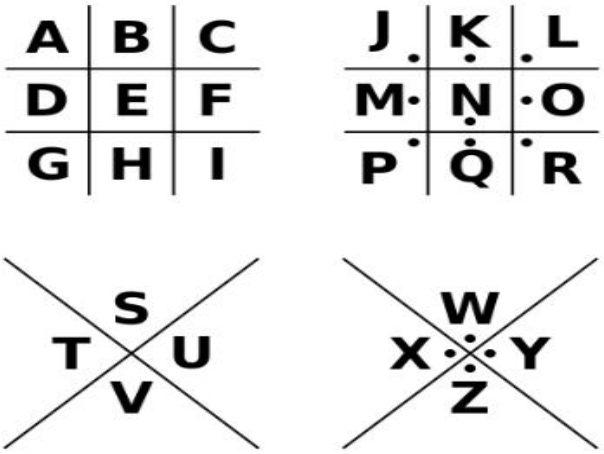


3.**篱笆墙很明显联想到栅栏密码，用栅栏在线解密即可**

4.**维吉尼亚密码特征**
维吉尼亚密码的原理与凯撒密码类似，其实是凯撒的一种强化和变形，通过使加密相同明文的秘钥不同，来掩盖字符的频率。
相同的明文 e，经过不同的字符加密之后变成了不同的密文，掩盖了明文字符 e 的字符频率。但也不是找不到字符频率，将用"h"字符加密的明文取出之后，就变成了普通的凯撒加密，这是可以通过字符频率分析来破解的。

5.**置换密码(又称易位密码)**：明文的字母保持相同，但顺序被打乱了。
代表：栅栏加密
代替密码：就是将明文的字符替换为密文中的另一种的字符，接收者只要对密文做反向替换就可以恢复出明文。
代表：恺撒加密

6. **猪圈密码**



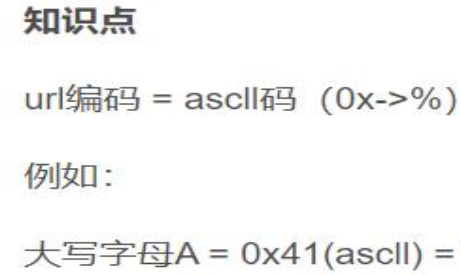
7. **低加密指数攻击**：低加密指数攻击 e 很小 n 很大又不好分解
所谓低加密指数指的就是 e 非常小的情况下，通常为 3。
这种题目通常有两种类型，**一种直接爆破，另外一种是低指数广播攻击。**


problem - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

#n:
0x52d483c27cd806550fbe0e37a61af2e7cf5e0efb723dfc81174c918a27627779b21fa3c851e9e94188eaee3d5cd6f752406a43fbecb53e80836ff1e185d3ccd7782ea846c2e91a7b0808986666e0bdadbfb7bdd65670a589a4d2478e9adcafe97c6ee23614bcb2ecc23580f4d2e3cc1ecfec25c50da4bc754dde6c8bfd8d1fc16956c74d8e9196046a01dc9f3024e11461c294f29d7421140732fedacac97b8fe50999117d27943c953f18c4ff4f8c258d839764078d4b6ef6e8591e0ff5563b31a39e6374d0d41c8c46921c25e5904a817ef8e39e5c9b71225a83269693e0b7e3218fc5e5a1e8412ba16e588b3d6ac536dce39fcdfce81eec79979ea6872793L
#e: 0x3
#c:0x10652cdfaa6b63f6d7bd1109da08181e500e5643f5b240a9024bfa84d5f2cac9310562978347bb232d63e7289283871efab83d84ff5a7b64a94a79d34cfbd4ef121723ba1f663e514f83f6f01492b4e13e1bb4296d96ea5a353d3bf2edd2f449c03c4a3e995237985a596908adc741f32365
so,how to get the message?

8. **url 编码**
%66%6c%61%67%7b%61%6e%64%20%31%3d%31%7d


知识点

url编码 = ascll码 （0x->%）

例如：

大写字母A = 0x41(ascll) = %41(url)

9. Rabbit 是一种**高速流密码**，它具有以下特征：

- 由26个英文大小写字母、=、+、/ 组成
- **以U2FsdGVkX1开头**
- 可能以=结尾

10. 字符串基本与 flag 的格式相对应，考虑**单字母替换加密**

五、Hash 算法：

哈希（Hash）算法，即散列函数。它是一种单向密码体制，即它是一个从明文到密文的不可逆的映射，只有加密过程，没有解密过程。同时，哈希函数可以将任意长度的输入经过变化以后得到固定长度的输出。哈希函数的这种单向特征和输出数据长度固定的特征使得它可以生成消息或者数据。

无论“电子签名、可信时间戳、哈希值校验、区块链等证据收集、固定和防篡改的技术手段”还是“通过电子取证存证平台认证”都离不开一种广义的加密算法（或者说一类算法）——Hash 算法。

Windows 校验 hash 码的自带工具

我们时常从网络上下载文件，却很少检验文件的完整性，试想如果下载了一个系统镜像，或是大型软件，得到的是不完整的文件，而表面上却看不出来，那么在安装过程中就会出错。

通常情况下载页都会提供checksum（校验码），格式常为MD5、SHA1 或SHA256，格式不同是因为采用了不同的算法，因此得到的校验码也是不同的。

如果你用的是Windows10，这个系统貌似是自带了一个MD5检测工具，能够应付大多数的场景。不过Windows中还有一个命令行工具提供更强大的功能——certutil。

下面我们要做几件事：

1. 打开要检测文件所在的目录
2. 在此处打开命令窗口
3. 输入命令
4. 等待结果并查看

哈希算法，又称为摘要算法，散列算法，杂凑算法，哈希算法。

其主要特点如下：

a.**输出长度固定**——可以把任意长度的原始数据计算输出为固定长度的哈希值；

b.结果不随机——同样的原始数据，使用同样的算法无论计算多少次，输出的哈希值都是一致的；

C.**输入变动敏感**——输入的原始数据哪怕出现一丁点儿变动，输出的哈希值均会出现巨大的变化；

d.**单向不可逆——无法通过一个特定的哈希值反向推导出原始数据。**

另外，一个优秀的哈希算法还应当具有一个特点：很难找到两个对应同一散列值的不同输入，即：碰撞避免。

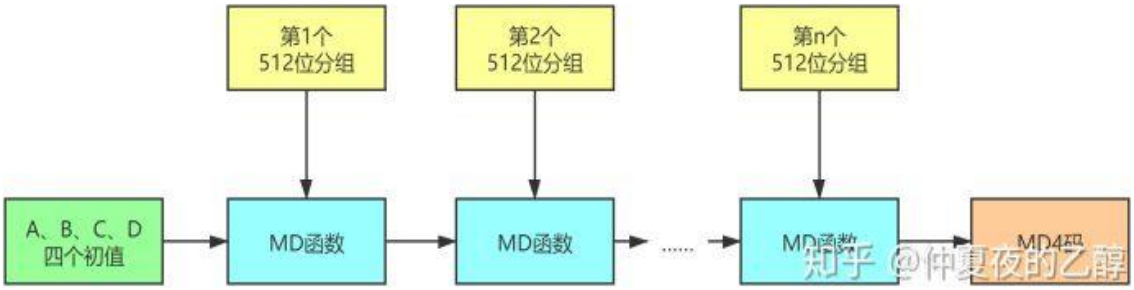
正是基于上述特点，哈希值被形象地称为**“数字指纹”**。

常用 hash 算法的介绍：

（1）MD4

MD4（RFC 1320）是 MIT 的 Ronald L. Rivest 在 1990 年设计的，MD 是 Message Digest（消息摘要） 的缩写。

它适用在 32 位字长的处理器上用高速软件实现——它是基于 32 位操作数的位操作来实现的。



（2）MD5

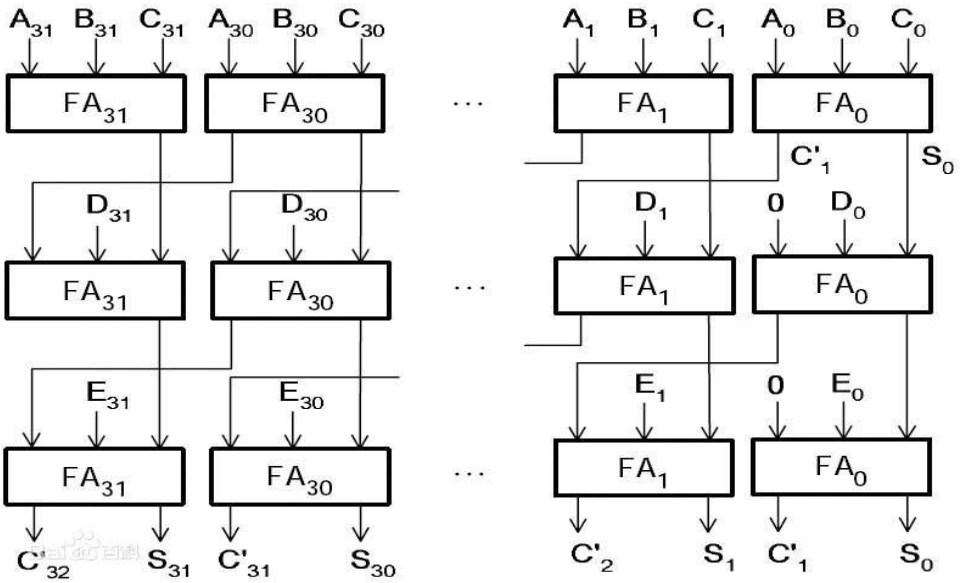
MD5（RFC 1321）是 MD4 的改进版本。它对输入仍以 512 位分组，其输出是 4 个 32 位字的级联，与 MD4 相同。

MD5 比 MD4 来得复杂，并且速度较之要慢一点，但更安全，在抗分析和抗差分方面表现更好。

（3）SHA-1 及其他

SHA1 是为同 DSA 一起使用的，它对长度小于 264 的输入，产生长度为 160bit 的散列值，因此抗穷举（brute-force）性更好。

SHA-1 设计时基于和 MD4 相同原理，并且模仿了该算法。

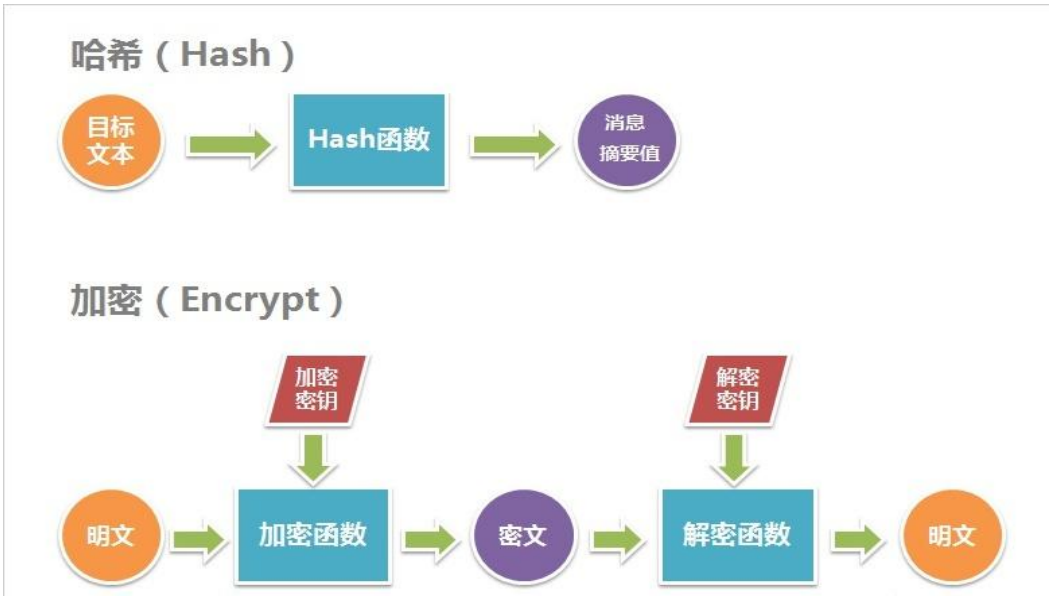


Hash 算法在信息安全方面的应用主要体现在以下的 3 个方面：

（1）文件校验

我们比较熟悉的校验算法有**奇偶校验**和 **CRC 校验**，这 2 种校验并没有抗数据篡改的能力，它们一定程度上能检测并纠正数据传输中的信道误码，但却不能防止对数据的恶意破坏。

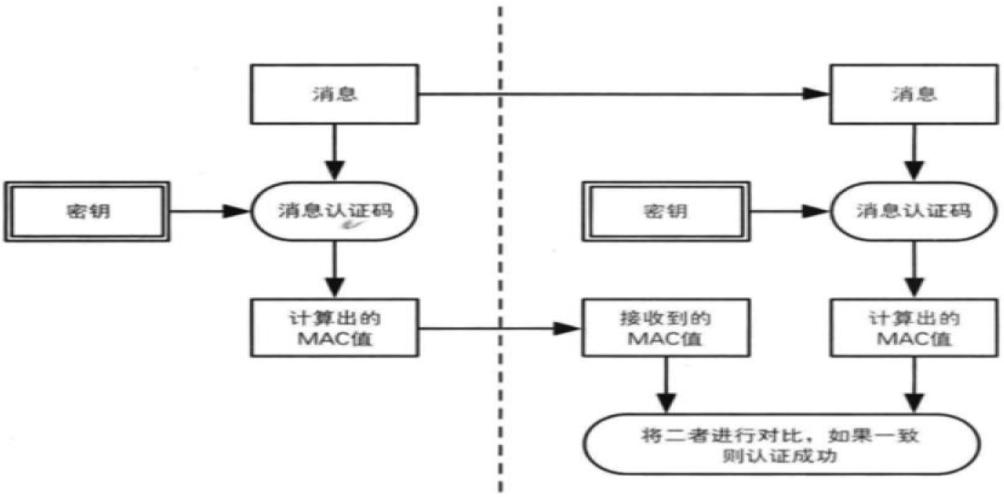
MD5 Hash 算法的“数字指纹”特性，使它成为目前应用最广泛的一种文件完整性校验和（Checksum）算法，不少 Unix 系统有提供计算 md5 checksum 的命令。



（2）数字签名

Hash 算法也是现代密码体系中的一个重要组成部分。由于非对称算法的运算速度较慢，所以在数字签名协议中，单向散列函数扮演了一个重要的角色。

对 Hash 值，又称“数字摘要”进行数字签名，在统计上可以认为与对文件本身进行数字签名是等效的。



（3）鉴权协议

如下的鉴权协议又被称作挑战--认证模式：在传输信道是可被侦听，但不可被篡改的情况下，这是一种简单而安全的方法。

文件 hash 值

MD5-Hash-文件的数字文摘通过 Hash 函数计算得到。不管文件长度如何，它的 Hash 函数计算结果是**一个固定长度的数字**。

与加密算法不同，这一个 Hash 算法是一个不可逆的单向函数。采用安全性高的 Hash 算法，如 MD5、SHA 时，两个不同的文件几乎不可能得到相同的 Hash 结果。因此，一旦文件被修改，就可检测出来。

```
D:\>certutil -hashfile Hash_test.txt MD5
MD5 的 Hash_test.txt 哈希:
96e79218965eb72c92a549dd5a330112
CertUtil: -hashfile 命令成功完成。

D:\>certutil -hashfile Hash_test.txt SHA1
SHA1 的 Hash_test.txt 哈希:
3d4f2bf07dc1be38b20cd6e46949a1071f9d0e3d
CertUtil: -hashfile 命令成功完成。

D:\>certutil -hashfile Hash_test.txt SHA256
SHA256 的 Hash_test.txt 哈希:
bcb15f821479b4d5772bd0ca866c00ad5f926e3580720659cc80d39c9d09802a
CertUtil: -hashfile 命令成功完成。
```

Hash 函数还有另外的含义。实际中的 Hash 函数是指把一个大范围映射到一个小范围。把大范围映射到一个小范围的目的往往是为了节省空间，使得数据容易保存。除此以外，Hash 函数往往应用于查找上。

所以，在考虑使用 Hash 函数之前，**需要明白它的几个限制：**
1. **Hash 的主要原理就是把大范围映射到小范围**；所以，你输入的实际值的个数必须和小范围相当或者比它更小。不然冲突就会很多。
2. **由于 Hash 逼近单向函数**，所以，你可以用它来对数据进行加密。
3. 不同的应用对 Hash 函数有着不同的要求；比如，用于加密的 Hash 函数主要考虑它和单项函数的差距，而用于查找的 Hash 函数主要考虑它映射到小范围的冲突率。

Hash 函数应用的主要对象是数组（比如，字符串），而其目标一般是一个 int 类型。

SHA-256：

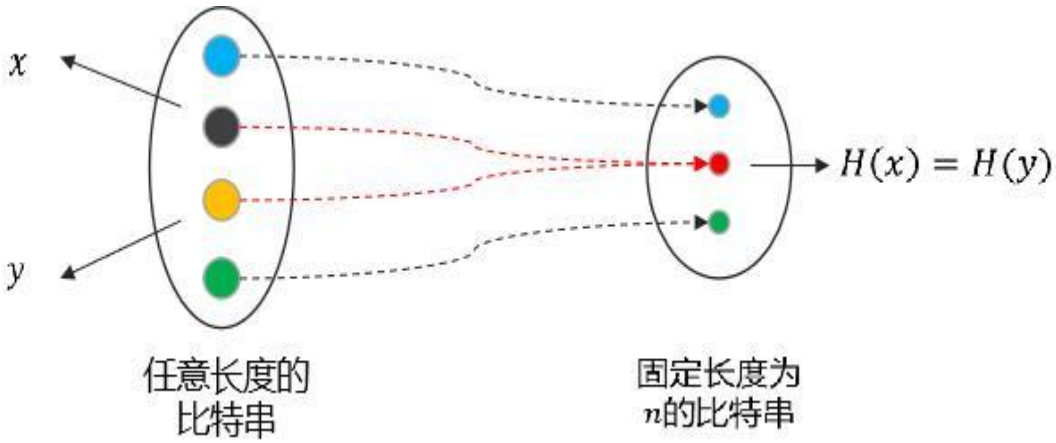
SHA-256 是由美国国家安全局(NSA)设计并由美国国家标准与技术研究院(NIST)在 2001 年发布的一种将任意长度的输入转化为一个 256 位的二进制数输出的哈希算法。

一、输出长度固定 & 单向不可逆

1. 人民日报电子版2021年1月1日第1版的Hash值（SHA-256）为：
b29a6cd5e65f0b568f4a57f710c90dc3979f953c0e7525b0164297e0ae54019b
（格式：PDF；大小：1.76 MB；来源：人民网；网址：
http://paper.people.com.cn/rmrb/html/2021-01/01/nbs.D110000renmrb_01.htm）

2. 记录片《科学星图》介绍视频的Hash值（SHA-256）为：
65bddba8e3997a772caeff7db81e24e0ac8bf8f790e13951b0ce967862203ff4
（格式：mp4；大小：17.6 MB；来源：人民网；网址：
http://v.people.cn/GB/28140/433969/index.html 在视频播放界面右键另存为）

从上述实例可以看出，无论输入的是什么，**输出的 Hash 值都是一个固定长度的字符串**。而“单向性”就更容易理解，给你一串十六进制的字符，你没可能反向推出《科学星图》的介绍视频（当然，“1”这类简单的输入除外，这个很容易通过枚举法找到原始输入）。



不同的原始数据，有可能产生同一个哈希值的(即哈希碰撞)。以 SHA-256 为例，其散列值为 256 位的二进制数(0 和 1)，因此 0 和 1 的组合存在 2^256 种。理论上使用枚举法以二进制数作为输入值，最多尝试 2^256+1 次必然出现一次同样的哈希值。

目前，针对 SHA - 256 的有效攻击方法尚未出现，**因此如需寻找 SHA - 256 的散列碰撞只能以枚举法随机测试。比特币挖矿即是使用的枚举法。**

Hash Attack：

它是一种单向密码体制，即它是一个从明文到密文的不可逆的映射,只有加密过程，没有解密过程。

字典攻击、暴力攻击、查表法、反向查表法、彩虹表……

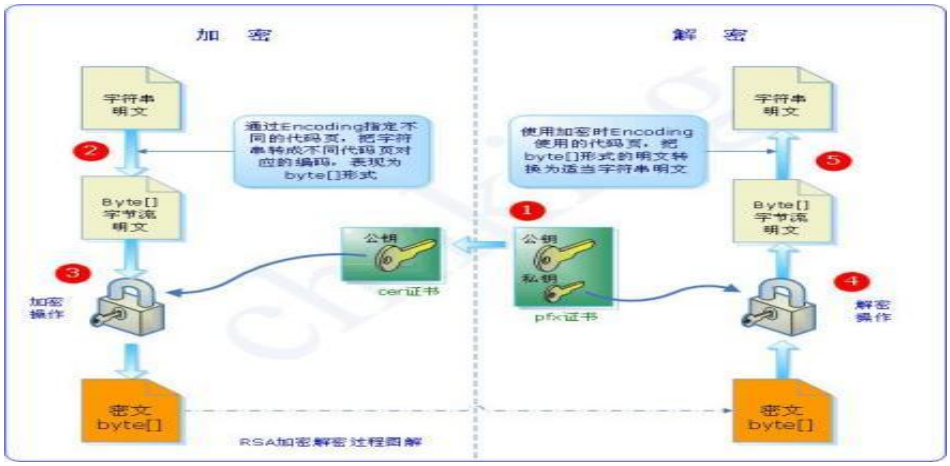
六、RSA：

RSA 公开密钥密码体制是一种使用不同的加密密钥与解密密钥，“由已知加密密钥推导出解密密钥在计算上是不可行的”密码体制。

它通常是先生成一对 RSA 密钥，其中之一是保密密钥，由用户保存；另一个为公开密钥，可对外公开，甚至可在网络服务器中注册。为提高保密强度，RSA 密钥至少为 500 位长，这就使加密的计算量很大。

为减少计算量，在传送信息时，**常采用传统加密方法与公开密钥加密方法相结合的方式**，即信息采用改进的 DES 或 IDEA 对话密钥加密，然后使用 RSA 密钥加密对话密钥和信息摘要。对方收到信息后，用不同的密钥解密并可核对信息摘要。

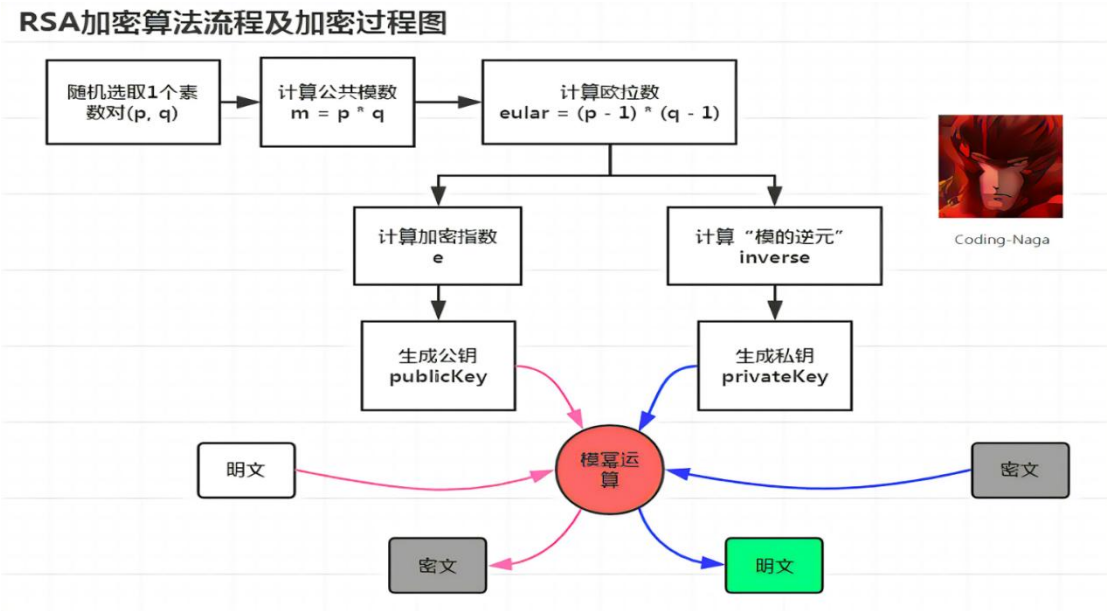
RSA 是被研究得最广泛的公钥算法，普遍认为是目前最优秀的公钥方案之一。



RSA 允许你选择公钥的大小。512 位的密钥被视为不安全的；768 位的密钥不用担心受到除了国家安全管理（NSA）外的其他事物的危害；RSA 在一些主要产品内部都有嵌入，像 Windows、网景 Navigator、Quicken 和 Lotus Notes。

算法原理：

RSA 公开密钥密码体制的原理是：根据数论，寻求两个大素数比较简单，而将它们的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。



算法攻击：

迄今为止，对 RSA 的攻击已经很多，但都没有对它构成真正的威胁。

1.选择密码攻击

一般攻击者是将某一信息进行下伪装，让拥有私钥的实体签名；然后，经过计算就可得到它所想要的信息。

攻击利用的都是同一个弱点，即存在这样一个事实：乘幂保留了输入的乘法结构。**这个固有的问题来自于公钥密码系统的最基本的特征，即每个人都能使用公钥加密信息。**

从算法上无法解决这一问题，改进措施有两条：
一是采用好的公钥协议保证工作过程中实体不对其他实体任意产生的信息解密，不对自己一无所知的信息签名。
二是决不对陌生人送来的随机文档签名，或签名时首先对文档作 Hash 处理，或同时使用不同的签名算法。

2.小指数攻击
当公钥 e 取较小的值，虽然会使加密变得易于实现，速度有所提高，但这样做也是不安全的。最简单的办法就是 e 和 d 都取较大的值。

因为密钥的产生受素数产生技术的限制，所以也有它的局限性。
（1）密钥的产生受素数产生技术的限制，因而难以做到一次一密；
（2）分组长度太大，为保证安全性，n 至少也要 600 比特以上，使运算代价很高，尤其是速度较慢，比对称密码算法慢几个数量级；随着大整数素因数分解算法的改进和计算机计算能力的提高，对 n 的长度在不断增加，不利于实现数据格式的标准化。