2. 加密原理

2.1 术语

加密语言：
对称密钥密码学：
发送方和接收方的密钥相同
公开密钥密码学：
发送方使用接收方的公钥进行加密，接收方使用自己的私钥进行解密

对称密钥加密：
Bob和Alice共享一个对称式的密钥
e.g.,密钥在单码替换加密方法中是替换模式
替换模式：见图

2.2 对称密钥加密学

对称密钥加密学：DES Data Encryption Standard
US加密标准 [NIST 1993]
56-bit 对称密钥，64-bit 明文输入
DES有多安全：
DES挑战：56-bit密钥加密的短语 (“Strongcryptography makes the world a safer place”) 被解密，用了4个月的时间
可能有后门
使DES更安全：
使用3个key，三重DES运算
密文分组成串技术

替换密码: 将一个事情换成另外一个事情

- 单码替换密码: 将一个字母替换成另外一个字母

```
plaintext:   abcdefghijklmnopqrstuvwxyz

ciphertext:  mnbvcxzasdfghjklpoiuytrewq
```

E.g.: Plaintext: bob. i love you. alice

ciphertext: nkn. s gktc wky. mgsbc

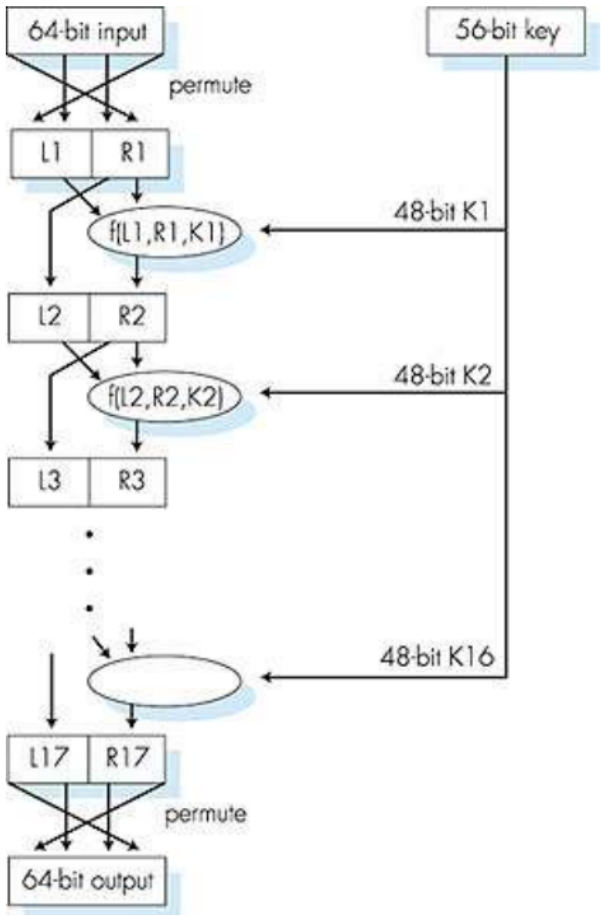
对称密钥加密学:DES

DES operation

初始替换

16 轮一样的函数应用，每一轮使用的不同的48bit密钥

最终替换



AES: Advanced Encryption Standard

新的对称：密钥NIST标准（Nov.2001）用于替换DES

数据128bit成组加密

128，192，or 256 bit keys

穷尽法解密如果使用1秒钟破解DES，需要花费149万亿年破解AES

块密码：

一个循环：一个输入bit影响8个输出bit

多重循环：每个输入bi影响所有输出bit

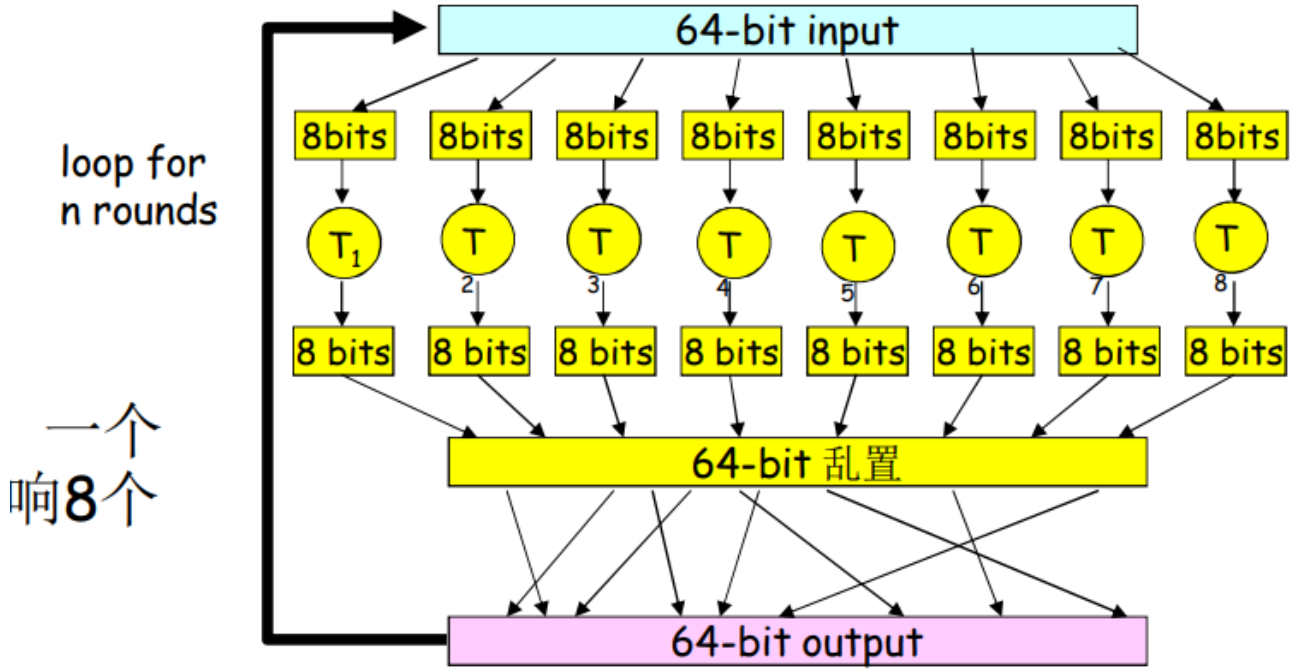
块密码：DES,3DES,AES

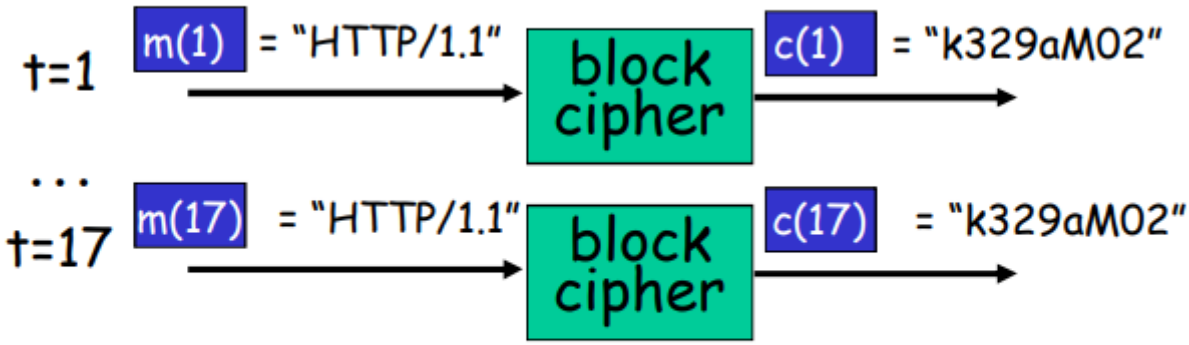
如果输入块重复，将会得到相同的密文块

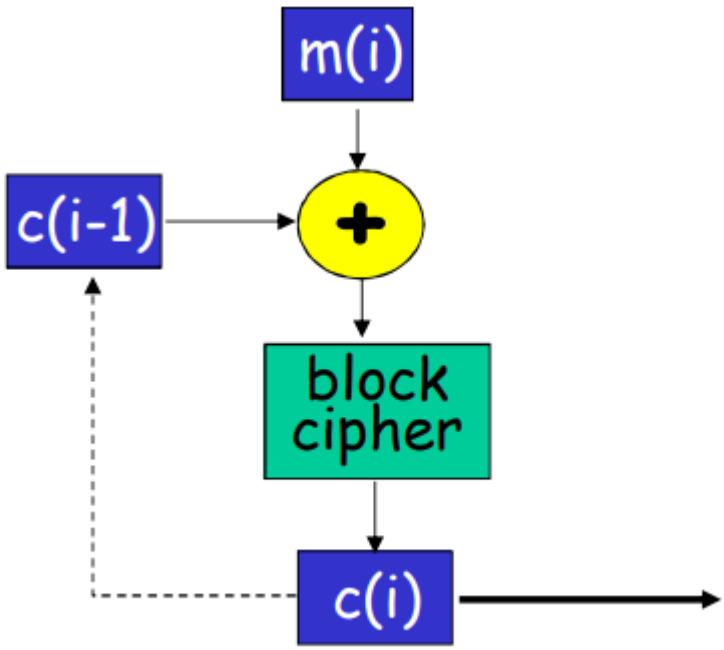
密码块链：异或第i轮输入m（i）与前一轮的密文，c（i-1）

c（0）铭文传输到接收端

what happens in HTTP/1.1 scenario from above


64-bit input
8bits
8bits
8bits
8bits
8bits
8bits
8bits
8bits
loop for
n rounds
T1
T2
T3
T4
T5
T6
T7
T8
8 bits
8 bits
8 bits
8 bits
8 bits
8 bits
8 bits
8 bits
一个
响8个
64-bit 乱置
64-bit output


t=1
m(1) = "HTTP/1.1"
block cipher
c(1) = "k329aM02"
...
t=17
m(17) = "HTTP/1.1"
block cipher
c(17) = "k329aM02"


m(i)
c(i-1)
+
block cipher
c(i)

2.3 公开密钥密码学

加密代价是对称的大约1000倍

发送方和接收方无需共享密钥
一个实体的公钥公诸于众
    一般是证书
私钥只有他自己知道

要求：
    不能用公钥推算出私钥
    私钥解密（公钥加密的数据）=明文

先加密再解密：常规加解密
先解密再加密：数字签名

RSA：见图

RSA: 选择密钥

1. 选择2个很大的质数 p, q.
   (e.g., 1024 bits each)
2. 计算 $n = pq$, $z = (p-1)(q-1)$
3. 选择一个 e (要求 $e<n$) 和z 没有一个公共因子，互素
   ("relatively prime").
4. 选择 d 使得 $ed-1$ 正好能够被z整除.
   (也就是： $ed \bmod z = 1$).
5. 公钥 $\underbrace{(n,e)}_{K_B^+}$. 私钥 $\underbrace{(n,d)}_{K_B^-}$.

RSA: 加密,解密

0. 给定按照上述算法得到的 (n,e) 和 (n,d)
1. 加密一个bit模式, m, 如此计算：
   $c = m^e \bmod n$ (i.e., m^e除以n的余数)
2. 对接收到的密文c解密，如此计算
   $m = c^d \bmod n$ (i.e., c^e除以n的余数)

Magic happens!

$$m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$$

RSA 例子:

Bob 选择 $p=5, q=7$. 因此 $n=35, z=24$.

$e=5$ (so e, z 互素).

$d=29$ (so $ed-1$ 能够被 z整除).

encrypt:

letter	m	m^e	$c = m^e \bmod n$
l	12	1524832	17

decrypt:

c	c^d	$m = c^d \bmod n$	letter
17	481968572106750915091411825223071697	12	l

RSA: 为什么

$$m = (m^e \bmod n)^d \bmod n$$

一个有用的数论定理: 如果 p,q 都是素数, $n = pq$, 那么:

$$x^y \bmod n = x^{y \bmod (p-1)(q-1)} \bmod n$$

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$

$$= m^{ed \bmod (p-1)(q-1)} \bmod n$$

(使用上述定理)

$$= m^1 \bmod n$$

(因为我们选择ed 使得正好被z 除余1)

$$= m$$

RSA：另外一个重要的特性

下面的特性将在后面非常有用

$$\underbrace{K_B^-(K_B^+(m))}_{\text{先用公钥，然后用私钥}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{先用私钥，然后用公钥}}$$

结果一致!

2.4 解密的集中类型

加密算法已知，求密钥
加密算法和密钥都不知道

唯密文攻击
已知明文攻击
    已经知道部分密文和明文的对应关系
选择明文攻击
    攻击者能够选择一段明文，并得到密文