本文为密码学笔记。

- 分组密码运行模式
  - 电话本 ECB 模式
  - 密码分组链接 CBC 模式
  - 密码反馈 CFB 模式
  - 输出反馈 OFB 模式
  - 计数器 CTR 模式
- IDEA
  - 设计原理
    - 混肴
    - 扩散
  - 加密过程
    - 轮结构
    - 子密钥产生
- AES
  - X乘

公钥密码

- 公钥密码体制
  - 对称密码算法缺陷
  - 双重加密方案
  - 公钥密码体制的基本原理
    - 基于公钥密码体制的加解密过程
    - 基于公钥密码体制的认证过程
    - 基于公钥密码体制的同时实现加密和认证的过程
  - 公钥密码算法应满足的要求
  - 对公钥密码体制的攻击
- RSA
  - 算法描述
  - 计算问题
    - 加密与解密过程
  - 安全性
  - 攻击手段
- 背包密码体制
- ElGamal密码体制
  - 原根
- 椭圆曲线密码体制
  - Diffie-Hellman密钥交换协议
  - ElGamal密码体制

密钥管理与密钥分配

- 单钥加密体制的密钥分配
  - 密钥分配的基本方法
  - NS密钥分配协议
  - 无中心(KDC)的密钥分配
- 公钥加密体制的密钥管理
  - 公钥分配
  - 基于公钥加密的会话密钥分配协议
    - 简单分配
    - 具有保密性和认证性的密钥分配
  - Diffie-Hellman密钥交换协议
- 随机数
  - 伪随机数产生器
  - 基于分组密码算法的随机数产生器
- 秘密分割

绪论

信息安全面临的威胁

安全威胁

- **被动攻击**

  也称窃听，以获取信息为目的。可分为两类：获取消息的内容和业务流分析

- **主动攻击**

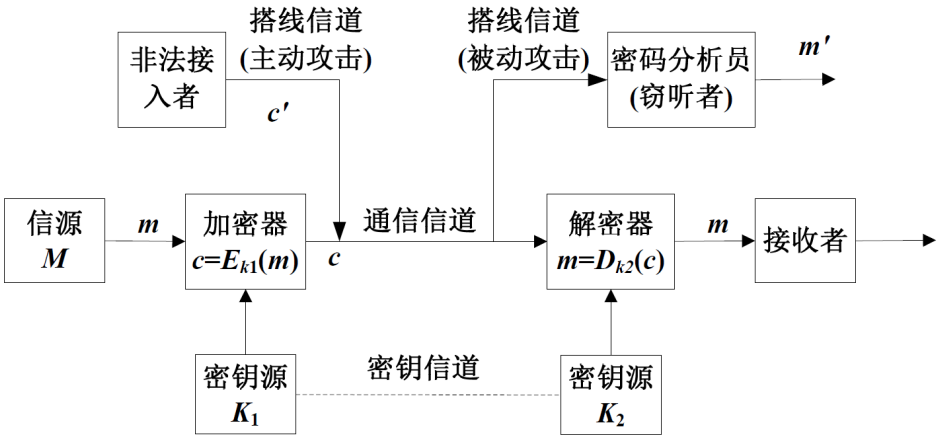
  对数据流进行篡改或产生假的数据流

安全业务

- **鉴别业务**：也称认证业务，最基本的安全服务，是对付假冒攻击的有效方法，以保障通信的真实性，可细分为对等实体鉴别和数据源鉴别
- **访问控制**：用于防止资源的未授权使用，检查用户是否具有对某一资源的访问权
- **机密性业务**：保护信息（数据）不泄露或不泄露给那些未获授权访问信息的实体
- **数据完整性业务**：保证接收的消息未经复制、插入、篡改、重排或重放。即保证接收的消息和发送的消息完全一样
- **抗抵赖业务**：即不可否认性业务，防止通信双方中的某一方对所传送消息的否认，保护通信实体免遭来自其他合法实体的威胁

密码学基本概念

保密通信系统



M 表示明文空间、C 表示密文空间；密钥空间为 K_1 和 K_2；加密变换 E_{K_1}：$M \to C$；解密变换 D_{K_2}：$C \to M$；称 $(M, C, K_1, K_2, E_{K_1}, D_{K_2})$ 为保密通信系统或密码体制

密码体制分类

$$\text{密码体制}\begin{cases}\text{对称密码体制}\begin{cases}\text{流密码}\\\text{分组密码}\end{cases}\\\text{非对称密码体制——公钥密码}\end{cases}$$

密码攻击

攻击类型	攻击者掌握的内容
唯密文攻击	加密算法，截获的部分密文
已知明文攻击	加密算法，截获的部分密文，一个或多个明密文对
选择明文攻击	加密算法，截获的部分密文，自己选择的明文消息及由密钥产生的相应密文
选择密文攻击	加密算法，截获的部分密文，自己选择的密文消息及相应的被解密的明文

古典密码算法

单表代换密码

凯撒密码

a	b	c	d	e	f	$\cdots$	z
0	1	2	3	4	5	$\cdots$	25

- 加密代换：$c = E_3(m) \equiv m + 3 (\mod 26), 0 \le m \le 25$
- 解密代换：$m = D_3(c) \equiv c - 3 (\mod 26), 0 \le c \le 25$

移位变换

- 加密变换：$c=E_k(m)\equiv m+k(\mod 26), 0\leq m,k\leq 25$
- 解密变换：$m=D_k(c)\equiv c-k(\mod 26), 0\leq c,k\leq 25$

仿射变换

a,b 是密钥，满足 $0\leq m,k\leq 25$ 和 $\gcd(a,26)=1$ 的整数

- 加密变换：$c=E_{a,b}(m)\equiv am+b(\mod 26)$
- 解密变换：$m=D_{a,b}(c)\equiv a^{-1}(c-b)(\mod 26)$

多表代换密码

维吉尼亚密码

在长为 m 的密码中,任何一个字母可被映射到 26 个字母中的一个

- 明文 $p\in(Z_{26})^m$，密文 $c\in(Z_{26})^m$，密钥 $k\in(Z_{26})^m$
- 加密 $c=(p_1+k_1,p_2+k_2,\cdots,p_m+k_m)\mod 26$
- 解密 $p=(c_1-k_1,c_2-k_2,\cdots,c_m-k_m)\mod 26$

多字母代换密码

Hill 密码

m 个连续明文字母被 m 个密文字母代替，由 m 个线性方程决定替代方法

$m=3$ 时的系统描述：编码 $(a=0,b=1,\cdots,z=25)$，$C=KP$

$$\begin{pmatrix}C_1\\C_2\\C_3\end{pmatrix}=\begin{pmatrix}k_{11}&k_{12}&k_{13}\\k_{21}&k_{22}&k_{23}\\k_{31}&k_{32}&k_{33}\end{pmatrix}\begin{pmatrix}p_1\\p_2\\p_3\end{pmatrix}(\mod 26)$$

- 破解：$C=KP\Longrightarrow K=CP^{-1}$

多字母仿射变换密码

- 首先将明文 M 分为由 n 个字母构成的分组 $M_1,M_2,\cdots,M_j$，对每个分组 M_i 的加密为

  $C_i\equiv AM_i+B(\mod N), i=1,2,\cdots,j$，其中，$(A,B)$ 是密钥，A 是 $n\times n$ 的可逆矩阵，满足 $\gcd(|A|,N)=1$ ($|A|$ 是行列式)

  $B=(b_1,b_2,\cdots,b_n)^T$

  $C_i=(c_1,c_2,\cdots,c_n)^T$

  $M_i=(m_1,m_2,\cdots,m_n)^T$

- 对密文分组 C_i 的解密为

  $M_i\equiv A^{-1}(C_i-B)(\mod N), i=1,2,\cdots,j$

置换密码

又称换位密码，明文字母保持相同，但顺序被打乱了。即对明文字母的某种置换取得一种类型完全不同的映射,即明文中字母重新排列,本身未变,位置发生改变

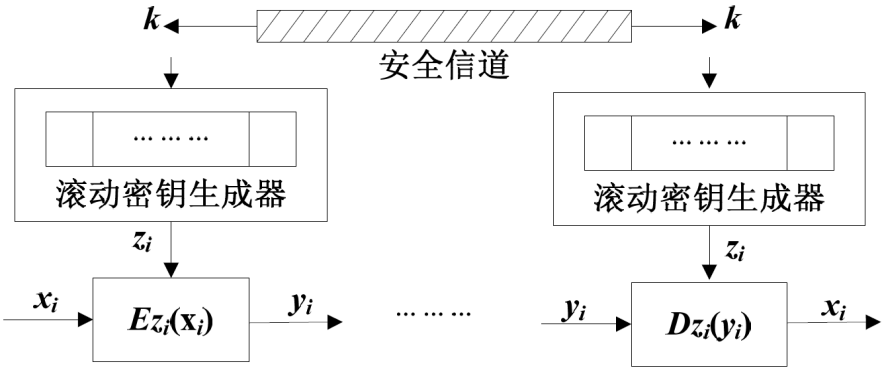
栅栏密码

- 加密：明文以固定的宽度水平地写在一张图表纸上,密文按垂直方向读出
- 解密：将密文按相同的宽度垂直地写在图表纸上,然后水平地读出明文

流密码

概念

同步流密码

- 由**密钥流发生器** f 产生：$z_i = f(k, \sigma_i)$，σ_i 是加密器中的记忆元件在时刻 i 的状态，可表示为 $\sigma_i = (a_n, a_{n-1}, \cdots, a_1)$
- 按照加密器中记忆元件的存储状态 σ_i 是否依赖于明文字符流，流密码可进一步分为**同步流密码**和**自同步流密码**
- 

有限状态自动机

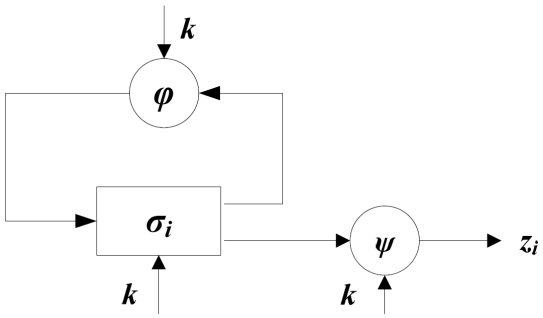
- 流密码中任意时刻密钥流和密文的输出与状态密切相关。可用具有离散输入集和输出集的有限状态自动机模型表述
  - **有限状态集**：$S = \{s_i | i = 1, 2, \cdots, l\}$ 共 l 个可能状态
  - **有限输入字符集**：$A_1 = \left\{A_j^{(1)} | j = 1, 2, \cdots, m\right\}$

    有限输出字符集：$A_2 = \left\{A_k^{(2)} | k = 1, 2, \cdots, n\right\}$
  - **输出函数**：$A_k^{(2)} = f_1\left(s_i, A_j^{(1)}\right)$

    状态转移函数：$s_h = f_2\left(s_i, A_j^{(1)}\right)$

密钥流产生器

- 可将密钥流产生器看成参数为 k 的有限状态自动机，由输出符号集 Z、状态集 Σ (初始状态 σ_0)、状态转移函数 φ 和输出函数 Ψ 构成

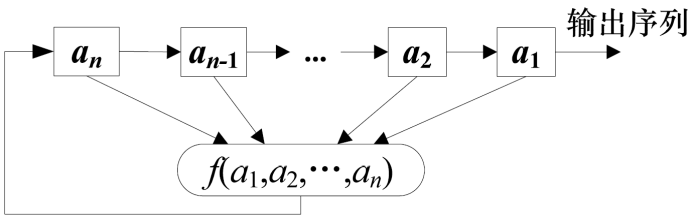


- 具有非线性的 φ 的有限状态自动机理论很不完善，相反，采用线性的 φ 和非线性的 Ψ 时能够进行深入分析并可以得到好的生成器

线性反馈移位寄存器

反馈移位寄存器

- $\mathrm{GF}(2)$ 上的 n 级 FSR 由 n 个二元存储器与一个反馈函数 $f(a_1, a_2, \cdots, a_n)$ 组成



LFSR

- $f(a_1, a_2, \cdots, a_n) = c_n a_1 \oplus c_{n-1} a_2 \oplus \cdots \oplus c_1 a_n, c_i = 0, 1$
- 若其初始状态非全 0，则其后继状态也不会为全 0。因此 n 级 LFSR 此输出序列的周期 $=$ 状态周期 $\leq 2^n - 1$
- 周期达到最大值的线性序列称为 m 序列

LFSR 的一元多项式表示

- 递推关系式 $a_{k+n} = c_1 a_{k+n-1} \oplus c_2 a_{k+n-2} \oplus \cdots \oplus c_n a_k, k \geq 1$

  特征多项式 $p(x) = 1 + c_1 x + c_2 x^2 + \cdots + c_n x^n$（NOTE：这里有个 $+1$）
- 记 a_i 的 $2^n - 1$ 个非零序列的全体为 $G(p(x))$
- n 级 LFSR 产生的序列有最大周期 $2^n - 1$ 的**必要**条件是其特征多项式 $p(x)$ 是不可约的，称这样的多项式为 n 次本原多项式

m 序列的伪随机性

游程

序列中连续的 0 或连续的 1 串称为 1 个游程

自相关函数

$R(\tau)=\frac{1}{T}\sum\limits_{k=1}^{T}(-1)^{a_k}(-1)^{a_{k+\tau}},0\le\tau\le T-1$

即为两个序列 $\{a_i\}$ 与 $\{a_{i+\tau}\}$ 在一个周期内对应位相同的位数与不同的位数之差

当 $\tau\neq 0$ 时，称 $R(\tau)$ 为**异相自相关函数**

伪噪声序列

好的伪随机序列应满足的 3 个公设

- $0,1$ **平衡性**：一个周期内，0、1 出现的次数分别为 $2^{n-1}-1$ 和 2^{n-1}（不能全 0）
- **游程特性**：一个周期内，总游程数为 2^n-1；对 $1\le i\le n-2$，长度为 i 的游程有 2^{n-i-1} 个，且 0、1 游程各半；长度为 $n-1$ 的 0 游程一个，长度为 n 的 1 游程一个
- **自相关函数**：$R(\tau)=\begin{cases}1, & \tau=0\\ -\frac{1}{2^n-1}, & 0<\tau\le 2^n-2\end{cases}$

m 序列的破译

$$\begin{aligned}(\ a_{n+1}\quad a_{n+2}\quad\cdots\quad a_{2n}\ )&=(\ c_n\quad c_{n-1}\quad\cdots\quad c_1\ )\begin{pmatrix}a_1 & a_2 & \cdots & a_n\\ a_2 & a_3 & \cdots & a_{n+1}\\ \vdots & \vdots & & \vdots\\ a_n & a_{n+1} & \cdots & a_{2n-1}\end{pmatrix}\\&=(\ c_n\quad c_{n-1}\quad\cdots\quad c_1\ )X\end{aligned}$$

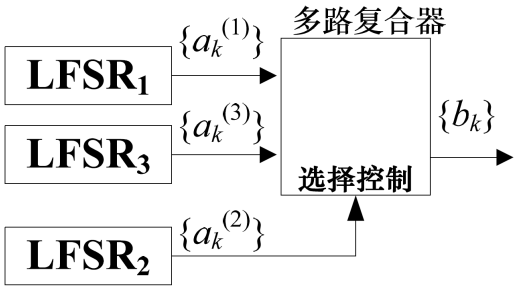
$$(\ c_n\quad c_{n-1}\quad\cdots\quad c_1\ )=(\ a_{n+1}\quad a_{n+2}\quad\cdots\quad a_{2n}\ )X^{-1}$$

非线性序列

密钥流满足的性质

- 种子密钥的长度足够长
- 极大的周期
- 良好的统计特性
- 极大的线性复杂度
- 极大的 k 错线性复杂度
- 抗统计分析
- 混乱性
- 扩散性
- 抗线性分析

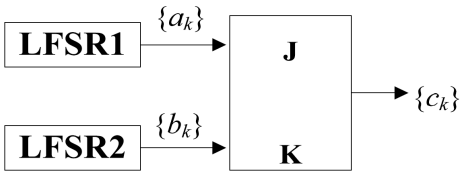
Geffe 序列生成器



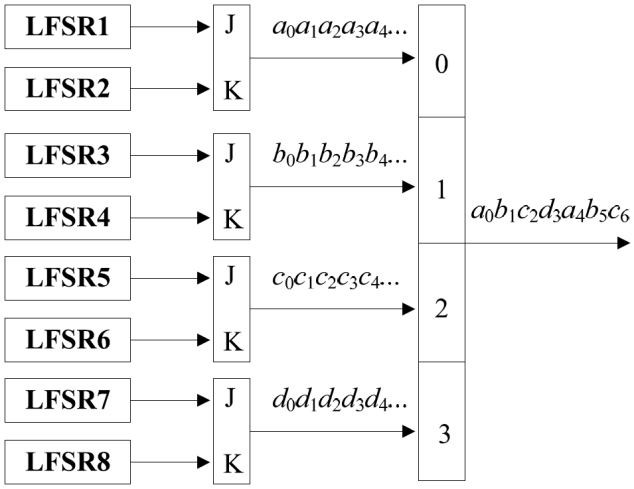
当 LFSR$_2$ 输出 1 时，LFSR$_2$ 与 LFSR$_1$ 相连接；当 LFSR$_2$ 输出 0 时，LFSR$_2$ 与 LFSR$_3$ 相连接

周期 $T=\prod_{i=1}^{3}(2^{n_i}-1)$

J-K 触发器

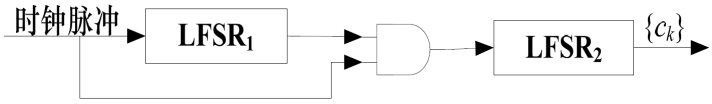


Pless 生成器



输出序列为 $a_0b_1c_2d_3a_4b_5c_6\cdots$

钟控序列生成器



当 LFSR$_1$ 输出 1 时，移位时钟脉冲通过与门使 LFSR$_2$ 进行一次移位，从而生成下一位；当 LFSR$_1$ 输出 0 时，移位时钟脉冲无法通过与门影响 LFSR$_2$，LFSR$_2$ 重复输出前一位

分组密码

分组密码概述

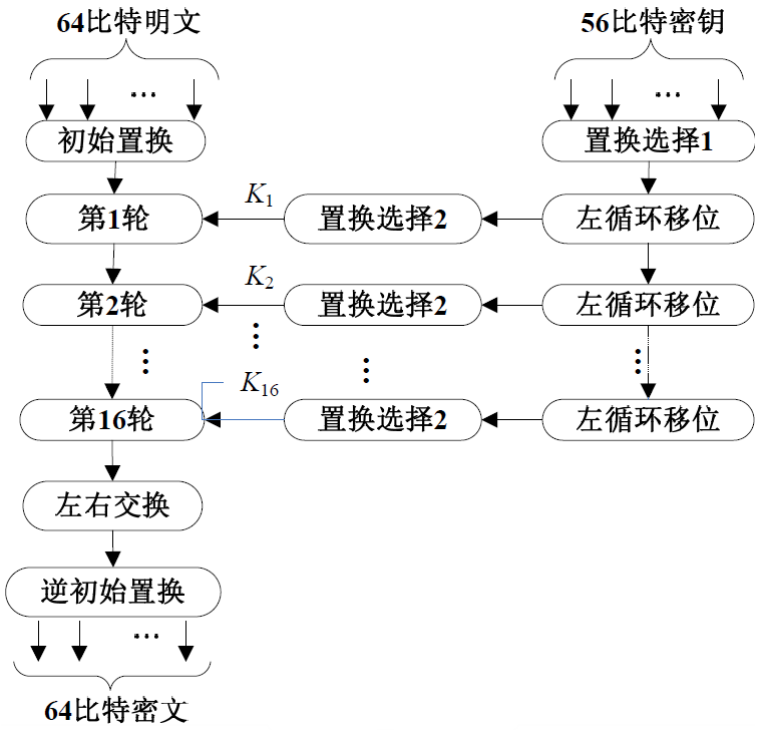
- 扩散

  将明文的统计特性散布到密文中去，明文的每一位影响密文的很多位，即将明文尽可能广泛的扩散到密文中去

- 混淆

  使明文、密文和密钥间的统计关系变得尽可能复杂，即使攻击者能够得到明文和密文的一些统计关系，也无法得到密钥

DES



- 取反特性

  对于明文分组 M，密文分组 C 和密钥 K，若 $C = DES_K(M)$， 则 $\overline{C} = DES_{\overline{K}}(\overline{M})$

- 弱密钥与半弱密钥

  $DES_{K'}(DES_K(m)) = m$，则称密钥 K 与密钥 K' 互为对合

  若 K 是自己的对合，即 $K_1 = K_2$ ，则称 K为 DES 的一个弱密钥

  若 K 存在异于自己的对合，则称 K 为DES的一个半弱密钥

- 不能抵抗穷搜索攻击

二重DES

$C = EK_2\left(EK_1\left[P\right]\right)$

- **中途相遇攻击**

  已知通过两重 DES 加密得到的一个明文密文对为 (P, C)，即 $C = EK_2(EK_1[P])$，那么存在 $X = EK_1[P] = DK_2[C]$

EDE(两个密钥的三重DES)

$C = EK_1(DK_2(EK_1[P]))$

三个密钥的三重DES

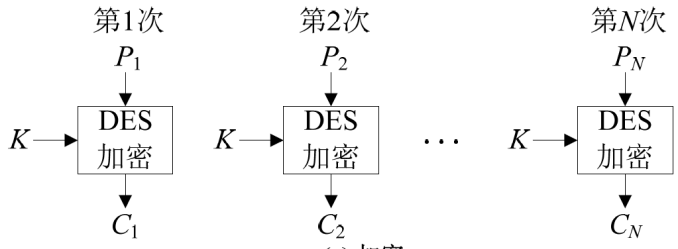
$C = EK_3(DK_2(EK_1[P]))$

分组密码运行模式

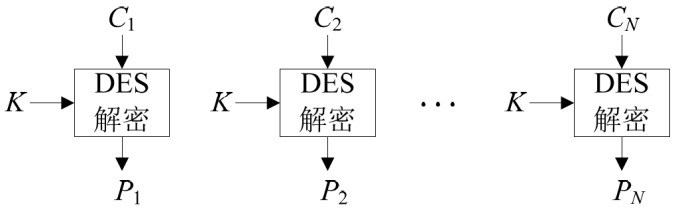
电话本 ECB 模式

Electric CodeBook

- 对明文分组后，用同一密钥逐一加密
- 适合短消息，如传递DES密钥
- 错误不会传播



(a) 加密

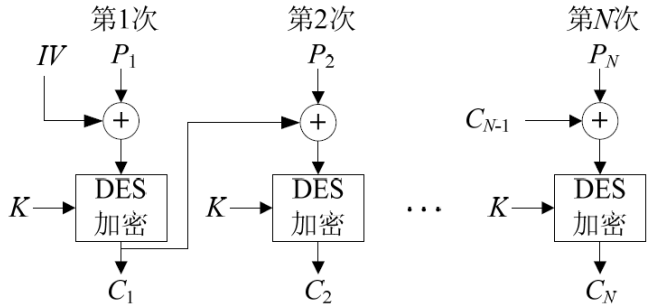


(b) 解密

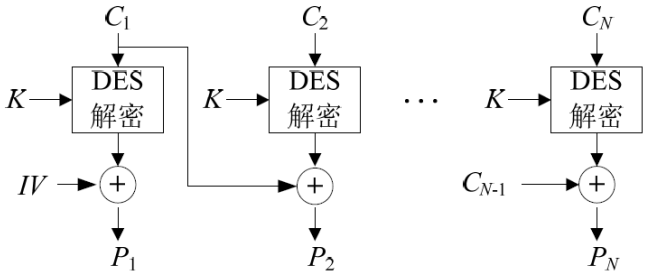
密码分组链接 CBC 模式

Cipher Block Chaining

- $C_i = \mathrm{DES}_K(P_i \oplus C_{i-1}), C_0 = IV$
- 可用于消息认证
- 错误传播：当前和下一分组



(a) 加密

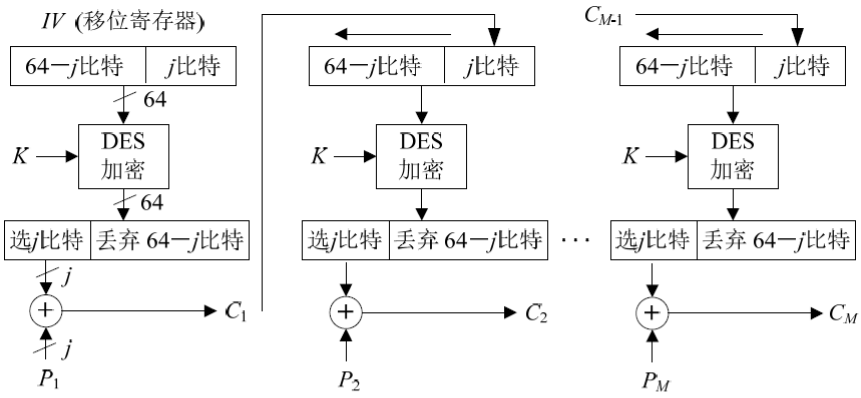


(b) 解密

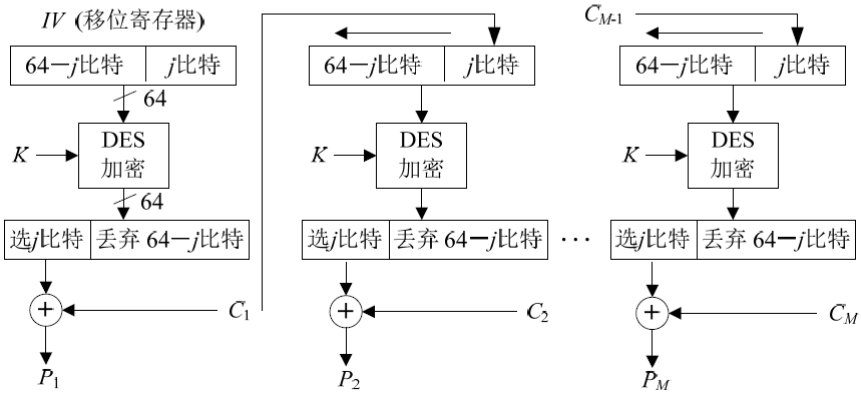
密码反馈 CFB 模式

Cipher FeedBack

- j 可选 $1, 8, 64$ 等， 记作 CFB-1, CFB-8, CFB-64
- 1 比特密文传输错误传播约 $64/j$ 个分组(不包括当前分组)
- 可用于认证



(a) 加密

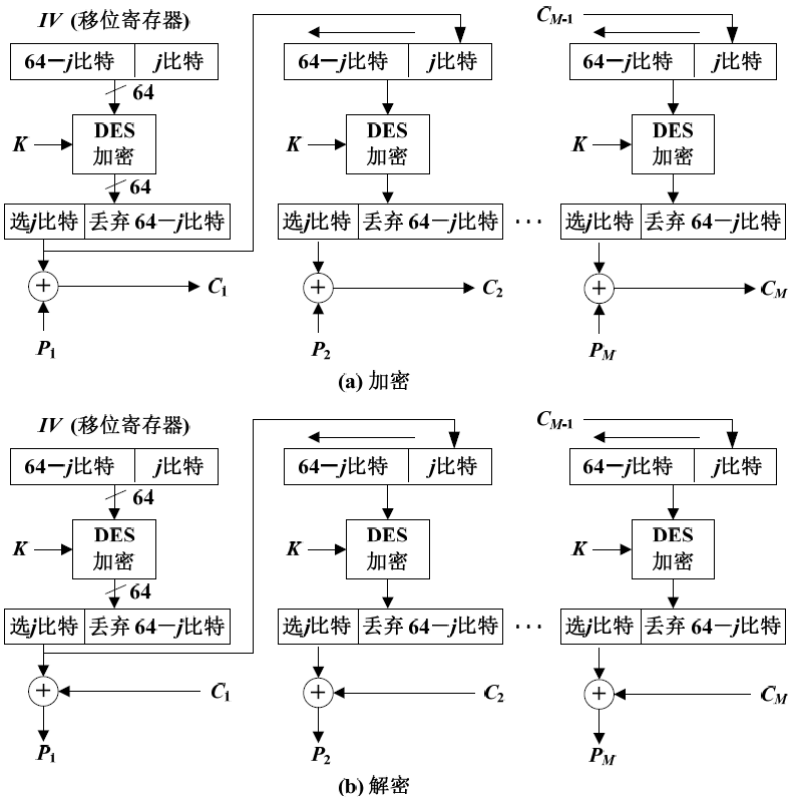


(b) 解密

输出反馈 OFB 模式

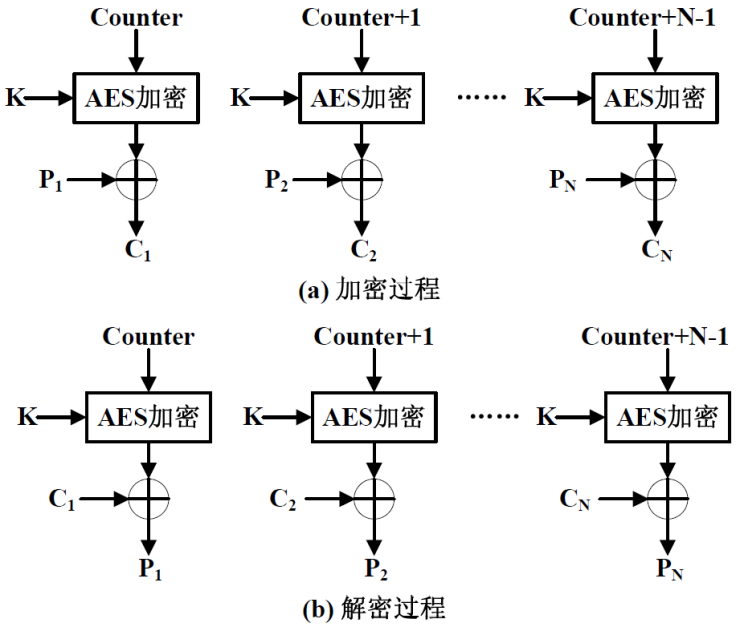
Output FeedBack

- 错误不易传播
- 密文易于篡改，不适合认证



计数器 CTR 模式

- 效率高，可并行加密，各块可单独处理，可预处理
- 加密数据块可随机访问



IDEA

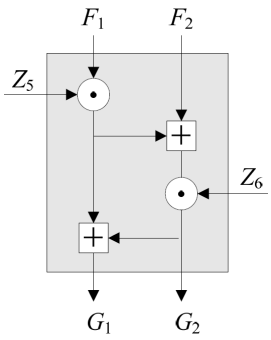
设计原理

混肴

- 逐比特异或 $\oplus$
- 模 2^{16} 加法 $\boxplus$
- 模 $2^{16}+1$ 乘法 $\odot$，0000000000000000 作 2^{16} 处理

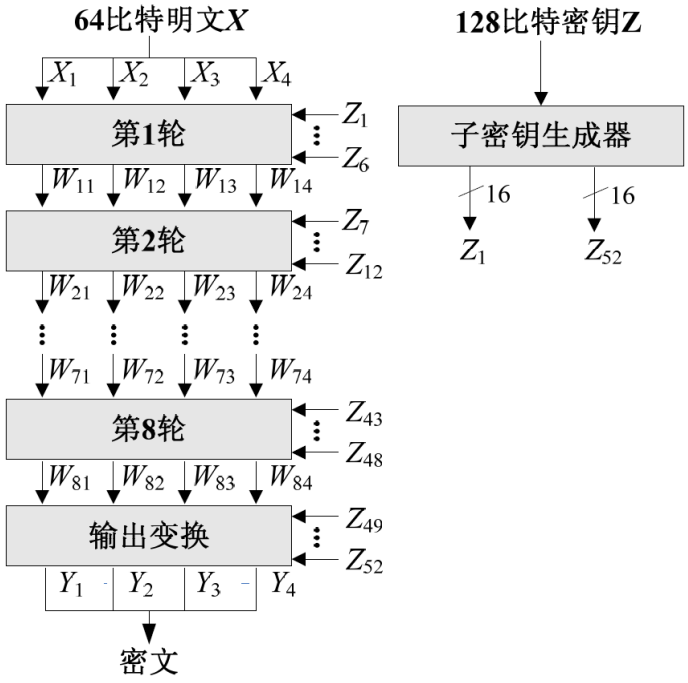
扩散

MA 结构



加密过程

64 比特的明文划分成 4 个 16 比特子段



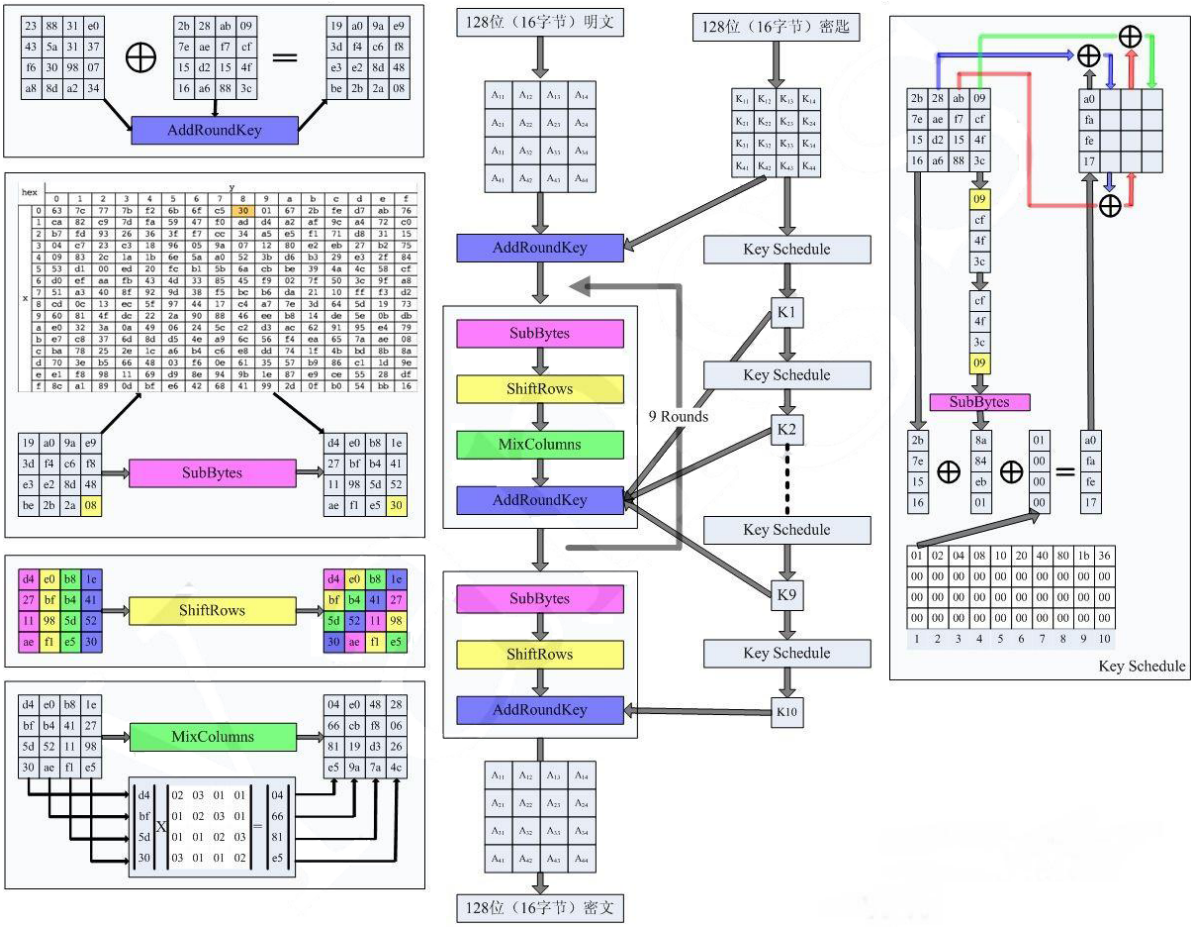
轮结构

- 变换：输入 4 个子段和 6 个子密钥；输出 4 个子段
- 输出变换：仅需 4 个子密钥

子密钥产生

- Z_1，Z_2，$\cdots$，Z_8 直接从加密密钥中取
- 加密密钥循环左移 25 位，再取 Z_9，Z_{10}，$\cdots$，Z_{16}，$\cdots$

AES



X乘

$xtime(b(x)) = x \cdot b(x)(\mod m(x))$

> Q: 计算 $57 \cdot 13, m(x) = x^8 + x^4 + x^3 + x + 1$
>
> $57 \cdot 02 = xtime(57) = AE$
> $57 \cdot 04 = xtime(AE) = 47$
> $57 \cdot 08 = xtime(47) = 8E$
> $57 \cdot 10 = xtime(8E) = 07$
> $57 \cdot 13 = 57 \cdot (01 \oplus 02 \oplus 10) = 57 \oplus AE \oplus 07 = FE$

公钥密码

公钥密码体制

对称密码算法缺陷

- 密钥分配问题：双方保密通信前需要通过安全信道协商密钥
- 密钥管理问题：任何两用户间都需要有共享的秘密钥，n 个用户需要 C_n^2 个密钥
- 无法实现签名功能：当 Alice 收到 Bob 用其密钥加密生成的电子文挡时，Bob 无法向第三方证明该电子文档确实来源于 Bob

双重加密方案

- Alice 发送 $E_A(P)$ 给 Bob
- Bob 发送 $E_B\left(E_A(P)\right)$ 给 Alice
- Alice 发送 $D_A(E_B(E_A(P)))=D_A(E_A(E_B(P)))=E_B(P)$ 给 Bob
- Bob 解密 $D_B(E_B(P))=P$

缺陷：

- 要求构造一个函数, 满足 $E_B(E_A(P))=E_A(E_B(P))$
- 无法验证 Alice 或 Bob 的身份

公钥密码体制的基本原理

基于公钥密码体制的加解密过程

- 密钥的生成：生成用户 Alice 的公钥 PU_a 和私钥 PR_a
- 公开钥的发布：将 Alice 的公钥 PU_a 公开
- 加密：用户 Bob 想向 Alice 发送消息 m，则需获得 Alice 的公钥 PU_a，然后加密消息m得到 $c=E_{PU_a}[m]$，其中，E 是加密算法
- 解密：Alice 收到密文 c 后，用自己的私钥 PR_a 解密，即 $m=D_{PR_a}[c]$，其中，D 是解密算法。因为只有 Alice 知道PR_a，所以其他人无法对密文 c 解密

基于公钥密码体制的认证过程

- 用户 Bob 用自己的私钥 PR_b 对消息 m 加密，表示为 $c=E_{PR_B}[m]$
- Bob 将 c 发给 Alice。Alice 收到 c 后，用 Bob 的公钥 PU_B 对 c 解密，表示为 $m=D_{PU_B}[c]$

公钥密码算法运算速度很慢，对较长的明文直接加密用来签名不可行。改进的方法是先将文件经过单向压缩函数(hash)压缩成长度较小的比特串，得到的比特串称为认证符(哈希值)，然后对认证符(哈希值)进行加密。

基于公钥密码体制的同时实现加密和认证的过程

- 先签名后加密： Alice 首先用自己的私钥 PR_a 对消息 m 加密，用于提供认证(数字签名)；再用 Bob 的公钥 PU_b 第 2 次加密，表示为 $c=E_{PU_b}\left[E_{PR_a}[m]\right]$
- 先解密再验证： Bob 的解密认证过程为 $m=D_{PU_a}\left[D_{PR_b}[c]\right]$

公钥密码算法应满足的要求

- 密钥对(PU 和 PR)的生成在计算上是容易的
- 用公钥对消息 m 加密，即 $c=E_{PU}[m]$ 在计算上是容易的
- 用私钥对 c 解密，即 $m=D_{PR}[c]$ 在计算上是容易的
- 攻击者由公钥 PU 求私钥 PR 在计算上是不可行的，等同于困难问题
- 攻击者由 c 和公钥 PU 恢复明文 m 在计算上是不可行的，等同于困难问题
- 两个密钥使用的次序可调换，即 $D_{PU}[E_{PR}(m)]=D_{PR}[E_{PU}(m)]$

满足以上要求的本质在于构造一个陷门单向函数

陷门单向函数：若额外给定某些附加信息后，由给定的 y 找到 $x \in X$很容易，则称这样的单向函数为限门单向函数，即：

- 当 k 和 y 已知时，求解 $x = f_k^{-1}(y)$ 很容易
- 当 y 已知但 k 未知时，求解 $x = f_k^{-1}(y)$ 很困难

对公钥密码体制的攻击

- 穷搜索攻击
- 寻找从公钥计算私钥的方法
- 可能字攻击

RSA

算法描述

- 密钥的产生
  - 选两个大素数 p 和 q
  - 计算 $n = pq$，$\varphi(n) = (p-1)(q-1)$
  - 选一整数 e，满足 $1 < e < \varphi(n)$，且 $\gcd(\varphi(n), e) = 1$
  - 计算 d，满足 $de \equiv 1 \mod \varphi(n)$，即 d 是 e 在模 $\varphi(n)$ 下的乘法逆元，e 与 $\varphi(n)$ 互素，模 $\varphi(n)$ 的乘法逆元一定存在
  - 公钥：$\{e, n\}$；私钥：$\{d, p, q\}$ 或 $\{d\}$（密钥生成若是由系统负责完成，则用户可能不知道 p 和 q）
- 加密
  - $m < n$，计算 $c \equiv m^e \mod n$
  - 若 m 较大，将 m 分组，每个分组对应的十进制数小于 n
- 解密
  - 计算 $m \equiv c^d \mod n$

计算问题

加密与解密过程

- 模运算的累次乘法

  $ab \mod n = [(a \mod n)(b \mod n)] \mod n$

- 快速指数算法

  $m = b_k 2^k + b_{k-1} 2^{k-1} + \cdots + b_1 2 + b_0$

  $a^m = ((\cdots(((a^{b_k})^2 a^{b_{k-1}})^2 a^{b_{k-2}})^2 \cdots a^{b_1})^2 a^{b_0}$

- 中国剩余定理CRT加速解密

  $$m = c^d \mod n \Longleftrightarrow \begin{cases} m = c^d \mod p \\ m = c^d \mod q \end{cases}$$

  $$\Longrightarrow \begin{cases} m = a_1 \mod p \\ m = a_2 \mod q \end{cases}$$

  $m = [a_1 q q^{-1} + a_2 p p^{-1}] \mod n$

> **广义欧几里得除法**

j	s_j	t_j	q_{j+1}	r_{j+1}
-3				a
-2	1	0		b
-1	0	1	q_0	r_0
0	s_0	t_0	q_1	r_1
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
n	s_n	t_n	q_{n+1}	$r_{n+1}=0$

$$\begin{cases} s_j=(-q_j)s_{j-1}+s_{j-2} \\ t_j=(-q_j)t_{j-1}+t_{j-2} \\ q_{j+1}=\left[\frac{r_{j-1}}{r_j}\right] \\ r_{j+1}=(-q_{j+1})r_j+r_{j-1} \end{cases}$$

安全性

- $|p-q|$ 足够大，即 p 和 q 的长度应相差几位
- $p-1$ 和 $q-1$ 均有大素数因子(分别记为 p' 和 q')；$p'-1$ 和 $q'-1$ 也均有大素数因子
- $\gcd(p-1,q-1)$ 应该较小

攻击手段

- 共模攻击
- 共指数攻击
- 低指数攻击
- 选择密文攻击(RSA密码算法不能抵抗)

背包密码体制

$A=(a_1,a_2,\cdots,a_n)$ 是由 n 个不同的正整数构成的 n 元组，s 是另一已知的正整数。背包问题就是从 A 中找出所有的 a_i，使其和等于 s。其中，A 称为背包向量，s 是背包容积

将 $x(1\le x\le 2n-1)$ 写成长为 n 的二元表示, $f(x)$ 定义为 A 中所有可能选择 a_i 的和，即
$f(1)=f(0\cdots001)=a_n$
$\cdots$
$f(2n-1)=f(1\cdots111)=a_1+a_2+\cdots+a_n$

背包向量 $A=(a_1,a_2,\cdots,a_n)$ 称为超递增的，如果 $a_j>\sum_{i=1}^{j-1}a_i, j=2,\cdots,n$

- 密钥产生
  - 构造超递增背包向量 $A=(a_1,a_2,\cdots,a_n)$
  - 用模乘对 A 进行伪装，其中，模数 k 和乘数 t 皆取为常量，满足，$\gcd(t,k)=1$，即 t 在模 k 下有乘法逆元。
  - 设 $b_i\equiv t\cdot a_i \mod k, i=1,2,\cdots,n$，得一新的背包向量 $B=(b_1,b_2,\cdots,b_n)$，记为 $B\equiv t\cdot A \mod k$
  - 用户以 B 作为自己的公钥，A , t , k 为私钥
- 加密

    - 对明文分组 $x=(x_1x_2\cdots x_n)$ 的加密运算为 $c=f(x)=B\cdot B_x \mod k$
- 解密
    - 由 $s\equiv t^{-1}c \mod k$，求出 s 作为超递增背包向量 A 的容积
    - 再由超递增背包向量 A 解背包问题即得 $x=(x_1x_2\cdots x_n)$

ElGamal密码体制

- 密钥产生
    - 选择大素数 p；选择本原根 g，$1<g<p$；选择 x，$1<x<p-1$
    - 计算 $y=g^x \mod p$，公钥是 (p,g,y)，私钥是 x
- 加密
    - 明文消息 M，$0<M<p$
    - 随机选一整数 k , $1\leq k\leq p-1$
    - 计算对 $C_1\equiv g^k \mod p, C_2\equiv y^kM \mod p$，密文 $C=C1||C2$ (级联)
- 解密
    - $M=C_2/C_1^x \mod p$
    - $C_2/C_1^x \mod p=y^kM/g^{kx} \mod p=y^kM/y^k \mod p=M \mod p$

原根

- **求奇素数p原根**

> - STEP1: **求一个原根g**
>
> 求出$p-1$的所有素因数$q_1,\cdots,q_s$，则g是模p的原根$\Longleftrightarrow \forall i,g^{\frac{p-1}{q_i}}\not\equiv 1 \mod p)$
> - STEP2: **求所有原根**
> 对于$(d,\varphi(m))=1$，g^d为原根

- **求p^α原根**

> - STEP1: **求p的一个原根g**
> - STEP2: **求p^α的原根**
>   - 若$g^{p-1}\not\equiv 1 \mod p^2)$，则$g$为原根
>   - 若$(g+p)^{p-1}\not\equiv 1 \mod p^2)$，则$g+p$为原根

- **求$2p^\alpha$原根**

> - STEP1: **求p^α的一个原根g**
> - STEP2: **求$2p^\alpha$的原根**
> g与$g+p^\alpha$中的奇数为原根

椭圆曲线密码体制

Diffie-Hellman密钥交换协议

- 取素数 $p \approx 2180$ 和参数 a、b，则得椭圆曲线上的点及无穷远点构成Abel群 $E_p(a,b)$
- 取 $E_p(a,b)$ 的某个生成元 $G=(x_1,y_1)$，G 的阶，即满足 $nG=O$ 的最小正整数 n很大。$E_p(a,b)$ 和 G 作为公开参数
- Alice 和 Bob 之间的密钥交换如下进行
  - Alice 随机选取保密的整数 $n_A < n$，计算 $P_A = n_A G$ 并发给 Bob
  - Bob 随机选取秘密的 n_B 并计算 $P_B = n_B G$ 发给 Alice
  - Alice 和 Bob 分别由 $K = n_A P_B$ 和 $K = n_B P_A$ 生成共享的密钥
    $K = n_A P_B = n_A(n_B G) = n_B(n_A G) = n_B P_A$

ElGamal密码体制

- 选择椭圆曲线 $E_p(a,b)$，将明文 m 通过嵌入到椭圆曲线上得点 P_m
  - 设明文 m，计算 $x=\{mk+j, j=0,1,2,\cdots,k-1\}$， k为正整数，通常取值 $30 \sim 50$。若 $k=30$，计算一系列 x：$\{30m, 30m+1, 30m+2, \cdots\}$
  - 直到 $x^3+ax+b(\mod p)$ 是平方剩余，即得到椭圆曲线上的点 $(x, \sqrt{x^3+ax+b})$
  - 在 1 到 $p-1$ 的整数中，一半是模 p 的平方剩余。k 次找到 x，使得 $x^3+ax+b(\mod p)$ 是平方剩余的概率不小于 $1-2^{-k}$。
  - 从椭圆曲线上的点 (x,y) 得到 m，只须求 $m=\lfloor x/k \rfloor$
- 取 $E_p(a,b)$ 的一个生成元 G，$E_p(a,b)$ 和 G 作为公开参数
- Alice 选 n_A 作为私钥，并以 $P_A = n_A G$ 作为公开钥
- Bob 向 Alice 发送消息 P_m，可选取随机数 k，产生点对 $C_m = \{kG,\ P_m + kP_A\}$ 作为密文
- Alice 以密文点对中的第二个点减去其私钥与第一个点倍乘的结果，即
  $(P_m + kP_A) - n_A kG = P_m + k(n_A G) - n_A kG = P_m$

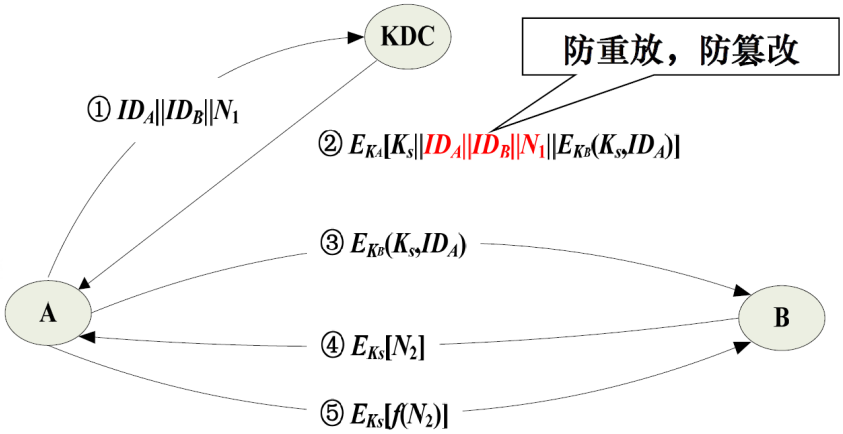
密钥管理与密钥分配

单钥加密体制的密钥分配

密钥分配的基本方法

- Alice 选取或生成 K_S并通过物理手段发送给 Bob
- 可信第三方 KDC(Key Distribution Center) 选取或生成 K_S 并通过物理手段将会话密钥发送给 Alice 和 Bob
- Alice 和 Bob 事先已有一共享密钥 K，其中一方选取或生成 K_S 后，用 K 加密 K_S 并发送给另一方
- Alice 和 Bob 分别与 KDC 享有一个保密信道，KDC 为 Alice 、Bob 选取或生成 K_S 后，再分别通过保密信道发给 Alice 和 Bob

NS密钥分配协议

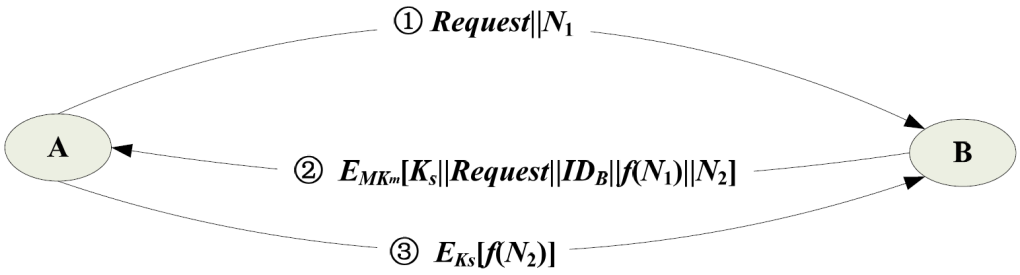


协议中可令 N_1 为随机数

第 ④、⑤ 两步，用于防止对第 ③ 步的重放攻击

- 假定敌手能获取旧的会话密钥，则在第 ③ 步中可冒充 Alice 向 Bob 重放旧的会话密钥，欺骗 Bob 使用旧会话密钥
- 敌手截获第 ④ 步中 Bob 发出的询问后，可假冒 Alice 作出第 ⑤ 步的应答
- 敌手可冒充 Alice 使用经认证过的旧会话密钥与 Bob 通信

无中心(KDC)的密钥分配



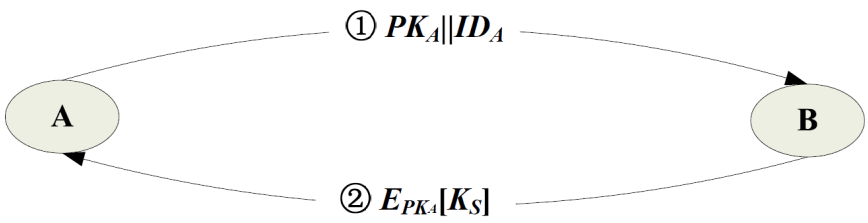
公钥加密体制的密钥管理

公钥分配

- 公开发布
- 公钥目录表
- 公钥管理机构
- 公钥证书
- IBC
- CL-PKC
- 自验证的公钥体制
- 一次性公钥分配

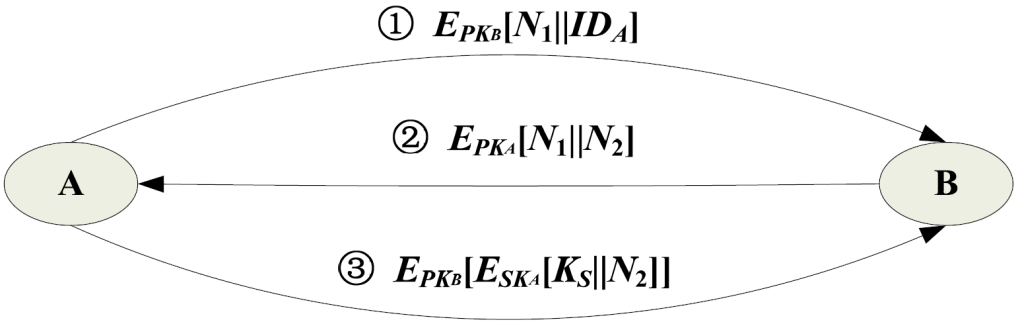
基于公钥加密的会话密钥分配协议

简单分配



- 中间人攻击
  - Alice 产生密钥对 $\{PK_A, SK_A\}$ 并向 Bob 发送 $PK_A||ID_A$
  - Eve 截获 Alice 的消息并建立自己的密钥对 $\{PK_E, SK_E\}$，并将 $PK_E||ID_A$ 发送给 Bob
  - Bob 产生会话密钥 K_S 后，将 $E_{PK_E}[K_S]$ 发送给 Alice
  - Eve 截获 Bob 的消息，由 $D_{PK_E}[E_{PK_E}[K_S]]$ 获得 K_S
  - Eve 再将 $E_{PK_A}[K_S]$ 发往 Alice

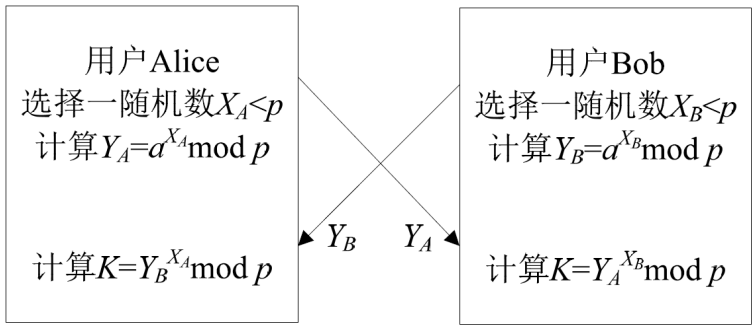
具有保密性和认证性的密钥分配



N_1 的存在使 Alice 相信对方的确是 Bob

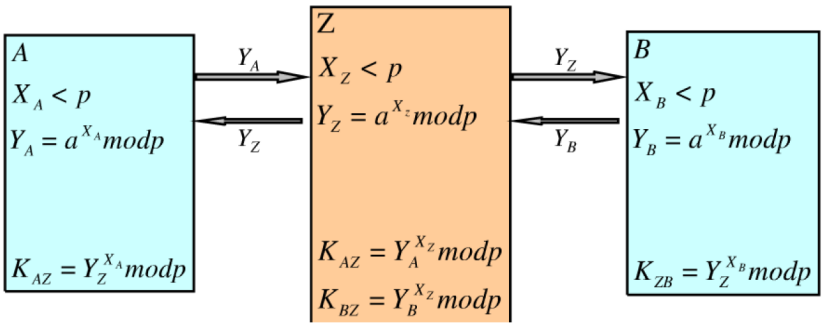
用 Bob 的公钥加密保证只有 Bob 能解读，用 Alice 的私钥加密保证该条消息只有 Alice 能生成

Diffie-Hellman密钥交换协议



a 是 p 的本原根

- 中间人攻击



随机数

随机数数列需满足的两个特性

- 随机性
- 不可预测性

伪随机数产生器

$X_{n+1} = aX_n + c \mod m$

基于分组密码算法的随机数产生器

- 循环加密
- DES/AES 输出反馈
- ANSI X9.17
- BBS

秘密分割

门限方案

设秘密 s 被分成 n 个部分信息，每一部分信息可称为一个子密钥或影子，由一个参与者持有，使得：

1. 获得大于等于 k 个参与者所持有的部分信息可重构 s
2. 获得少于 k 个参与者所持有的部分信息则无法重构 s

则称这种方案为 (k,n)-秘密分割门限方案，k 称为方案的门限值。当 $k=n$ 时，则需要所有用户参与合作才能恢复密钥

如果一个或一组未获授权的参与者在猜测秘密s时，并不比局外人猜测秘密有优势，则称方案是完善的

Shamir门限方案

Lagrange插值公式

已知 $\varphi(x)$ 在 k 个互不相同的点的函数值 $\varphi(x_i)(i=1,2,\cdots,k)$，可构造 $k-1$ 次 Lagrange 插值多项式

$$f(x)=\sum_{j=1}^{k}\varphi(x_j)\prod_{\substack{l=1\\l\neq j}}^{k}\frac{x-x_l}{x_j-x_l}$$

秘密分割

- $\mathrm{GF}(q)$ 是一有限域，其中 q 是素数，且满足 $q\geq n+1$
- 假设秘密为 s，令多项式常系数 a_0 等于 s
- 选取其它 $k-1$ 个系数 $a_i(i=1,\cdots,k-1)$
- $\mathrm{GF}(q)$ 上构造的 $k-1$ 次多项式记为 $f(x)=a_0+a_1x+\cdots+a_{k-1}x^{k-1}$
- n 个参与者记为 $P_1,P_2,\cdots,P_n$，P_i 的子密钥记为 $(i,f(i))$

秘密恢复

$$f(x)=\sum_{j=1}^{k}f(i_j)\prod_{\substack{l=1\\l\neq j}}^{k}\frac{x-i_l}{i_j-i_l}(\mod q)$$

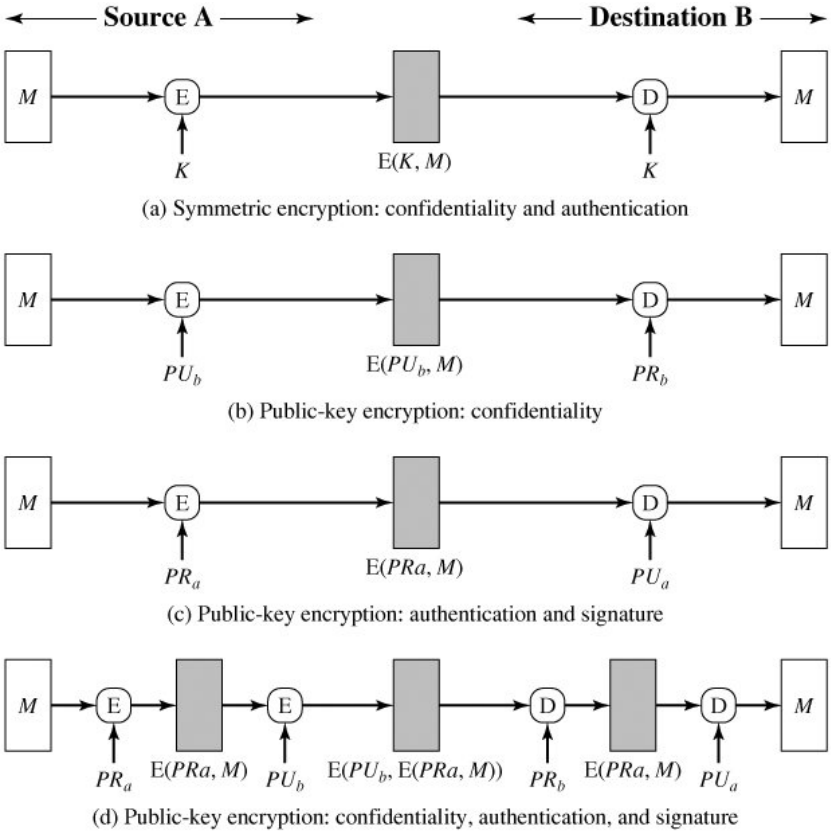
$$s=(-1)^{k-1}\sum_{j=1}^{k}f(i_j)\prod_{\substack{l=1\\l\neq j}}^{k}\frac{i_l}{i_j-i_l}(\mod q)$$

消息认证和哈希函数

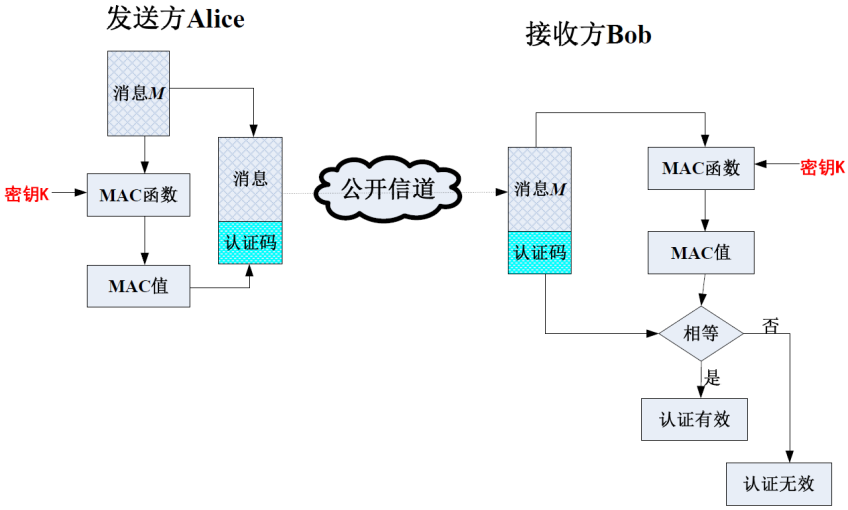
消息认证码

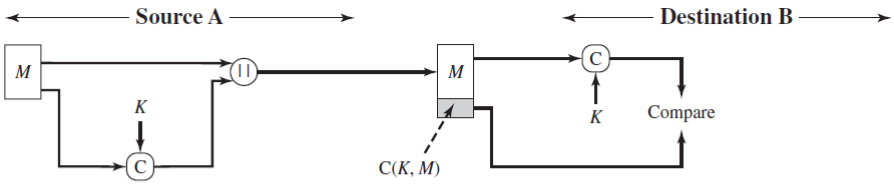
认证符的产生方式可分为如下三类

- 消息加密：对整个消息加密后得到的密文作为认证符
- 消息认证码：MAC(Message Authentication Code)
- 哈希函数(也称散列函数，hash function)：哈希值为认证符

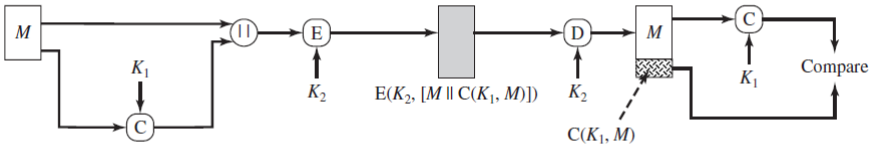


消息认证码指消息被一密钥控制的公开函数作用下产生的用作认证符的长度固定的数值，也称密码校验和

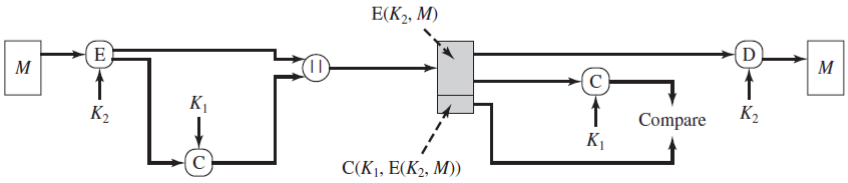




(a) Message authentication

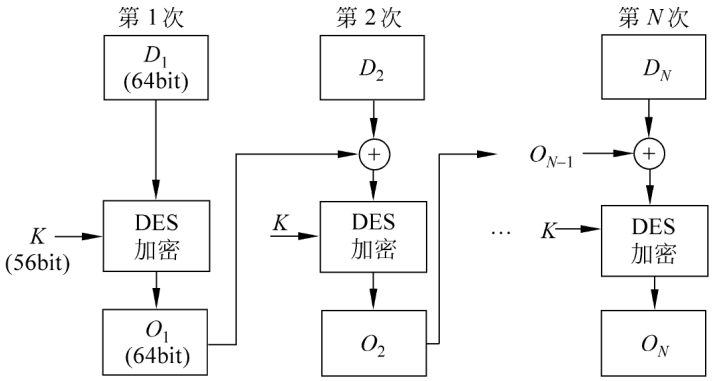


(b) Message authentication and confidentiality; authentication tied to plaintext



(c) Message authentication and confidentiality; authentication tied to ciphertext

数据认证算法DAA



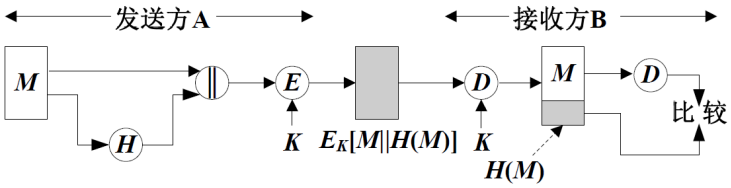
E 为 DES 加密算法，K 为密钥
消息认证码取为 O_N 或 O_N 的最左边 L 个比特，其中，$16 \leq L \leq 64$

哈希函数

定义

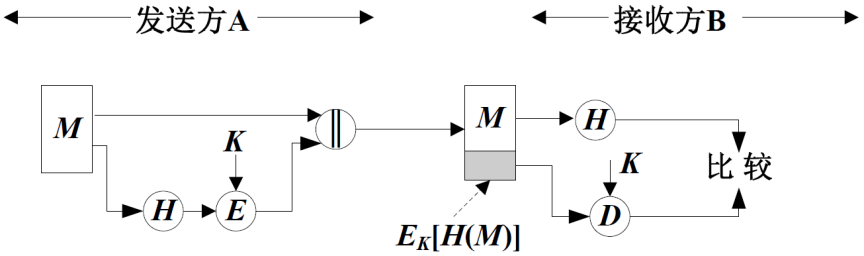
- 先 Hash 再单钥(对称)加密
  - 消息与哈希值链接后用单钥加密算法加密

    可提供消息的保密性、真实性和完整性



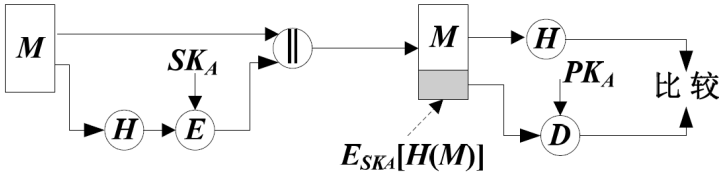
  - 用单钥加密算法仅对哈希值加密

    可提供消息的真实性和完整性



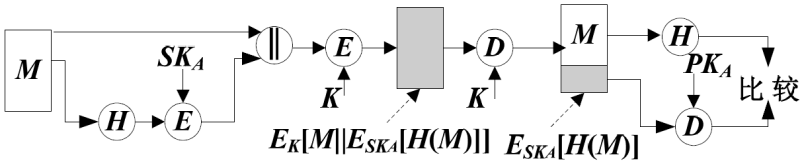
- 先 hash，再签名
  - 基于公钥加密算法用发方私钥仅加密(签名)哈希值

    提供了消息的真实性、完整性和发方不可否认性



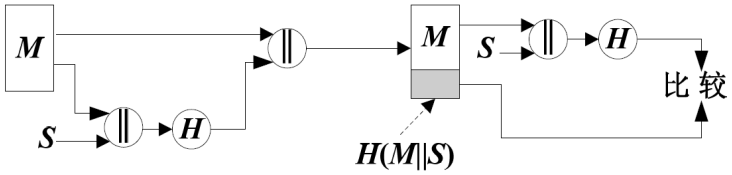
  - 上述方案再单钥加密

    提供了消息的保密性、真实性、完整性和发方不可否认性



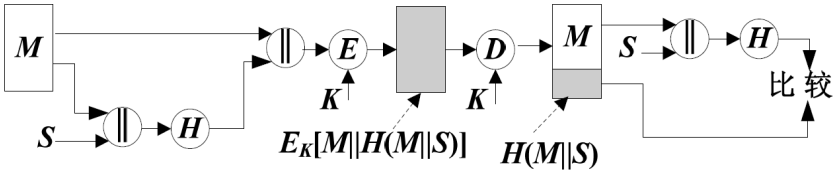
- 带密钥的 Hash (共享秘密值), 一种消息认证码
  - 发方计算消息 M 和秘密值 S 链接在一起的哈希值，作为消息 M 的认证码

    要求双方共享秘密值 S；提供消息的真实性与完整性



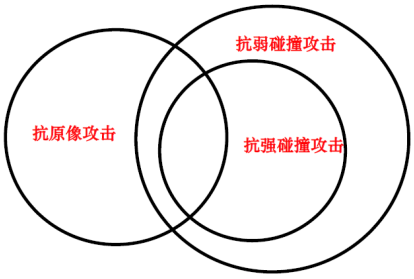
  - 上述方案中的消息和消息认证码再增加单钥加密运算

    提供了消息的保密性、真实性、完整性



哈希函数应满足的条件

- 函数的输入可任意长
- 函数的输出为固定长
- 已知 x，求 $H(x)$ 容易
- 已知 h，求满足 $H(x)=h$ 的 x 在计算上不可行，抗原像攻击
- 已知 x，找到 $y(y\neq x)$，使得 $H(y)=H(x)$ 在计算上不可行，抗弱碰撞攻击(第二原像攻击)
- 找出任意两个不同的输入 x 和 y，使得 $H(y)=H(x)$ 在计算上不可行，抗强碰撞攻击
- 伪随机性



生日攻击

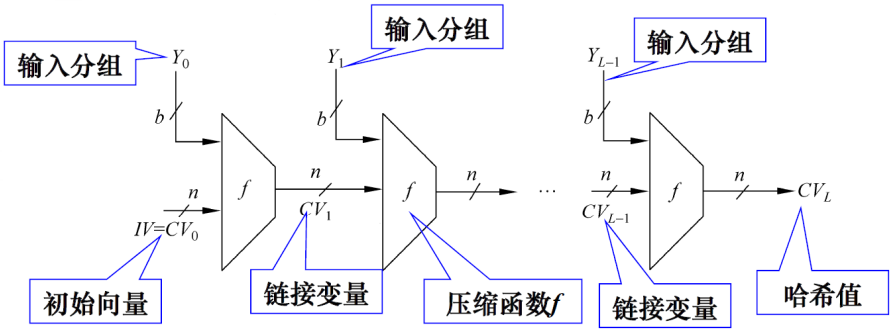
- 第I类生日攻击

  给定 Hash 值 $h = H(x)$，寻找 y 使得 $H(y) = H(x) = h$

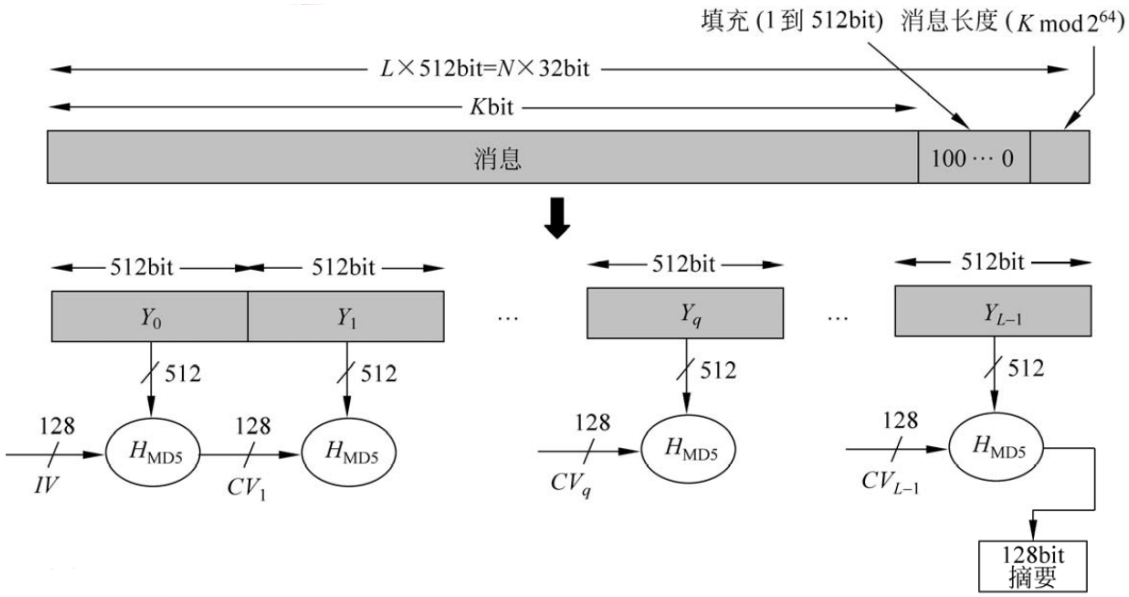
- 第II类生日攻击

  寻找 x 和 y 使得 $H(x) = H(y)$

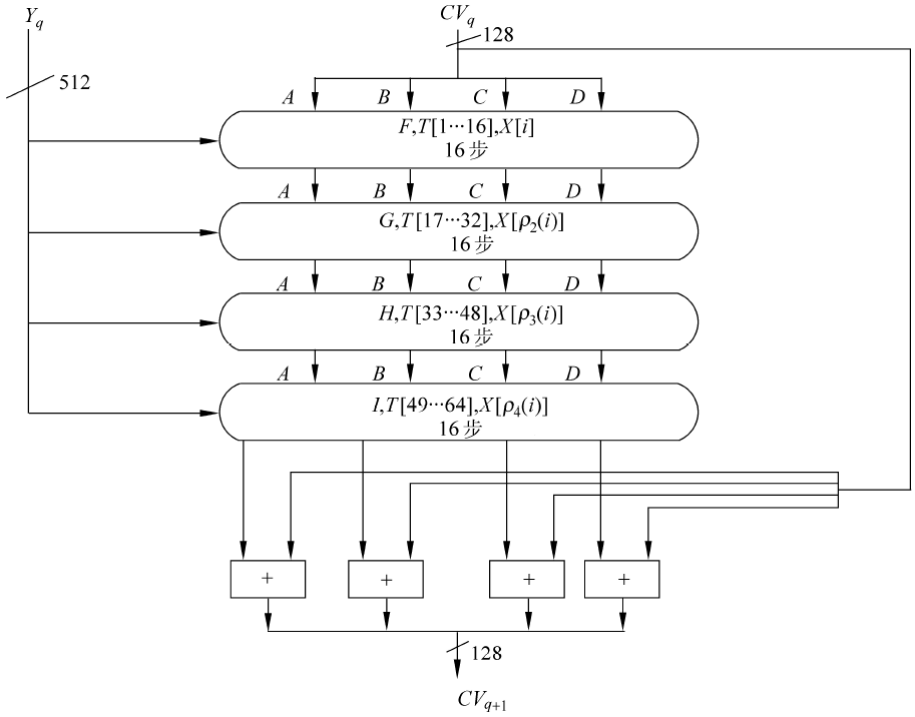
迭代型哈希函数的一般结构



MD5



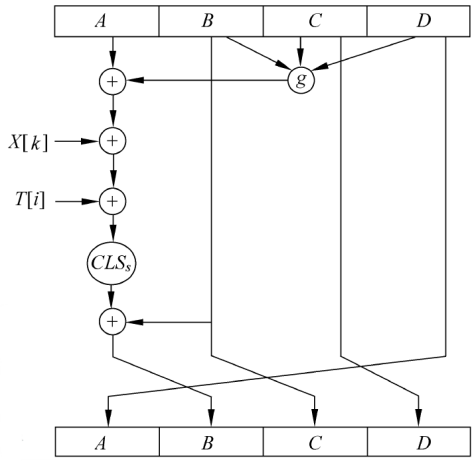
缓冲区表示为 4 个 32 比特长的寄存器 A、B、C、D，每个寄存器以little-endian方式存储数据，初值如下(以存储方式)：01234567、$89ABCDEF$、$FEDCBA98$、76543210。（实际值：67452301、$EFCDAB89$、$98BADCFE$、10325476）



$CV_0 = IV$

$CV_{q+1} = CV_q + RF_I\left[Y_q, RF_H\left[Y_q, RF_G\left[Y_q, RF_F\left[Y_q, CV_q\right]\right]\right]\right]$

$MD = CV_L$



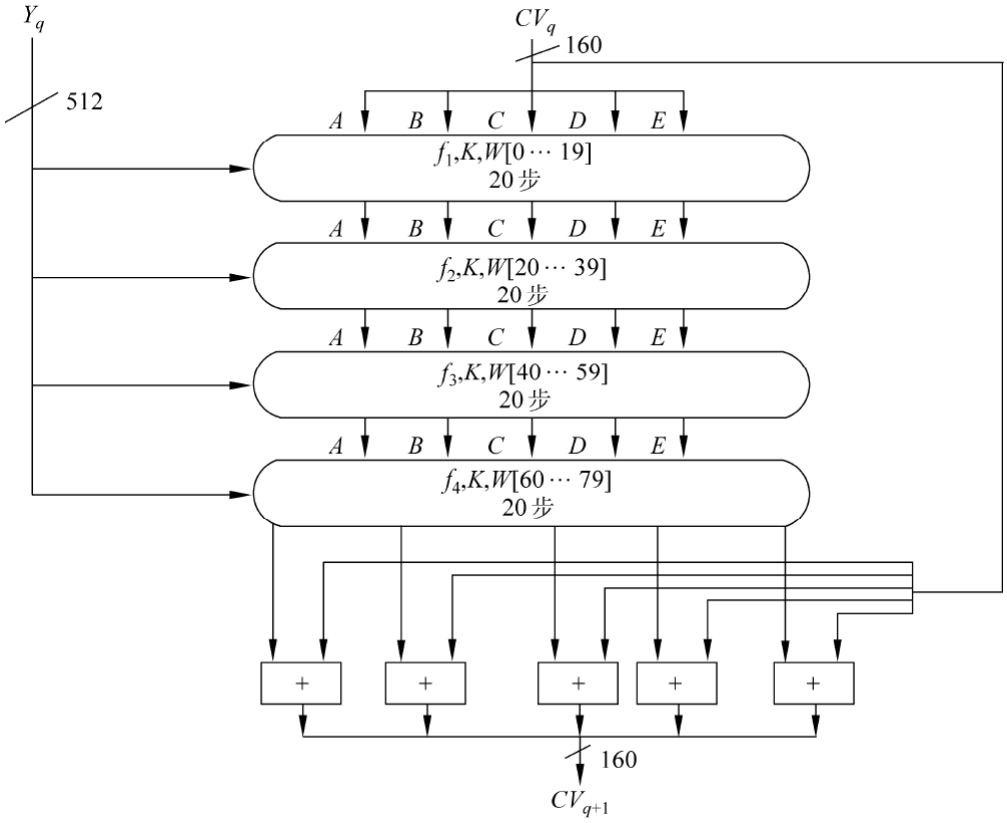
$A \leftarrow B + CLS_s\left(A + g(B,C,D) + X\left[k\right] + T\left[i\right]\right)$

CLS_s 是左循环移 s 位

g 是逻辑函数 F、G、H、I 之一

SHA-1

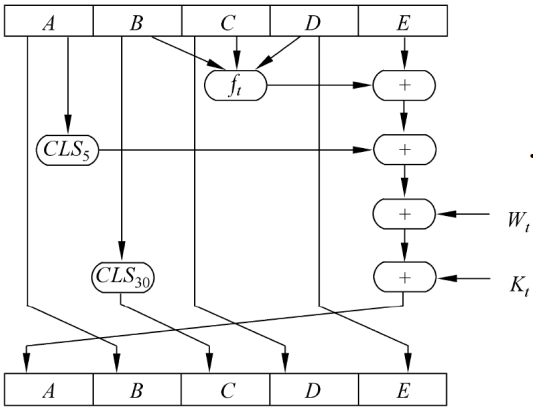
使用 160 比特长的缓冲区存储中间结果和最终哈希值，缓冲区表示为 5 个 32 比特寄存器(A、B、C、D、E)，寄存器以 big-endian 方式存储数据， 初始值 $A = 67452301$ 、$B = EFCDAB89$、$C = 98BADCFB$ 、$D = 10325476$、$E = C3D2E1F0$



$CV_0 = IV$

$CV_{q+1} = SUM_{32}(CV_q, ABCDE_q)$

$MD = CV_L$



$A, B, C, D, E \leftarrow (E + f_t(B, C, D) + CLS_5(A) + W_t + K_t), A, CLS_30(B), C, D$

数字签名算法

RSA签名

签名：$S \equiv M^d \mod n$，通常情况下 $S \equiv H(M)^d \mod n$

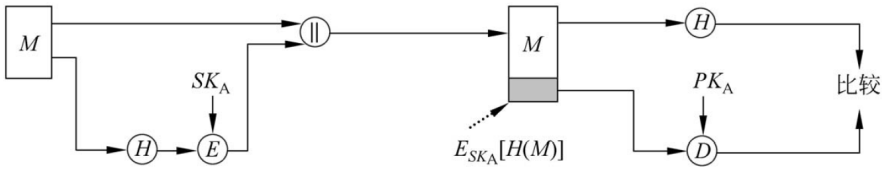
验证：$M \equiv S^e \mod n$

伪造方式

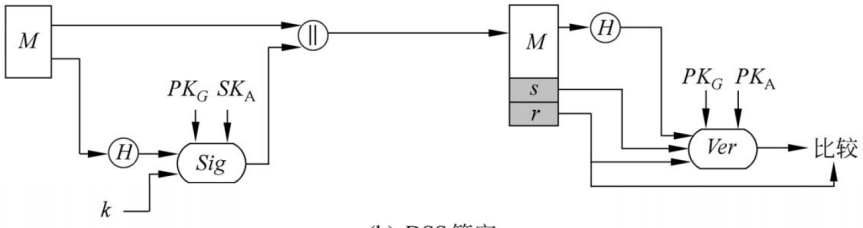
- 一般模式由公钥直接伪造签名
  - 攻击者任选数据 s 并用公钥 e 和 n 计算 $m' = s^e \mod n$，即 $(m')^d \mod n = s^{ed} \mod n = s$。则 (m', s) 就是一个可通过验证的伪造的签名
- 选择消息攻击模式
  - 攻击者选定消息 m_1 和 m_2，满足 $m = m_1 \times m_2$，并让签名者分别对 m_1 和 m_2 签名，$s_1 = m_1^d \mod n$ 和 $s_2 = m_2^d \mod n$
  - 计算 m 的签名 $s = s_1 \times s_2 = m_1^d \times m_2^d = (m_1 \times m_2)^d = m^d \mod n$
- 利用签名攻击获得明文
  - 假设攻击者获得密文 $c = m^e \mod n$，他要获得明文，于是选择一个小的随机数 r，计算 $s = r^e \mod n, l = s \times c \mod n, t = r^{-1} \mod n$
  - 因为 $s = r^e$，所以 $s^d = r \mod n$
  - 攻击者设法让签名者对 l 签名，于是得到 $k = l^d \mod n$，攻击者计算 $t \times k = r^{-1} \times l^d = r^{-1} \times s^d \times c^d = r^{-1} \times r \times c^d = c^d = m \mod n$ 获得明文 m
- 抵抗上述攻击的有效办法是对 hash 值进行签名

数字签名标准

DSS



(a) RSA 签字



(b) DSS 签字

DSA

- 全局公钥：p, q, g
- 用户私钥：x
- 用户公钥：$y \equiv g^x \mod p$
- 秘密数：k
- 签名：(r, s)
  - $r \equiv (g^k \mod p) \mod q$
  - $s \equiv \left[k^{-1}(H(M) + xr)\right] \mod q$，$H$ 为 SHA-1
- 验证：收到消息 M'，签名为 (r', s')
  - $w \equiv (s')^{-1} \mod q$
  - $u_1 \equiv [H(M')w] \mod q$
  - $u_2 \equiv r'w \mod q$

- $v \equiv [(g^{u_1} y^{u_2}) \mod p] \mod q$
- 检查 $v = r'$