

Lec. 02

密码学基本术语

M: 变换前的原始消息 (后文中也可称为 P)

C: 变换后的加密消息

K: 用于密码变换的只有发送者与接收者拥有的秘密消息.

$C = E(K_1, M)$: 使用特定密钥把明文转化为密文的数学方法.

$M = D(K_2, C)$: 使用特定密钥把密文转化为明文的数学方法.

密码系统是一个五元组 $(M, C, K, E(\cdot), D(\cdot))$ 满足下列条件.

M 是可能明文的有限集 (明文空间)

C 是可能密文的有限集 (密文空间)

K 是一切密钥构成的有限集 (密钥空间)

加密解密算法满足以下条件: $E(M, K_1) = C$ 则 $D(C, K_2) = M$

即 $D(E(M, K_1), K_2) = M$.

密码体制分单钥体制与多钥体制两类。

对称算法 非对称算法. 解密密钥不可以从加密密钥中得出

$K_1 = K_2$, 或可互相推导 私钥 SK 公钥 PK(可公开)

密码分析 在不知道密钥情况下, 恢复出明文的科学 (攻击)

攻击密码的方法包括穷举法与分析法. 分析法可分为如下几类

唯密文攻击: 从已知密文恢复出明文或密钥

已知明文攻击: 从已知密文和明文-密文对中分析明文

选择明文攻击: 可选定任意明文-密文对进行分析(攻击)

选择密文攻击: 分析者可选择不同的被加密的密文, 并得到对应的解密的原文 (主要用于公钥算法)

Kerckhoff假设: 攻击者知道正在使用的密码体制. 即一个密码系统的安全性都应基于密钥的安全性, 而不应该基于算法细节的安全性.

- 原因：易于保存、分享，防止反向工程，易更换
- 结论：公开的密码学设计，可经受更多实践检验，易于发现漏洞，避免反向工程的危害，同时利于构建标准。

信息论 (Shannon, 1949 / 1950)

建立了通信保密/密码学严格的理论基础。

证明了一次一密 (one-time pad, OTP) 系统是完善保密的，导致了对流密码研究应用提出分组密码设计准则（扩散性 & 混淆性）
（越大，越难以破译）

证明了消息冗余使破译者统计分析成功的理论值（唯一解距离）减小。

信息熵 H 是概率的单调递减函数。

无条件安全（完善保密） 无论有多少可使用的密文，都不足以唯一确定由该体制产生密文所对应的明文。无论花多少时间，攻击者都无法将其解密。

在唯一密文攻击情况下，如果一个密码系统其密文与明文之间的相互信息为 0，即 $I(M:C)=0$ ，则该系统是完善保密的，其必要条件为 $H(K) \geq H(M)$ 。

无条件安全密码体制是存在的 (OTP)。

计算安全 破译密文代价超过了密文信息的价值

（或）破译密文的时间超过了密文信息的有效生命周期。

古典密码基本技术

- ① 代换：将明文字母替换为其他字母/数字/符号的方法；如果明文被看作二进制比特串，则用密文比特串代替原比特串。
- ② 置换：对明文字母的某种置换取得一种类型完全不同的映射
(换位) (不改变明文，改变其字母顺序)

古典加密技术范例

1. Caesar 密码 最早代换密码 (每个字母用其后 3 个字母替代)

△ 不安全

$$C = E(M) = (M + 3) \bmod 26$$

$$M = D(C) = (C - 3) \bmod 26$$

密码分析 (穷举) 密码空间仅 25 个元素，加解密算法已知，明文具可读性。

2. 移位密码 每个字母用其后 K ($0 \leq K \leq 25$) 个字母替代 ($K=3$ 即 Caesar 密码)

△不安全

$$C = E(M) = (M + K) \bmod 26$$

$$M = D(C) = (M - K) \bmod 26$$

密码分析 (穷举) 密码空间仅 25 个元素，加解密算法已知，明文具可读性。

结论：安全的密码体制密钥空间应足够大。

3. 单表代换密码 密钥 K 是一个代换表，长度为 26，进行字符空间的置换。

△不安全

$$C = E(M) = P(M) \quad \text{其中 } P \text{ 为字符空间的置换}$$

$$M = D(C) = P^{-1}(C)$$

密码分析 (穷举) 密钥空间 26！过大，不可行。

(语言分析) 语言统计特性：人类语言是有冗余度的，字母使用频率不同，如 $E > T > A > O > N > I > S > \dots$

问题：明文的统计特性被原样传到密文中。

改进：一次加密多个字符 / 多表代换

4. 仿射密码：代换密码特殊情形。此时 $P(x) = (ax + b) \bmod 26$

△不安全

需满足 $(a, 26) = 1$

加密： $C = P(M) = (aM + b) \bmod 26$

解密： $M = P^{-1}(C) = a^{-1}(C - b) \bmod 26$ 其中 a^{-1} 为 a 在 26 下乘法逆元。

密码分析 (穷举) 密钥空间仅 $\underline{12} \times \underline{26}$ 。

5. Playfair 密码 多表代换密码，一次加密两个字符。(1984)

△不安全

选择无重复字符密钥词，构造 5×5 矩阵，按顺序先填密钥词，再将其余字符填入。(I, J 合占 1 格)

填充

加密：预处理明文(分组 + 同字母填充 + 重分组) $Balloon \rightarrow Ba \downarrow x \downarrow l o \downarrow on$

若一组两字符同行/列，则用该行/列后一个字符替代

非同行同列字母用对角两个字母替代

解密: 加密的反向操作 + 人工识别插入字符与 i/j.

密码分析 (语言分析) 虽然频率分析变得困难, 但双字母频率分析仍可行.

6. **Hill 密码** 多表代换密码. 一次加密多个字符. (1929) m 个连续明文字符被 m 个连续密文字符代替, 由 m 个线性方程决定替代方法. 即

(对已知明文攻击)

$$\begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & \dots & K_{1m} \\ K_{21} & K_{22} & \dots & K_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ K_{m1} & K_{m2} & \dots & K_{mm} \end{pmatrix} \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_m \end{pmatrix} \pmod{26}$$

密钥: $K = (k_{ij})_{m \times m}$

$$C = KM, \quad M = K^{-1}C$$

故需 K 的逆矩阵 K^{-1} 使 $KK^{-1} = I_m$

密码分析 优点: 完全掩盖频率特性, 可抵抗已知密文攻击

缺点: 不能抵抗已知明文攻击, m 个明文-密文对即可解出 K .

7. **Vigenère Cipher** 使用多个单字母替换表, 因此一个字母可被多个字母替换

(**Vernam 密码**) 密钥的第 i 个字符确定加解密使用第 i 个字母表. 依次使用每个字母表. 密钥周期使用. 故

$$C_i = (M_i + K_{i \bmod r}) \pmod{26} \text{ 其中 } r \text{ 为密钥长度}$$

密码分析: 每隔 r 的整数倍, 出现单表代换. 因此在确定 r 的大小后, 通过破解单表代换即可. 无法抵御频率攻击.

Kasiski 方法 令 $I(x)$ 表示串 x 中两个随机元素相同的概率

如果 x 为英文文本, 有 $I(x) \approx 0.065$. 若 x 完全随机, $I(x) \approx 0.038$

于是即可通过枚举 r 后计算每段的重合指数来找到 r

8. **一次一密** (One-Time Pad, OTP) 基于二进制数据 注: 2TP 不安全

⑤安全 $M = m_1 m_2 \dots m_n \quad K = k_1 k_2 \dots k_n \quad C = c_1 c_2 \dots c_n \quad \text{其中 } c_i = m_i \oplus k_i$

(改进) 密钥随机串服从均匀分布, 与明文等长, 每次加密均随机新密钥.

密码分析: 复用 K 的加密无法抵御已知明文攻击, 而改进后是完善保密的

9. 置换密码 对明文字母的某种置换取得一种类型完全不同的映射.

△不安全 即打乱明文顺序. 如 are \rightarrow rea. 置换 $\Pi = \{3, 1, 2\}$.

密码分析: 字母频率没有改变. 且不能抵御已知明文攻击.

10. 栅栏加密 将明文以对角线方式写为若干行. 再按行顺序读出(特殊的置换密码)

△不安全 如 meetmehere \rightarrow $\begin{matrix} m & e & m & h & r \\ e & t & e & e & e \end{matrix}$ ($l=2$) \rightarrow memhretee

密码分析: 事实上, 密钥空间大小仅为明文长度 $|M|$. 穷举即可

11. 换位技术 将明文写成矩阵块. 再按列读出. 但列顺序打乱, 密钥 K 即列次序.

△不安全 例: $K = \begin{matrix} 3 & 4 & 2 & 1 & 5 & 6 & 7 \end{matrix}$

$M = \text{attack postpone.} \Rightarrow \begin{matrix} a & t & t & a & c & k & p \\ o & s & t & p & o & n & e \end{matrix} \Rightarrow \text{ttaptsaoocoknpe}$

密码分析: 字母频率没有改变. 且不能抵御已知明文攻击. 对简单换位密码, 可先写成矩阵块再推断列次序

乘积密码 由于单独使用代替/置换均不安全(语言有冗余性与特定统计概率), 故若考虑连续使用几种密码来增加破译困难性.

连接古典加密与对称加密的桥梁.

其他例子	转轮机(Enigma)	复杂的代替密码
	达芬奇密码(密码筒)	复杂的代替密码
	隐写术	并不算加密. 需大量数据掩盖少量数据 e.g. 固像 LSB 隐藏信息, IoT 的 Smart Config 技术

Lec. 03

对称加密 包括 { 分组密码 信息被分组加密解密，连读明文元素使用相同 K 加密。

(包括现有
大量密码，
应用广阔) $C = C_1 C_2 \dots = E_K(m_1) E_K(m_2) \dots$ 如 SM4. 三重加密 ...
可看作分组长度大的代替。

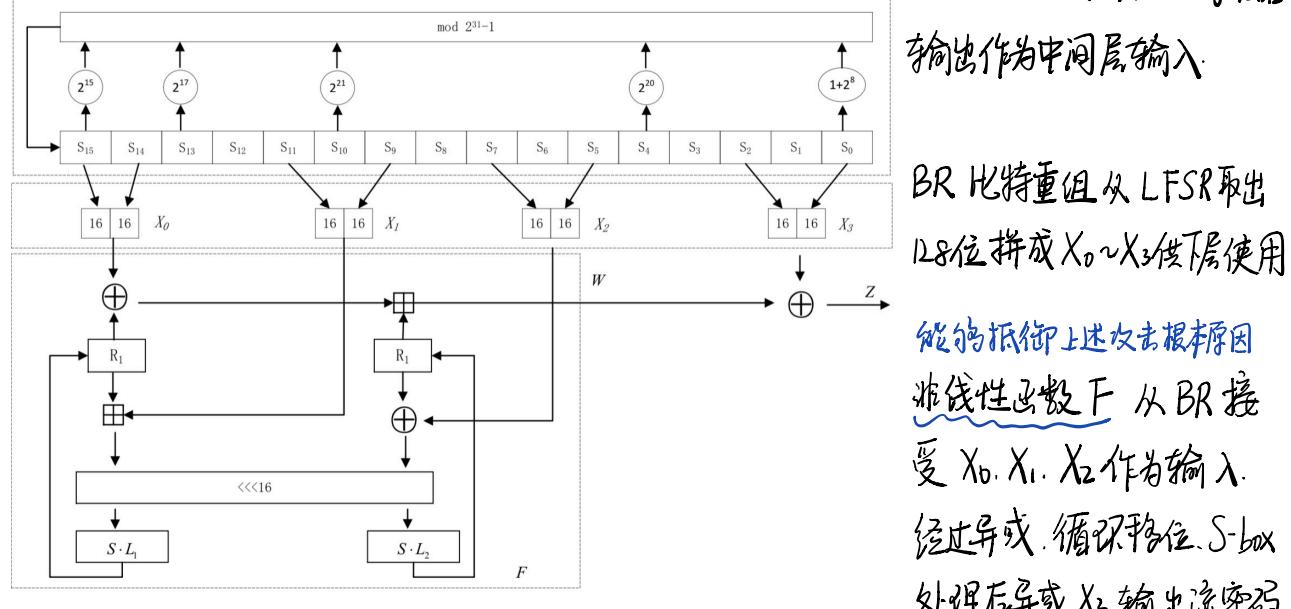
流密码 产生密钥流 $K = k_1 k_2 \dots$ 对明文流 $m_1 m_2 \dots$ 生成密文流

$C_1 C_2 \dots$ 其中 $C_i = E_{k_i}(m_i)$ 如 ZUC. RC4. SNOW ...

流密码一般以 bit / byte 为单位

祖冲之密码 (ZUC) 128 bit 初始密钥；128 bit 初始向量 共同决定 $S_0 \sim S_{15}$ 初始值。

为什么不能单独使用 LFSR？设依赖长度 n ，则连续收集 $2n$ 数据即可破译 LFSR 线性反馈移位寄存器



分组密码 原理：具有参数 K 的可逆变换。 $C = E(M, K)$ $M = E^{-1}(C, K)$

应用：实质为消息分组后“替代密码”，可提供安全性/认证性的安全服务。

分组大小：小 → 安全性差；大 → 实现困难 (混淆扩散, Shannon)

设计准则：可逆变换是必要条件，使用 乘积密码逼近代换函数 (Feistel)

设计安全密码 → 轮数多 → 慢 (权衡取舍)

代换置换网络 (SPN) 现代分组加密的基础，其中

S-box. 代换 (substitution)

P-box 置换 (permutation)

对消息及密钥进行

混淆与扩散。

混淆: 使密钥与密文之间统计关系尽可能复杂, 防止攻击者发现密钥.

扩散: 使明文的统计特性分散在密文中, 让每个明文影响尽可能多密文.

使密文与明文间统计关系尽可能复杂..

Feistel 密码: 基于可逆的乘积密码.

加密: 输入分组分为左右两半 LE_i, RE_i , 执行多轮迭代 (可加入初始置换 IP 与逆
初始置换 IP⁻¹ 不影响)

$$LE_i = RE_{i-1} \quad RE_i = LE_{i-1} \oplus F(RE_{i-1}, K_i)$$

其中 K_i 为第 i 轮轮密钥, F 为轮函数 (过程中不变)

n 轮迭代后令 $LE_{n+1} = RE_n, RE_{n+1} = LE_n$, 则 $(LE_{n+1}, RE_{n+1}) = C$.

解密: 与加密结构完全相同, 以密文为输入, 逆序使用轮密钥即可

加密 i 轮与解密 $r-i$ 轮对应 (r 为总轮数). 即

$$LD_{r-i} \parallel RD_{r-i} = RE_i \parallel LE_i$$

设计
元素

{ 分组大小 分组越长安全性越高, 但会降低加解密速度. 64 位合理

轮数 多轮可取得很高安全性. 一般取 16

密钥长度 密钥越长安全性越高, 但会降低加解密速度. 通常取 128.

子密钥产生算法 越复杂, 密码分析越困难

轮函数 越复杂, 抗攻击能力越强.

简化分析难度. 描述清晰, 易于分析, 可以开发更强算法

快速软件加解密 加解密速度较快

简化 DES 8位明文组, 8位密文组, 10位密钥. 仅有教学意义.

加密 $M = IP^{-1} \circ F_{K_r} \circ SW \circ F_{K_{r-1}} \circ SW \circ \dots \circ F_{K_1} \circ IP(C)$

其中 $F_K(L \parallel R) = (L \oplus F(R, K), R)$ (此处 IP 不影响, 可忽略)

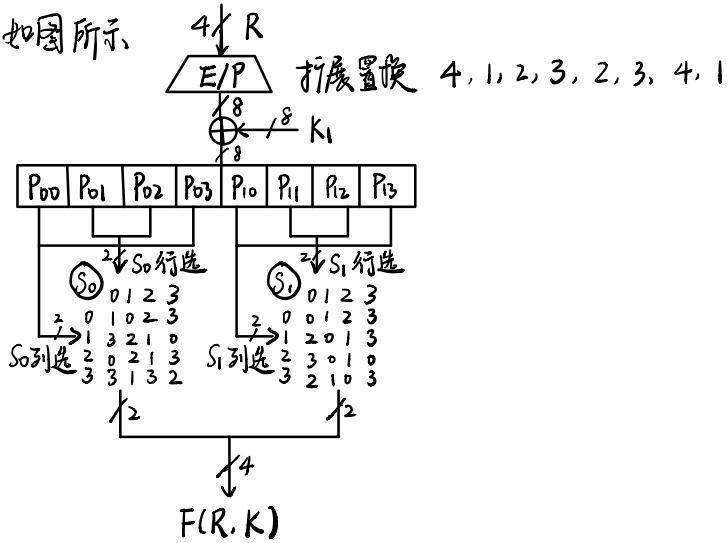
故 $SW \circ F_{K_i}(L \parallel R) = (R, L \oplus F(R, S_k))$ 即 Feistel 结构.

轮密钥 $K_1 = P_8(Shift_1(P_{10}(K)))$

$K_2 = P_8(Shift_1(Shift_1(P_{10}(K))))$

其中 $Shift_t(\cdot)$ 表示循环左移 t 位, $P_t(\cdot)$ $P_{10}(\cdot)$ 为特定长度置换

轮函数 $F(R, K)$ 如图所示



密钥分析 (穷举). 密钥空间 $2^{10} = 1024$. 平均试验 $2^9 = 512$ 次.

△不安全 (已知明文攻击) 仅 S-box 含非线性成分, 故每个明文-密文对可列出 8 个非线性等式. 每个等式有 10 个未知量. 含 10 个未知项的二进制多项式 (二进制多项式中) 可能项为 $1 + \binom{1}{10} + \binom{2}{10} + \dots + \binom{10}{10} = (1+1)^{10}$. 于是平均意义上含有 2^9 个项, 故需 2^9 个明文-密文对. 与穷举代价相当。

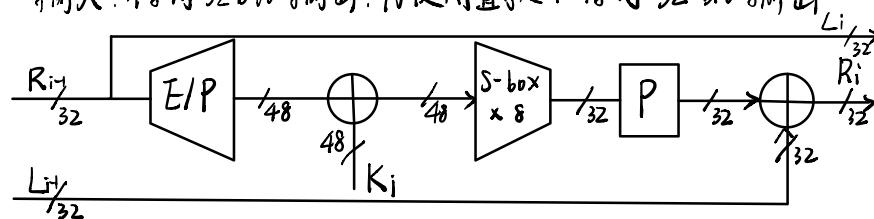
DES (数据加密标准) 1977. NBS 公布数据加密标准. 最成功的商用密码算法.

一般描述. 输入明文 64 bit; 密钥长度 56 bit; 密文长度 64 bit. 轮数 16 轮. 与 Feistel 密钥结构完全相同.

初始置换 IP: 对输入数据重排序. 偶数位在上半边. 奇数位在下半边. 结构非常有序. 作用为使问题表述一致.

轮函数 F: 64 位中间数据被分为两个 32-bit 的部分: L & R.

下输入为 32 bit R 与 48 bit 轮密钥。每轮使用拓展置换将 R 扩展为 48 bit, 再与 48 bit 轮密钥相加 (异或), 将所得作为 8 个 S-box 输入. 得到 32 bit 输出. 再使用置换 P 得到 32 bit 输出.

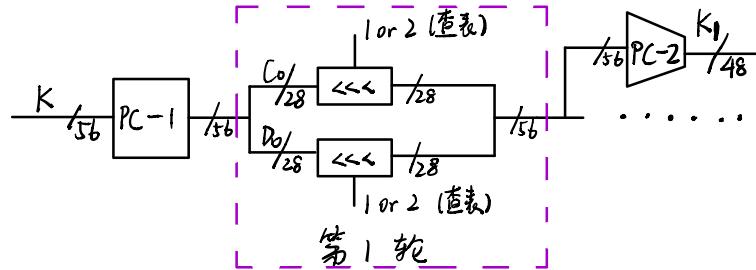


- * 扩展置换中，32 bit 输入分为 $4 \text{ bit} \times 8$ 组，每组取相邻两组靠近 1 bit $\Rightarrow 6 \text{ bit}$
- * 每个 S-box 将 6 bit 输入映射为 4 bit 输出。实质上每个 S-box 是 4 个 6 bit S-box。第 2、3、4、5 个 bit 确定 S-box 列，第 1、6 个 bit 确定 S-box 的行。用行列交叉处的 S-box 值替换即可。

密钥产生 56 bit 密钥经第 1 轮置换选择 PC-1，分为 28 bit 两部分 C_0, D_0

通过 16 轮迭代，形成每一轮子密钥，每轮迭代中

- 依据 密钥循环表， C_i 与 D_i 分别循环左移 1 位 / 2 位。
- 移位后的 56-bit 作为下轮输入，同时作为压缩置换 PC₂ 输入。
- PC₂ 从左右各选 24 bit 得到 48 bit 作为轮函数 F 输入。



解密 由于 Feistel 结构，解密步骤与加密步骤一致。

密钥空间 2^{56} ，但仍远大于理想分组密码密钥空间 2^{64} ！。

密码分析 (穷举) $2^{56} \sim 10^{16}$ ，故暴力破解可行。

△不安全 要求：明文可识别，对数字文件的加密，很难实现自动化破译。

(计时攻击) 对 DES 无效。

收集加密中信息
可恢复全部/部分
轮密钥，若必
要，可对剩余密
钥进行穷搜索

{ (差分密码分析) 2^{47} 对选择明文密文，对 DES 需 $2^{55.1}$ 操作 故无效
(线性密码分析) 2^{43} 对已知明文密文 (尝试在输入输出 bit 间建立)
(相关密钥攻击) 明文不动，动密钥 (根据线性方程，解之可分析)

构造好的 S-box ① 随机生成，测试特性，手工修正。| 对 DES 质疑所在。
② 基于数学理论，数理方法构造。| 没有给出理由，不透明

雪崩效应 明文或密钥的微小变化将对密文产生很大影响，是加密算法核心需求。

Avalanche Effect 要求输入或密钥的 1 比特变化大约引起输出比特一半的变化。

避免给密码分析者提供缩小密钥空间或明文空间的渠道.

DES具有严格雪崩效应(SAE). $\forall i, j$. 若 S-box i -位发生变化. 输出位 j 变化概率为 $\frac{1}{2}$; 对于一般非线性函数, 也可有 SAE.

计时攻击 某些加密解密算法对不同输入所用时间有细微差别. 可利用这个差别进行攻击从而还原输入. 在 smartcard 上问题严重

位独立准则 (BIC) $\forall i, j, k$. 当输入 i -bit 变化时. 输出 j -bit, k -bit 应独立变化.
DES 具有位独立准则. 古典密码一般不具备 BIC, SAE.

密钥调度 推测子密钥和主密钥难度尽可能大

密码设计的评价 好: 雪崩效应. 位独立特性. 不可预料性

差: 缺乏随机性. 具有太大可预料性.

最好的方法是测试密码以发现漏洞

Lec. 04

双重 DES: 使用两次密钥不同的 DES 加密方法，即

$$C = E_{K_2}[E_{K_1}(M)] \quad M = D_{K_1}[D_{K_2}(M)]$$

与 DES 的关系：双重 DES 中对应的映射大部分不被单个 DES 所定义。即在大部分情况下，不存在 K_3 使 $E_{K_2}[E_{K_1}(M)] = E_{K_3}(M)$

密码分析 中间相遇攻击

描述 根据 $C = E_{K_2}[E_{K_1}(M)]$ 有 $D_{K_2}(C) = E_{K_1}(M) = X$ 。给定一个明文密文对

△不安全

(M, C)。将 M 用全部 2^{56} 个密钥 K_1 加密后按 X 排序存入一张表中，再对 C 用全部 2^{56} 个密钥 K_2 解密后，通过 X 在表中查询对应的 K 得到若干可能密钥对 (K_1, K_2) 。再用若干明文-密文对验证所有候选密钥对即可找出对应密钥 (K_1^*, K_2^*)

分析 双重 DES 中密文空间长度 2^{64} ，密钥空间长度 2^{112} ，故平均第一轮结束后有 $2^{112}/2^{64} = 2^{48}$ 个可能的密钥对 (K_1, K_2) 。第二轮时，用候选密钥对对新明文-密文对 (M', C') 加密时，若所得结果为 C' ，则认为 (K_1, K_2) 正确。故出错概率约为 $2^{48}/2^{64} = 2^{-16}$

结论 使用中间相遇攻击，在已知两组明文密文时，付出数量级为 2^{56} 的代价，可以成功攻击密钥尺寸为 2^{112} 的双重 DES，而普通 DES 攻击代价为 2^{55} 。故 **双重 DES 不能提高 DES 安全性**。

三重 DES: 为对抗中间相遇攻击，使用三个密钥进行三次 DES 加密，其广泛代（3DES）替了 DES。用于密钥管理标准 ANSI X9.17 与 ISO 8732。其弱点为密钥长度 168 位，过长。
目的: 与 DES 兼容，令 $K_1 = K_2$ 即可

改进 (Tuchman) 用两个密钥进行三次加密，即 加密-解密-加密 (EDE)

$$\text{加解密 } C = E_{K_1}[D_{K_2}[E_{K_1}(M)]] \quad M = D_{K_1}[E_{K_2}[D_{K_1}(C)]]$$

有研究者对双密钥 3DES 不放心，推荐使用三密钥 EDE 结构的 3DES。
目前三密钥 3DES 应用于 PGP、S/MIME 等协议/场景中

密码分析 目前对 3DES 没有可行攻击方法。

(**安全**) 平均代价 2^{111} , 为 10^{33} 数量级

(**Merkle-Hellman 攻击**) 不实际, 需 2^{56} 个特定明文密文对.

- 随机选择 A . 为方便一般选 $A=0$
- 对于所有 2^{56} 个 K_1 , 由 $P = D_{K_1}(A)$ 得到 2^{56} 个明文
- 通过 3DES 得到这 2^{56} 个明文 P 对应的 2^{56} 个密文 C .
- 通过 $B = D_{K_1}(C)$ 得到 2^{56} 个中间结果 B 并按 B 排序后存入表中。 ($P \xrightarrow{E_{K_1}} A = 0 \xrightarrow{D_{K_2}} B \xrightarrow{E_{K_1}} C$)
- 对 $A=0$ 用全部 2^{56} 个密钥解密, $B = D_{K_2}(0)$, 并在表中查找对应 K_1 得 (K_1, K_2) 对.
- 对所有 (K_1, K_2) 可能对, 另取明密文对 (P', C') 验证即可得到密钥.

(**已知明文攻击**): 给定几个已知明文、密文对解得密码, 但运算数量级约 $2^{120-\log n}$, 也不实际。

优点 密钥长度 112 bit 足够长, 底层算法与 DES 相同, 故受密码分析时间远远长于其他加密算法, 对密码分析有较强免疫力。仅若考虑安全性, 3DES 会成为未来数十年加密算法的合适选择。

缺点 DES 设计主要针对硬件, 相对于今天多数软件应用场合效率低, 分组长度 64 似乎应更长; 故 3DES 不能成为长期使用的加密标准。

IDEA (国际数据加密算法, 专利) 应用于 PGP 等一些专门应用。

描述 IDEA 是一个分组长度 64 位, 密钥长度 128 位的分组密码算法。64 位明文分成 4 块, X_1, X_2, X_3, X_4 经 8 轮迭代。可利用同一算法加解密, 可以抵抗差分攻击。

运算 \oplus 按位异或. 田 模 2^{16} 整数加 \odot 模 $2^{16}+1$ 整数乘
IDEA 中“混淆”与“扩散”的来源

速度 比 DES 快 2 倍，且 4 轮 IDEA 已足够快。

密码分析 (穷举) 可抗穷举攻击，代价高达 2^{127} 。

① 安全

(其他) 目前比穷举更快的攻击仅对 2.5 轮以下有效。

(差分攻击) Lai 证明了 IDEA 第四轮以后对差分免疫。

(密钥选择) 存在极少量 ($1/2^{16}$) 弱密钥，且易被修订。

AES 来源 1997 年 NIST 征集下一代加密算法，要求：① 比 3DES 快；
② 至少与 3DES 一样安全；③ 分组长度 128 bit；④ 支持密钥长度为 128/192/256 bit；2000 年，Rijndael 算法被选中，即 AES。

* NIST 评估标准：安全性、成本（软硬件、存储空间）、算法与执行特征、灵活性（密钥长度、分组长度、轮数）、ILP 潜力等。

结构

分组长度 128 bit，密钥长度不同所对应轮数不同

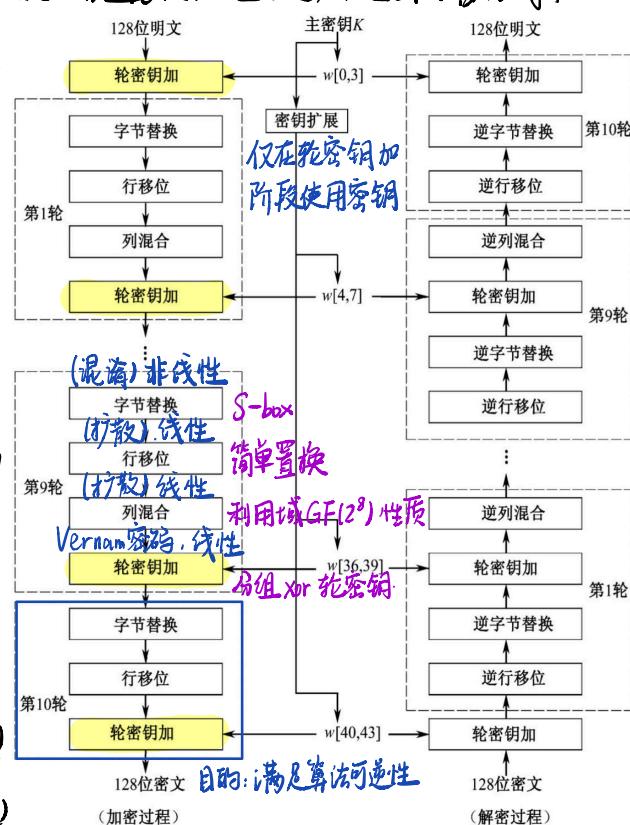
$$\begin{cases} 128 \text{ bit key} \Rightarrow 10 \text{ 轮} \\ 192 \text{ bit key} \Rightarrow 12 \text{ 轮} \\ 256 \text{ bit key} \Rightarrow 14 \text{ 轮} \end{cases}$$

非 Feistel 结构，加解密算法不同，每个阶段均可逆，代数结构清晰。

状态 用矩阵列向量表示，每列为一个 4 字节 (32 bit)，则所需

$$\text{列数} = \frac{n}{32} \quad (n \text{ 为分组长度})$$

$$\text{密钥列数} = \frac{k}{32} \quad (k \text{ 为密钥长度})$$



目的：满足算法可逆性

128 位密文
(加密过程)

128 位密文
(解密过程)

in ₀	in ₄	in ₈	in ₁₂
in ₁	in ₅	in ₉	in ₁₃
in ₂	in ₆	in ₁₀	in ₁₄
in ₃	in ₇	in ₁₁	in ₁₅

(Input)

S _{0,0}	S _{0,1}	S _{0,2}	S _{0,3}
S _{1,0}	S _{1,1}	S _{1,2}	S _{1,3}
S _{2,0}	S _{2,1}	S _{2,2}	S _{2,3}
S _{3,0}	S _{3,1}	S _{3,2}	S _{3,3}

(State)

out _{0,0}	out _{0,1}	out _{0,2}	out _{0,3}
out _{1,0}	out _{1,1}	out _{1,2}	out _{1,3}
out _{2,0}	out _{2,1}	out _{2,2}	out _{2,3}
out _{3,0}	out _{3,1}	out _{3,2}	out _{3,3}

(Output)

本例中，

$$n = k = 128$$

ByteSub 字节替换 (非线性) 将每个状态字节在 $GF(2^8)$ 下求逆 ($0 \rightarrow 0$)，然后对字节中的 8 个 bit 进行仿射变换。也可以简单等价于字节置换表 S-box (计算得出)

Shift Row 行移位 (线性) 将状态的最后三行移位不同的位移量。各层位移量与 N_b 有关。本例中，第 2 层循环左移 1 位，第 3 层循环左移 2 位，第 4 层循环左移 3 位。

Mix Column 列混合 (线性) 将矩阵列圈左乘一个固定矩阵 C。若把矩阵行看作多项式

$$C(x) = C_{0,0}x^3 + C_{0,1}x^2 + C_{0,2}x + C_{0,3}$$

则列混合相当于在 $GF(2^8)$ 上与 $C(x)$ 在模 $x^4 + 1$ 意义下相乘

$$S'_{0,0}x^3 + S'_{0,1}x^2 + S'_{0,2}x + S'_{0,3} = (S_{0,0}x^3 + S_{0,1}x^2 + S_{0,2}x + S_{0,3})c(x) \pmod{x^4 + 1}$$

Round Key Addition 轮密钥加 简单的比特异或将轮密钥作用在状态上。其中轮密钥通过密钥调度获得，长度为 N_b 个字 (32-bit)，于是第 i 列异或第 i 个字即可。

Key Expansion 密钥扩展 在 10 轮 AES 中共使用了 11 次轮密钥加。故需把 4 字初始密钥扩展为 44 字密钥。设初始密钥为 w_0, w_1, w_2, w_3 。则

$$\left\{ \begin{array}{l} w_{4k} = \text{ByteSub}(w_{4k-1} \ll 8) \oplus w_{4(k-1)} \oplus [R C_k, 0, 0, 0]^T \\ \quad \text{按字置换(同前文) 循环左移1字节} \qquad \qquad \qquad \text{轮常量异或} \\ w_{4k+i} = w_{4(k-1)+i} \oplus w_{4k+i-1} \end{array} \right. \quad (k \neq 0)$$

密码分析 (穷举法) 时间代价约为 2^{128} (对 128 bit 密钥)；

(**⑤安全**) (最佳密钥恢复攻击) 仅比穷举法快 4 倍。

(相关密钥攻击) 需 4 个关联密钥等前提，预计需 2^{97} 时间。

(边侧道攻击) 对加密时功率等参数进行监视，可行性低。

SM4 我国颁布的商用密码算法中的分组密码算法，是迭代分组密码算法。分组长度

(**⑤安全**) 128 bit，32 轮迭代结构，非平衡 Feistel 结构，安全性与 AES 相当。

加解密 设明文输入为 X_0, X_1, X_2, X_3 ，密文输出为 Y_0, Y_1, Y_2, Y_3 (4 个字，每字 32 bit)

$$X_{i+4} = F(X_i, X_{i+1}, X_{i+2}, X_{i+3}, rk_i) = X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i)$$

其中，T 为复杂置换 (每个字节进入 S-box 后再分别循环左移 0, 2, 10, 18, 24 位并异或结果)。

$$\text{则密文 } (Y_0, Y_1, Y_2, Y_3) = R(X_{32}, X_{33}, X_{34}, X_{35}) = (X_{35}, X_{34}, X_{33}, X_{32})$$

密钥扩展 首先进行加密密钥初始化

$$(K_0, K_1, K_2, K_3) = MK \oplus FK = (MK_0 \oplus FK_0, MK_1 \oplus FK_1, MK_2 \oplus FK_2, MK_3 \oplus FK_3)$$

其中 FK 为规定的 128 bit 常数，然后对 $i=0, 1, 2, \dots, 31$ 执行

$$rk_i = K_{i+4} = K_i \oplus T'(K_{i+1} \oplus K_{i+2} \oplus K_{i+3} \oplus CK_i)$$

其中 CK_i 为 32 个 32 bit 固定常数， T' 为复杂置换（每个字节进入 S-box 后再分别循环左移 0, 13, 24 位并异或结果）

分组密码应用模式：ECB, CBC, CFB, OFB, CTR. (NIST 标准)

ECB 模式（电码本模式）用相同密钥对各个明文组分别加密，一次处理一组明文的加密
对明文分段，最后一段可能需要填充。

加密 $C_i = E_K(P_i)$

解密 $P_i = D_K(C_i)$

特征 相同明文对应相同密文。

攻击 ① 若消息有固定开头（报文头），则其加密后开头仍相同，可识别有关信息。

② 交换明文组顺序，并交换对应密文顺序，即可进行对分组的代换重排

(e.g. “A → B 10元”，“B → C 10元”两个明文组可交换顺序产生错误)

应用 单个数据的安全传输。

误差扩散：密文块传输错误只会导致当前块出错。

CBC 模式（密码分组链接模式）为克服 ECB 缺点，将明文分为若干块，并通过加

密将密文分组与当前明文分组连接在一起，同时使用 初始向量 IV，即

加密 $C_0 = IV, C_i = E_K(P_i \oplus C_{i-1})$ 必须均匀随机选择！

解密 $C_0 = IV, P_i = D_K(C_i) \oplus C_{i-1}$

应用 对大量数据的加密与认证。

消息填充 消息最后一个分块长度小于分组长度时，可填充 null 或写入填充字数。（比如分组长度 64 时，[b1 c2 a3] 可填充为 [b1 c2 a3 00 00 00 00 05]）

PKCS #5 Padding CBC 模式在最后的块中，若缺了 b 个字节，则填入 b 个 “0b”

优点: ① 每个密文分组均依据于之前全部分组 ② 对任意一个分组改动会影响后续所有密文分组. ③ 初始向量 IV.

缺点: 如果攻击者获得了 IV, 可通过预先改变 IV 中的某些位, 使接收者明文中部分被取反. (由于 $P_i = D_K(C_i) \oplus IV$) (可是攻击者仍不知道消息, 因此这种解决方法 在发送消息前用 ECB 来保护 IV 攻击攻击的是完整性, 而非保密性)

误差扩散: 一块密文传输错误影响该块及下一块的明文解密.

带验证的 PKCS #5 - Padding 的 CBC 模式: 根据 Padding 顺序进行完整性验证.

Padding Oracle 利用密文的 Padding 有效性测试确定明文长度, 并从后往前推

△不安全 断每个字节的数据

e.g. 设共有 N 个块, 则先枚举修改 C_{N-1} 的第 i 个字节. 若修改第 i^* 个字节后无法通过有效性测试, 说明最后一个明文分组块有 $(i^* - 1)$ 个字节.

从后往前测试, 设已知当前块长度为 2, 故明文为

$P_{N,1} P_{N,2} 0b 0b 0b 0b 0b 0b$

修改 C_{N-1} 的后 6 个字节为 $C'_{N-1,i} = C_{N-1,i} \oplus 0b \oplus 07$. ($3 \leq i \leq 8$)

那么 $P'_{N,3} = D_K(C_{N,3}) \oplus C'_{N-1,3} = D_K(C_{N,3}) \oplus C_{N-1,3} \oplus 0b \oplus 07 = 07$

修改 C_{N-1} 的第 2 个字节为 $C'_{N-1,2} = X$ ($00 \leq X \leq FF$, 枚举)

则要使有效性验证通过, 有 $D_K(C_{N,2}) \oplus C'_{N-1,2} = 07$. 设此时 $C'_{N-1,2} = X^*$

从而知 $D_K(C_{N,2}) = 07 \oplus X^*$. 故 $P_{N,2} = D_K(C_{N,2}) \oplus C_{N-1,2} = 07 \oplus X^* \oplus C_{N-1,2}$

从而得到明文最后一字节, 进而可得所有明文

启示: NIST 标准仅为加密标准, 不能因为其具有的部分可用作认证性(完整性)验证的特点来同时进行其他功能, 否则可能会丧失加密安全性

CFB 模式(密码反馈模式) 消息被看作比特流, 本质是分组加密, 可以用于实现流密码. 实现每次对任意比特的加密 (CFB-1, CFB-8, CFB-14, CFB-128). 明文消息与分组加密的输出进行异或, 所得反馈进入移位寄存器作为下一阶段的输入.

加密 $C_i = P_i \oplus \underbrace{A_S(E_K(S_s(C_{i-1})))}_{\text{取前}s位 移位寄存器截断左移}s位并填充C_{i-1}至后}s位$ 其中 s 为分组长度

解密 $P_i = C_i \oplus A_s(E_k(S_s(C_{i-1})))$, $C_0 = IV$

优点 适合数据流的加密，可以被视为流密码；但又与流密码的典型构造不一致（不能依据初始值与密钥输出密钥流）；在已知全部密文时，解密可并行

缺点 每得到 s 个输入比特需暂停进行分组加密，存在误差扩散。

误差扩散 密文传输中某位错误，会影响后继 $\lceil \frac{L}{s} \rceil + 1$ 块密文的解密，其中 L 为移位寄存器的长度。

OFB 模式（输出反馈模式） 与 CFB 相似，消息被看作比特流；与 CFB 不同，加密函数的输出被反馈回移位寄存器作为下一次的输入（而非密文单元）。因为反馈值与明文消息无关，故均可预先计算。分组加密的输出与明文异或得到密文。

加密 $C_i = P_i \oplus D_i$, $D_i = A_s(E_k(S_s(D_{i-1})))$

$D_0 = IV$

NIST 标准中，IV 不移位

解密 $P_i = C_i \oplus D_i$, D_i 生成与加密相同

应用 有信道噪声环境下的流密码。

优点 传输过程中，比特差错不会扩散，可以预计算分组，流加密不需停顿。其仍具有 CFB 模式的其他优点。

缺点 对抗消息流篡改攻击的能力弱于 CFB。若对密文取反，则解密后明文也取反。
密钥由 Vernam 密码生成
可看作一次一密的变形，故不能使用重复密钥序列及 IV。发送方与接收方应同步

误差扩散 密文块传输错误只会导致当前块出错

CTR 模式 与 OFB 相似，只是分组加密的输入为计数，可并行处理。均匀随机生成 $Counter_i$ 。

加密 $C_i = P_i \oplus D_i$, $D_i = E_k(Counter_i)$, $Counter_i = Counter_{i-1} + 1$

解密 $P_i = C_i \oplus D_i$, D_i 生成与加密相同

应用 高速网络加密。

优点 高效（可并行处理，预计算且支持随机访问）

可证安全（CPA 安全）前提：使用了真正随机数生成器且不重用计数器值

缺点 计数器值与密钥绝不重用

误差扩散 密文块传输错误只会导致当前块出错

随机数 用途 认证方案、会话密钥产生、RSA 公开加密算法中密钥产生等。

检验随机程度的准则 均匀分布(出现概率大致相等)、独立性。

分类 真随机数：由于随机性与精确程度问题，真随机数源难以获得。

伪随机数：使用确定算法生成的，但能经受随机性检测的算法，由伪随机数产生器 PRG 产生。

随机性测试 均匀性：0/1 出现概率与量大致相同

可伸缩性：测试可用于子序列。

一致性：测试针对所有的初始值(初始向量)

} 测试特征

* Maurer 通用统计测试：测试是否可以大幅压缩数据且不损失信息。

* 不可预测性 { 前向不可预测性：不知道种子，则无法从当前位置推知下一位。
后向不可预测性：无法从序列推知种子值

线性同余伪随机数产生器 $X_{n+1} = (aX_n + c) \bmod m$ m 为素数。

攻击方法 若知道 a, c, m 即可推得后续随机数

△不安全 知道 4 个连续随机数 形成 3 个线性同余式即可恢复 a, c, m 。

结论 一旦知道了序列的一部分，不可预测程度就会变得很差。

分组密码伪随机数产生器 常用于由主密钥生成会话密钥。

计数器模式 $X_i = E_{Km}(i)$

输出反馈模式 $X_i = E_{Km}(X_{i-1})$

ANSI X9.17 伪随机数产生器 在金融安全及 PGP 中被应用

①安全 $R_i = EDE_{K_1, K_2}[V_i \oplus EDE_{K_1, K_2}[DT_i]] \rightarrow 3DES$
 $V_{i+1} = EDE_{K_1, K_2}[R_i \oplus EDE_{K_1, K_2}[DT_i]]$

其中， R_i 为第 i 轮随机数 V_i 为第 i 轮种子， DT_i 为第 i 轮时间。

Blum 整数 基于公钥算法，选择两个素数满足 $p \equiv q \equiv 3 \pmod{4}$

②安全 令 $n = pq$ ，选择种子 s 与 n 互质。 $X_0 = s^2 \bmod n$

输出： $X_n = X_{n-1}^2 \bmod n$, $R_i = X_i \bmod 2$

优点: ① 在续位测试下满足不可预测性，即不存在多项式时间算法
在给定序列前 k 位输入时以不可忽略概率预测出第 $(k+1)$ 位
② 安全性基于分解大整数 n 困难性。
③ 给定任意位 bit 仍满足不可预测性
缺点: 由于 n 很大，故很慢以至于仅可用于密钥生成。