**分组密码设计原则：**

① 安全原则

安全性是分组密码设计时应考虑的最重要因素，保证分组密码安全性的两个基本设计原则的思想就是Shannono 1949年发现的隐藏信息的两种技术：混淆和扩散。（可以避免统计方法的密码分析）。

② 实现原则：硬件实现原则 & 软件实现原则

**“雪崩效应”**：在密码学中雪崩效应是指加密算法（尤其是块密码和加密散列函数）的一种理想属性；雪崩效应是指当输入发生最微小的改变（例如：反转一个二进制位）时，也会导致输出的不可区分性改变（输出中每个二进制位50%的概率发生反转）；合格块密码中，无论密钥或明文的任何细微变化都必须引起密文的不可区分性改变；目的是抗击敌手对密码系统的统计分析。

2.扩散的实现

扩散：将明文的统计特性分布到密文中去。

实现方式是使得明文的每一位影响密文中多位的值，等价于说密文中每一位均受明文中多位影响。这时明文的统计特性将被散布到密文中；因而每一字母在密文中出现的频率比在明文中出现的频率更接近于相等，双字母及多字母出现的频率也更接近于相等； 在二元分组密码中，可对数据重复执行某个置换，再对这一置换作用于某一函数（代换），可获得扩散。（置换+代换）

扩散的目的：使明文和密文之间的统计关系变得尽可能复杂，以使敌手无法得到密钥。

3. 混淆的实现

混淆：是是密文和密钥之间的统计关系变得尽可能复杂，以使敌手无法得到密钥。

即使敌手能得到密文的一些统计关系，由于密钥和密文之间的统计关系复杂化，敌手也无法获得密钥；使用复杂的代换算法可以得到预期的混淆效果。简单的线性代换函数得到的混淆效果则不够理想，一般采用非线性代换。

扩散和混淆成功地实现分组密码本质属性，因而成为设计现代分组密码的基础。

**DES 加密算法环节的安全性数学证明**

1.置换和替代操作的数学性质

DES 算法中包含了复杂的置换（Permutation）和替代（Substitution）操作。置换操作可以看作是对数据的重新排列，从数学角度可以用矩阵变换来描述。例如，初始置换 IP 将 64 位的数据重新排列成特定的顺序，这可以用一个 64×64 的置换矩阵来表示。替代操作则是根据特定的查找表进行的，例如 S 盒（Substitution box）替代，每个 S 盒都是一个非线性的映射，可以看作是一个复杂的函数。

这些操作的组合使得 DES 算法具有很高的复杂性，增加了破解的难度。从数学上可以证明，经过多次置换和替代操作后，数据的分布变得非常均匀，使得攻击者难以通过分析密文的统计特性来推断明文或密钥。

2.密钥扩展的安全性

DES 使用 56 位的密钥，并通过密钥扩展算法生成 16 轮的子密钥。密钥扩展算法是一种复杂的数学运算，它确保了每一轮的加密都使用不同的子密钥。

从数学角度来看，密钥扩展算法具有一定的扩散性和混淆性。扩散性使得密钥的每一位都影响到密文的多个位，混淆性则使得密文与密钥之间的关系变得非常复杂，难以通过分析密文来推断密钥。

密钥扩展算法中的扩散性和混淆性对 DES 安全性有着至关重要的影响。以下将详细阐述这两个特性如何具体影响 DES 的安全性。

**一、扩散性对 DES 安全性的影响**

增加密码分析难度：扩散性使得明文的每一位影响密文的多位，这样即使攻击者知道了密文的一部分信息，也很难通过这部分信息推断出明文的内容。例如在改进的 DES 加密算法中，通过对密钥进行扩展处理，将密钥的位数从传统的 56 位扩展到 128 位等更高的位数，使得加密过程中密钥的扩散性增强。这种增强的扩散性使得攻击者在进行密码分析时，需要考虑更多的可能性，从而大大增加了密码分析的难度。

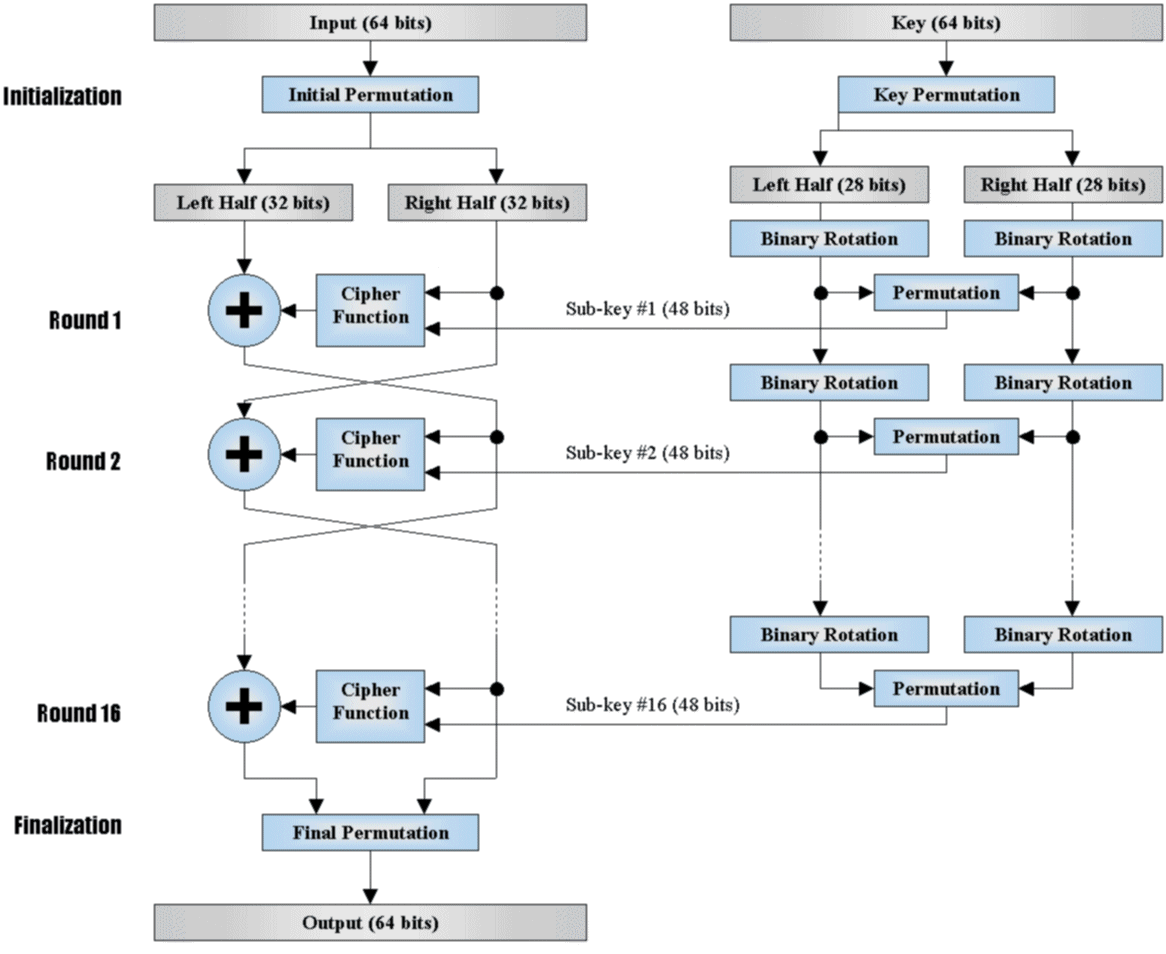
抵抗统计分析攻击：具有良好扩散性的密钥扩展算法可以有效地抵抗统计分析攻击。在传统的网络加密算法中，加密密钥仅为 56 位时，安全数据受攻击时易于被破解，安全性较低。而通过对密钥进行扩展并增强其扩散性，可以使得密文中的统计特征更加均匀，减少明文与密文之间的统计关联性，从而降低攻击者通过统计分析方法破解密码的可能性。

**二、混淆性对 DES 安全性的影响**

增加密钥的不确定性：混淆性使得密文与密钥之间的关系变得复杂，增加了密钥的不确定性。在 AES 等加密算法中，混淆性是通过置换等操作实现的。对于 DES 来说，改进的密钥扩展算法可以引入更多的混淆操作，使得攻击者难以从密文中推断出密钥的信息。例如，采用优化的 DES 算法加密级联倾斜光纤光栅网络安全数据时，将传统 DES 算法中 56 位密钥扩展处理至 128 位密钥后分为四轮，选取 32 位改造算法处理各轮密钥获取子密钥，异或运算 32 位明文以及子密钥，将 Feistel 加密算法应用于异或运算结果中实施轮函数运算后，对运算结果进行 48 位扩展变换处理，利用置乱处理方法对扩展变换后结果实施 P 排序操作，增强了混淆性。

提高抗差分密码分析能力：混淆性可以提高 DES 算法抗差分密码分析的能力。差分密码分析是一种针对分组密码的攻击方法，通过分析明文对密文的影响来推断密钥信息。良好的混淆性可以使得明文的微小变化在密文中产生较大的变化，从而增加攻击者进行差分密码分析的难度。例如，对 AES 算法中密钥扩展过程和 S 盒构造进行改进，提高了算法的安全性，增强了混淆性，从而提升了其抵抗攻击的能力。

下图为**DES算法过程**：



**DES 算法的安全性评估：**

**密钥长度**：DES 算法使用 56 位的密钥对数据进行加密。理论上，密钥空间大小为2^{56}（约 7.2 × 10^{16} 种可能的密钥组合）。在过去，对于计算能力有限的情况下，如此庞大的密钥空间使得通过穷举法破解密钥在时间和计算资源上是不可行的，这为 DES 算法提供了一定的安全性保障。然而，随着计算机技术的不断发展，计算能力不断提高，这种密钥长度逐渐变得不再足够安全。

密钥生成过程：密钥的生成过程是否安全也会影响算法的安全性。如果密钥生成过程存在漏洞或可预测性，那么即使密钥长度足够长，也可能降低算法的安全性。在 DES 算法中，密钥经过一系列的置换、移位等操作生成子密钥，这些操作的复杂性和随机性对于密钥的安全性至关重要。

**2.算法的数学基础和复杂性：**

Feistel 结构：DES 算法基于 Feistel 网络结构，这种结构具有良好的混淆和扩散特性。混淆是指使密文与密钥之间的关系变得复杂，难以通过分析密文来推断密钥；扩散是指将明文的每一位信息尽可能地扩散到密文中，使得明文的微小变化会导致密文的较大变化。Feistel 结构通过多轮的迭代运算，使得加密过程具有较高的复杂性，增加了破解的难度。

迭代次数：DES 算法经过 16 轮的迭代运算，每一轮都使用不同的子密钥对数据进行处理。足够的迭代次数可以使明文充分地被混淆和扩散，提高加密的强度。如果迭代次数过少，可能导致加密不够充分，容易被破解；而过多的迭代次数则会增加计算成本，影响算法的效率。研究表明，DES 的 16 轮迭代次数在安全性和效率之间取得了较好的平衡。

**3.密码分析攻击的抵抗能力：**

差分攻击：差分攻击是一种针对分组密码的分析方法，通过分析明文对的差值和密文对的差值之间的关系来寻找算法的弱点。对于 DES 算法，研究人员已经对其进行了大量的差分攻击研究。结果表明，DES 算法在一定程度上能够抵抗差分攻击，但如果攻击者能够获取大量的明密文对，并进行深入的分析，仍然有可能找到算法的弱点。

线性攻击：线性攻击是另一种常见的密码分析方法，通过寻找密文和明文之间的线性关系来破解密码。DES 算法也需要抵抗线性攻击的考验。研究人员通过对 DES 算法的线性特性进行分析，评估其在线性攻击下的安全性。

相关密钥攻击：相关密钥攻击是指攻击者利用多个相关的密钥对加密系统进行攻击。在某些情况下，如果密钥的生成或管理存在问题，可能会导致密钥之间存在相关性，从而使攻击者能够利用这种相关性进行攻击。DES 算法在设计时需要考虑到相关密钥攻击的可能性，并采取相应的措施来降低风险。

**4.密文与明文、密文与密钥的相关性：**

密文与明文的相关性越低，说明算法的混淆和扩散效果越好，安全性越高。研究人员通过对 DES 算法的密文和明文进行相关性分析，检验算法是否能够有效地将明文的信息隐藏在密文中。

密文与密钥的相关性也需要进行评估。如果密文与密钥之间的相关性较强，攻击者可能通过分析密文来获取关于密钥的信息，从而降低算法的安全性。

**5.实现和应用环境的安全性：**

软件实现：如果 DES 算法的软件实现存在漏洞，例如代码中的缓冲区溢出、内存泄漏等问题，可能会被攻击者利用来获取密钥或篡改加密数据。因此，在评估 DES 算法的安全性时，需要对其软件实现进行严格的审查和测试。

硬件实现：在一些特定的应用场景中，DES 算法可能会在硬件设备上实现，如加密芯片、智能卡等。硬件实现的安全性需要考虑硬件的物理安全性、抗攻击能力以及硬件与软件之间的交互安全性等因素。

密钥管理：密钥的安全管理是保证加密算法安全性的关键。对于 DES 算法，密钥的生成、存储、分发和更新等环节都需要严格的安全措施。如果密钥管理不善，例如密钥泄露、密钥被窃取或篡改等，将直接导致加密数据的安全性受到威胁。

**一、穷举攻击基于 DES 的弱点**

密钥长度较短：DES 算法的密钥长度为 56 位。在现代计算机强大的计算能力下，这个长度相对较短。攻击者可以通过穷举所有可能的 56 位密钥来尝试破解密文。虽然理论上密钥空间有 2^56 种可能，但随着计算机技术的不断发展，尤其是并行计算和分布式计算的出现，使得在合理时间内进行暴力破解成为可能。

加密速度较慢：DES 算法的加密速度比较慢，这使得攻击者有更多的时间来进行穷举尝试。虽然加密速度慢本身不是一个直接的弱点，但它为攻击者提供了更多的机会来进行攻击。

成功案例：1998年，电子前沿基金会（EFF）使用一台造价约25万美元的特制DES解密机，通过穷举攻击在两天多的时间内成功破解了DES加密。

**二、差分密码分析攻击基于 DES 的弱点**

对明文和密文之间的差异敏感：DES 算法在加密过程中，对于不同的输入明文，其输出密文之间的差异可能会被攻击者利用。差分密码分析攻击通过分析不同的输入和输出差异，以及密钥可能取值的概率，从而获得密钥。

加密过程中的非线性变换不够复杂：DES 算法中的非线性变换部分，如 S 盒，虽然在一定程度上增加了密码的复杂性，但仍然存在被攻击者分析和破解的可能性。如果非线性变换不够复杂，攻击者可以通过分析大量的明文和密文对，找到其中的规律，从而进行差分密码分析攻击。

成功案例：1990年，Biham和Shamir公开了差分密码分析技术后，理论上证明了该方法可对DES进行有效攻击。

**三、线性密码分析攻击基于 DES 的弱点**

线性近似的存在：DES 算法在加密过程中，可能存在一些线性近似关系。攻击者可以通过构造一些线性逼近，以及计算相应的概率，从而推导出密钥。

子密钥的独立性不足：DES 算法在加密过程中，使用了多个子密钥。如果这些子密钥之间的独立性不足，攻击者可以通过分析子密钥之间的关系，以及明文和密文之间的线性近似，进行线性密码分析攻击。

成功案例：1993年，Matsui提出了线性密码分析方法，并成功对简化版的DES进行了攻击。虽然对标准DES的攻击需要大量的已知明文，但从理论上证明了该方法的可行性。

**相关论文中成功攻击 DES 的情况**

穷举攻击：许多论文都提到了由于 DES 算法的密钥长度较短，容易受到穷举攻击。例如，论文指出，由于 H/W 技术的进步，DES 的 56 位密钥长度已经不够安全，容易受到暴力攻击。

[1]张丽丽,张玉清.基于分布式计算的暴力破解分组密码算法[J].计算机工程,2008,(13):121-123.

差分密码分析攻击：论文提到了图像加密方案中存在的弱点，这些弱点同样适用于 DES 算法。差分密码分析攻击是一种选择明文攻击，利用了特定明文 - 密文差异的高概率发生。攻击者可以通过分析大量的明文和密文对，找到其中的规律，从而进行差分密码分析攻击。

线性密码分析攻击：论文讨论了轻量级 DES 算法（DESL）对线性密码分析攻击的免疫力，并指出传统的 DES 算法容易受到线性密码分析攻击。线性密码分析攻击通过找到涉及明文位、密文位和子密钥位的线性表达式的高概率发生来确定密钥位。

综上所述，DES 算法由于密钥长度较短、加密过程中的非线性变换不够复杂以及子密钥的独立性不足等弱点，容易受到穷举攻击、差分密码分析攻击和线性密码分析攻击。相关论文中也有很多关于成功攻击 DES 算法的例子。为了提高密码的安全性，人们提出了许多改进的方法，如增加密钥长度、改进加密算法的非线性变换等。

《EFF-DES破解机》

介绍了DES加密算法长期面临的“密钥长度过短、安全性不足”的批评，以及在1997年至1999年间举办的DES挑战赛中，不同团队对DES的破解情况。

文中提到了洛克·沃瑟团队在第一届DES挑战赛中耗时96天破解成功；distributed.net的22,000名志愿者在第二届初赛中历时39天破解成功；电子前哨基金会的“深译”在复赛中仅用56小时成功破解；在1999年1月，电子前哨基金会与distributed.net团队合作不到一天（22小时15分）再次破解。

[EFF-DES破解机\_百度百科](https://baike.baidu.com/item/EFF-DES%E7%A0%B4%E8%A7%A3%E6%9C%BA/22718143)

《离线破解Kerberos中的DES密钥》

主要讲述了在AD域中通过构造DES加密的票据申请流程来获取hashcat可破解的DES hash从而破解DES密钥的方法。

文中详细介绍了破解流程，包括申请DES加密的TGT、使用TGT申请TGS、获取第一段明文分组、提取可破解的DES hash以及使用hashcat进行破解等步骤，还提到了利用破解密钥的几种方式。

[离线破解Kerberos中的DES密钥](https://mp.weixin.qq.com/s?__biz=MzkyMTI0NjA3OA==&mid=2247492962&idx=1&sn=23da09c660858f934a06bc1f1367b1d7&scene=0)

《一种分组密码防攻击的加解密方法和装置》

提出了一种分组密码防攻击的加解密方法和装置，主要针对故障攻击进行防范。

该方法包括对同一数据进行多次运算并比较结果，对运算结果进行逆运算并比较等方式来检测故障注入攻击，通过在加密和解密过程中注入故障并利用函数处理正确和错误结果来提高安全性。

[一种分组密码防攻击的加解密方法和装置\_百度百科](https://baike.baidu.com/item/%E4%B8%80%E7%A7%8D%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81%E9%98%B2%E6%94%BB%E5%87%BB%E7%9A%84%E5%8A%A0%E8%A7%A3%E5%AF%86%E6%96%B9%E6%B3%95%E5%92%8C%E8%A3%85%E7%BD%AE/59125669)