文章编号:1000－1638(2005)06-0693-05

# DES 算法安全性的分析与研究*

胡美燕,刘然慧
(山东科技大学信息工程系,山东 泰安 271019)

**摘要**:对 DES 算法中密钥的长度、弱密钥、S_盒的设计、迭代次数以及安全性进行了分析和研究,对 DES 中存在的漏洞,提出了几种变形 DES 方案,以求解决,增强其安全性.

**关键词**:DES 算法;密钥;迭代次数;DES 变形
**中图分类号**:TP309.7 **文献标识码**:A

## 1 DES 算法的安全性

自分组密码(DES 算法)1977 年首次公诸于世以来,引起了学术界和企业界的广泛重视. 学术界对 DES 密码进行了深入的研究,围绕它的安全性和破译方法展开了激烈的争论,在一定意义上对密码学的理论研究也起了推动作用. 同时人们也一直对 DES 的安全性持怀疑态度,对密钥的长度、迭代次数及 S 盒的设计众说纷纭.

### 1.1 对称分组密码算法的问题

DES 是对称的分组密码算法. 对称的分组密码算法最主要的问题是:由于加解密双方都要使用相同的密钥,因此在发送、接收数据之前,必须完成密钥的分发. 因而,密钥的分发便成了该加密体系中的最薄弱、风险最大的环节,各种基本的手段均很难保障安全地完成此项工作,从而使密钥更新的周期加长,给他人破译密钥提供了机会. 实际上这与传统的保密方法差别不大. 在历史战争中,破获他国情报的纪录不外是两种方式:一种是在敌方更换"密码本"的过程中截获对方密码本;另一种是敌人密钥变动周期太长,被长期跟踪,找出规律从而被破获. 在对称算法中,尽管由于密钥强度增强,跟踪找出规律破获密钥的机会大大减小了,但密钥分发的困难问题几乎无法解决. 如,设有 n 方参与通信,若 n 方都采用同一个对称密钥,一旦密钥被破解,整个体系就会崩溃;若采用不同的对称密钥则需 n(n－1)个密钥,密钥数与参与通信人数的平方数成正比,这便使大系统密钥的管理几乎成为不可能.

### 1.2 DES 算法的弱密钥

由于算法各轮的子密钥是通过改变初始密钥这种方式得到的,因此有些初始密钥成了弱密钥(weak key). 初始密钥分成两部分,每部分各自独立的移动. 如果每一部分的所有位都是 0 或 1,那么算法的任意一个周期的密钥都是相同的. 当密钥是全 1、全 0、或者一半全 1、一半全 0 时,会发生这种情况. 下面以十六进制编码的方式给出了四种弱密钥.

| 弱密钥值(带奇偶校验位) | 真实密钥 |
|---|---|
| 0101 0101 0101 0101 | 0000000 0000000 |
| 1F1F 1F1F E0E0 E0E0 | 0000000 FFFFFFF |
| E0E0 E0E0 1F1F 1F1F | FFFFFFF 0000000 |
| FEFE FEFE FEFE FEFE | FFFFFFF FFFFFFF |

另外,还有一些密钥对明文加密成相同的密文. 换句话说,密钥对里的一个密钥能解密另外一个

* 收稿日期:2005-12-12
作者简介:胡美燕(1963～),女,河北高碑店人,副教授.

密钥加密的信息. 这是由 DES 产生子密钥(subkey)的方式决定的. 这些密钥只产生 2 个不同的子密钥,算法中每个这样的子密钥都使用了 8 次. 这样的子密钥叫半弱密钥(semiweak key),下表以十六进制编码方式给出它们.

| | |
|---|---|
| 01FE 01FE 01FE 01FE | FE01 FE01 FE01 FE01 |
| 1FE0 1FE0 0EF1 0EF1 | E01F E01F F10E F10E |
| 01E0 01E0 01F1 01F1 | E001 E001 F101 F101 |
| 1FFE 1FFE 0EFE 0EFE | FE1F FE1F FE0E FE0E |
| 011F 011F 010E 010E | 1F01 1F01 0E01 0E01 |
| E0FE E0FE F1FE F1FE | FEE0 FEE0 FEF1 FEF1 |

另外,也只有产生 4 个子密钥的密钥,每个这样的子密钥在算法中使用了 4 次. 下面给出他们的十六进制编码形式:

| | |
|---|---|
| 1F1F 0101 0E0E 0101 | E001 01E0 F101 01F1 |
| 011F 1F01 010E 0E01 | FE1F 01E0 FE0E 01F1 |
| 1F01 011F 0E01 010E | FE01 1FE0 EF01 0EF1 |
| 0101 1F1F 0101 0E0E | E01F 1FE0 F10E 1FE1 |
| E0E0 0101 F1F1 0101 | FE01 01FE EF01 01FE |
| FEFE 0101 FEFE 0101 | E01F 01FE F10E 01FE |
| FEE0 1F01 FEF1 0E01 | E001 1FFE F101 0EFE |
| E0FE 1F01 F1FE 0E01 | FE1F 1FFE FE0E 0EFE |
| FEE0 011F F1FE 010E | 1FFE 01E0 0EFE 01F1 |
| E0FE 011F F1FE 010E | 01FE 1FE0 01FE 0EF1 |
| E0E0 1F1F F1F1 0E0E | 1FE0 01FE 0EF1 01FE |
| FEFE 1F1F FEFE 0E0E | 01E0 1FFE 01F1 0EFE |
| FE1F E001 FE0E F101 | 0101 E0E0 0101 F1F1 |
| E01F FE01 F10E FE01 | 1F1F E0E0 0E0E F1F1 |
| FE01 E01F FE01 F10E | 1F01 FEE0 0E01 FEF1 |
| E001 FE0F F101 FE0E | 011F FEE0 010E FEF1 |
| 01E0 E001 01F1 F101 | 1F01 E0FE 0E01 F1FE |
| 1FFE E001 01FE F001 | 011F E0FE 010E F1FE |
| 1FE0 FE01 0EF1 FE01 | 0101 FEFE 0101 FEFE |
| 01FE FE01 01FE FE01 | 1F1F FEFE 0E0E FEFE |
| 1FE0 E01F 0EF1 F10E | FEFE E0E0 FEFE F1F1 |
| 01FE E01F 01FE F10E | E0FE E0E0 FEFE F1F1 |
| 01E0 FE1F 01F1 FE0E | FEE0 E0FE FEF1 FEF1 |
| 1FFE FE1F 0EFE FE0E | E0E0 FEFE F1F1 FEFE |

虽然 DES 有弱密钥,但这张 64 个密钥的密钥表相对于总数位 72,057,594,037,927,936 个可能密钥的密钥集只是个零头,如果随机的选择密钥,选中这些弱密钥的可能性可以忽略. 而且我们可以在选择密钥时进行检查,以防止产生弱密钥.

### 1.3 密钥的长度

密钥仅有 56 位二进制未免太短,多密码学专家力荐使用更长的密钥,理由是穷举攻击的可能性. 设已知一段密码文 C 及与它对应的明码文 M,用一切可能的密钥 K 加密 M,直到得到 E(M)=C,这时所用的密钥 K 即为要破译的密码的密钥. 穷举法的时间复杂性是 T=O(n),空间复杂性是 S=O(1). 对于 DES 密码,$n=2^{56}\approx 7\times 10^{6}$,即使使用每秒种可以计算一百万个密钥的大型计算机,也需要算 $10^{6}$ 天才能求得所使用的密钥,因此看来是很安全的.

但是密码专家 Diffie 和 Hellman 指出,如果设计一种一微秒可以核算一个密钥的超大规模集成片,那么它在一天内可以核算 $8.64\times 10^{10}$ 个密钥. 如果由一个百万个这样的集成片构成专用机,那么

它可以在不到一天的时间内用穷举法破译 DES 密码. 他们当时(1977 年)估计:这种专用机的造价约为两千万美元. 如果在五年内分期偿还,平均每天约需付一万美元. 由于用穷举法破译平均只需要计算半个密钥空间,因此获得解的平均时间为半天. 这样,破译每个 DES 密码的花销只是五千美元. 后来,Diffie 在 1981 年又修改了他们的估计,认为以 1980 年的技术而论,用造价为五千万美元的专用机破译 DES 密码平均要花两天时间. 但是他与 Hellman 都预计:1990 年时,破译 DES 密码的专用机的造价将大幅度下降.

同时,DES 的硬件实现方法逐步接近 Diffie 和 Hellman 的专用机所要求每秒百万次的速度. 在 1993 年,Muchael Wiener 设计了一个一百万美元的机器,它能在平均 3.5 小时内,完成 DES 的穷举攻击,到 1990 年时,DES 是完全不安全的.

### 1.4　迭代次数

根据目前的计算技术和 DES 的分析情况,16-圈 DES 仍然是安全的,但提醒使用者不要使用低于 16-圈的 DES,特别是 10-圈以下的 DES,Bihan 和 Shamir 的差分密钥分析同样也阐述了这一点:对于低于 16 轮的任意 DES 的已知明文攻击要比穷举攻击有效,但当算法恰好有 16 轮时,只有穷举攻击最有效.

### 1.5　S_盒和 P_盒的设计

实现替代函数 Si 所用的 S 盒的设计原理尚未公开,其中可能留有隐患. 更有人担心 DES 算法中有"陷阱",知道秘密的人可以很容易地进行密文解密. 目前人们仍然不知道 DES 中是否存在陷门. 所谓陷门,通俗地讲,就是在算法的设计中设计者留了一个后门,知道某一秘密的人可进入这一后门获得使用该算法的用户的秘密密钥. DES 的设计准则除了极少数被公布外,其余的仍然是保密的. 围绕 S_盒人们讨论了一系列问题包括设计准则和构造等. 在差分分析公开后,IBM 公布了 S_盒和 P_盒的设计准则.

#### 1.5.1　S_盒的设计准则

1)每个 S_盒均为 6 位输入,4 位输出. (这是在 1974 年的技术条件下,单个芯片所能容纳的最大尺寸.)

2)没有一个 S_盒的输出位是接近输入位的线性函数.

3)如果将输入位的最左、最右端的位固定,变化中间的 4 位,每个可能的 4 位输出只得到一次.

4)如果 S_盒的两个输入仅有 1 位的差异,则其输出必须至少有 2 位不同.

5)如果 S_盒的两个输入仅有中间 2 位不同,则其输出必须至少有 2 位不同.

6)如果 S_盒的两个输入前 2 位不同,后两位已知,则其输出必不同.

7)对于输入之间的任何非零的 6 位差分,32 对中至多有 8 对显示出的差分导致了相同的输出差分.

#### 1.5.2　P_盒的设计准则

1)在第 i 轮 S_盒的 4 位输出中,2 位将影响 S_盒第 i+1 轮的中间位,其余 2 位将影响最后位;

2)每个 S_盒的 4 位输出影响 6 个不同的 S_盒,但没有一个影响同一个 S_盒;

3)如果一个 S_盒的 4 位输出影响另一个 S_盒的中间 1 位,那么后一个的输出位不会影响前一个 S_盒的中间 1 位.

在今天看来产生 S_盒很容易,但在 70 年代初,这是一个很复杂的工作. Tuchman 曾经引述说,他们当时将计算机程序运行几个月来产生 S_盒. 在对 DES 密码进行鉴定的期间,美国国家保密局和计算机科学技术学会组织各界专家研究了 DES 密码体制的安全性问题,讨论了破译 DES 密码体制的一切可能途径. 尽管有些专家和学者对它的安全性仍持怀疑态度,但官方却得出了十分乐观的结论. 他们宣布:"没有任何可以破译 DES 密码体制的系统分析法. 若使用穷举法,则在 1990 年以前基本上不可能产生出每天能破译一个 DES 密钥的专用计算机. 即使届时能制造出这样的专用机,它的破译成功率也只会在 0.1 到 0.2 之间,而且造价可能高达几千万美元."

## 2 DES 算法的漏洞

由 DES 算法我们可以看到:DES 算法中只用到 64 位密钥中的其中 56 位,而第 8、16 、24、……64 位 8 个位并未参与 DES 运算. 这一点,向我们提出了一个应用上的要求,即 DES 的安全性是基于除了 8,16,24,……64 位外的其余 56 位的组合变化 256 才得以保证的. 因此,在实际应用中,我们应避开使用第 8,16,24……64 位作为 DES 密钥的有效数据位,而使用其它的 56 位作为有效数据位. 只有这样,才能保证 DES 算法安全可靠地发挥作用. 如果不了解这一点,把密钥 Key 的 8,16,24,……64 位作为有效数据位使用,将不能保证 DES 加密数据的安全性,对运用 DES 来达到保密作用的系统将产生数据被破译的危险,这正是 DES 算法在应用上的误区,是各级技术人员、各级用户在使用过程中应绝对避免的.

基于以上的问题,我们就不能用汉字作为密钥. 因为汉字由两个字节组成,每个字节的 ASCII 码都大于 127,转换成二进制就是 8 位,不能加奇偶校验位,而且如果去掉第 8、16 、24、……64 位,就会丢失密钥,而且不同的汉字可能实际上是相同的密钥.

## 3 DES 的变形

DES 算法目前已广泛用于电子商务系统中. 随着研究的发展,针对以上 DES 的缺陷,DES 算法在基本不改变加密强度的条件下,发展了许多变形 DES. 人们提出了几种增强 DES 安全性的方法,主要有以下几种.

### 3.1 多重 DES

为了增加密钥的长度,建议将一种分组密码进行级联,在不同的密钥作用下,连续多次对一组明文进行加密,通常把这种技术称为多重加密技术. 对 DES,建议使用三重 DES,这一点目前基本上达成一个共识. 这里介绍三重 DES.

因为确定一种新的加密法是否真的安全是极为困难的,而且 DES 主要的密码学缺点,就是密钥长度相对比较短,所以人们并没有放弃使用 DES,而是想出了一个解决其长度问题的方法,即采用三重 DES. 其基本原理是将 128 比特的密钥分为 64 比特的两组,对明文多次进行普通的 DES 加密操作,从而增强加密强度.

#### 3.1.1 用两个密钥的三重 DES

这种方法用两个密钥对明文进行三次加密,假设两个密钥是 K1 和 K2.

1)用密钥 K1 进行 DES 加密.

2)用 K2 对步骤 1 的结果进行 DES 解密.

3)用步骤 2 的结果,使用密钥 K1 进行 DES 加密.

C=Ek1(Dk2(Ek1(P)))加密;P=Dk1(Ek2(Dk1(C)))解密.

#### 3.1.2 三个密钥的三重 DES

C=Ek3(Dk2(Ek1(P)))加密;P=Dk3(Ek2(Dk1(C)))解密.

三重 DES 算法是扩展其密钥长度的一种方法,可使加密密钥长度扩展到 128 比特(112 比特有效)或 192 比特(168 比特有效). 此方法为密码专家默克尔(Merkle)及赫尔曼(Hellman)推荐. 据称,目前尚无人找到针对此方案的攻击方法. 如果用三重 DES 加密,建议使用三个不同的密钥.

### 3.2 S_盒可选择的 DES(也称带用交换 S_盒的 DES 算法)

密码专家比哈姆(Biham)和沙米尔(Shamir)证明通过优化 S_盒的设计,甚至 S_盒本身的顺序,可以抵抗差分密码分析,以达到进一步增强 DES 算法的加密强度的目的.

在一些设计中,将 DES 作如下改进:使 S_盒的次序随密钥而变或使 S_盒的内容本身是可变的. 8 个 DES 的 S_盒的改变可使得 DES 变弱许多,使用某些特定次序的 S_盒的 16-圈 DES,仅需要大约 2 个选择明文就能用差分分析方法被破译. 采用随机的 S_盒的 DES 很容易被破译,即使是对 DES

的一个 S_盒的数字稍作改变也会导致 DES 易于破译. 因此可得出结论：不管怎样随机选择 S_盒都不会比 DES 更安全.

3.3　具有独立子密钥的 DES

DES 的另一种变形是每圈迭代都使用不同的子密钥，而不是由单个的 56 比特密钥来产生. 因为 16-圈 DES 的每圈都需要 48 比特密钥，所以这种变形的 DES 的密钥长度是 768 比特. 这一方法可以增强 DES 的加密强度，大大地增加了实现 DES 的难度.

但据密码专家比哈姆(Biham)及沙米尔(Shamir)证明利用 261 个选择明文便可破译这个 DES 变形，而不是人们所希望的 2768 个选择明文. 所以这种改变并不能使 DES 变得更安全.

3.4　G-DES

G-DES 是广义的 DES 的缩写，设计它的目的是为了提高 DES 的速度和强度. 总的分组长度增加了(分组长度是可变的)，但圈函数 f 保持不变. Biham 和 Shamir 仅使用 16 个已知明文就能用差分分析破译分组长度为 256 比特的 16-圈 G-DES. 使用 48 个选择明文就能用差分分析破译分组长度为 256 比特的 22-圈 G-DES. 即使是分组长度为 256 比特的 64-圈 G-DES 也比 16-圈 DES 弱. 事实证明，比 DES 快的任何 G-DES 也就比它不安全.

## 4　结束语

DES 算法的实现有很多方法，有基于移位和异或运算的，有基于类封装的，本文是就基于数组实现的 DES 算法中存在的问题，如安全性、漏洞等作了初步分析，并提出了几种 DES 变形，在不改变加密强度的前提下，以求增强 DES 的安全性.

## 参考文献：

[1]　卢开澄. 计算机密码学 [M]. 北京：清华大学出版社，1998. 7.

[2]　[美]　Schneier B. 应用密码学：协议、算法与 C 源程序 [M]. 吴世忠，祝世雄，张文政，等译. 北京：机械工业出版社，2000.

[3]　杨波. 网络安全理论与应用 [M]. 北京：电子工业出版社，2002.

[4]　于秀源，薛昭雄. 密码学与数论基础 [M]. 济南：山东科学技术出版社，1993.

[5]　冯登国，吴文玲. 分组密码设计与分析 [M]. 北京：清华大学出版社，2000.

(责任编委　叶新铭)

# Analysis and Research of the Security of DES Algorithm

HU Mei-yan，LIU Ran-hui

(*Department of Information Engineering*，

*Shandong University of Science and Technology*，*Taian Shandong* 271019，*PRC*)

**Abstract**：The length of the key，the weak key，the design of the S_box，the repetitive rounds and their safety on the basis of the DES algorithm are analyzed and studied. Several transfigurations of DES are also put forward to strengthen its safety for the existing loophole in DES.

**Key words**：DES algorithm；key；repetitive round；DES transfiguration