**SDES算法的开发者手册**

**一、算法介绍**

SDES(Simplified Data Encryption Standard)是DES算法的简化版本,用于学习和 demonstrat the principles of modern symmetric-key cryptography。SDES保留了DES的基本结构,但进行了简化,以便更容易理解和实现。

**1.1 SDES的结构**

SDES采用Feistel网络结构,包含加密和解密两个过程。

加密过程包含两轮Feistel迭代。每轮中,将输入分为左半部和右半部。先对右半部做位扩展和密钥混淆,再通过S盒替代和P盒置换生成输出,与左半部做异或,产生下一轮的输入。两轮迭代后进行终态转换,得到加密结果。

解密过程为加密的反过程,但使用的子密钥顺序相反。解密不需要逆S盒和逆P盒,通过相同的S盒和P盒,以及使用子密钥的顺序反过来,即可完成解密。

**1.2 参数选择**

相比DES算法,SDES做了以下简化:

密钥长度 shorten 到10位二进制。

数据块长度 shorten 到8位二进制。

迭代轮数减少到2轮。

S盒数量减少到2个。

P盒规模减少。

**二、代码实现**

**2.1 密钥生成**

首先定义密钥长度KEY\_LENGTH为10位,数据块长度DATA\_LENGTH为8位。

生成子密钥函数subkey会先对输入的10位密钥做P10置换,然后左移指定位数,再通过P8置换生成两个8位的子密钥k1和k2。

@staticmethod

def subkey(k, p10\_table, p8\_table):

p10\_key = SDES.permute(k, p10\_table)

k1 = SDES.permute(SDES.ls(p10\_key, 1), p8\_table)

k2 = SDES.permute(SDES.ls(SDES.ls(p10\_key, 1), 1), p8\_table)

return k1, k2

**2.2数据处理**

2.2.1 数据处理包含以下几个步骤:

IP初始置换

两轮Feistel迭代

IP^(-1)逆置换

2.2.2 Feistel轮函数包含:

E扩展

密钥加

S盒替代

P盒置换

加解密函数encrypt和decrypt实现了上述的数据处理流程。

@staticmethod

def encrypt(p, k):

# 子密钥生成

k1, k2 = SDES.subkey(k, SDES.P10\_TABLE, SDES.P8\_TABLE)

# IP置换

p = SDES.permute(p, SDES.IP\_TABLE)

l0, r0 = p[:4], p[4:]

# 第一轮Feistel

l1 = r0

f\_result = SDES.function(r0, k1, SDES.SBOX0, SDES.SBOX1, SDES.P4\_TABLE)

r1 = format(int(l0, 2) ^ int(f\_result, 2), '04b')

# 第二轮Feistel

f\_result = SDES.function(r1, k2, SDES.SBOX0, SDES.SBOX1, SDES.P4\_TABLE)

r2 = format(int(l1, 2) ^ int(f\_result, 2), '04b')

# IP^(-1)置换

return SDES.permute(r2 + r1, SDES.IP\_NI\_TABLE)

**2.3 ASCII字符串加解密**

额外实现了基于ASCII码的字符串加解密。将每个字符的ASCII码表示的整数转换为8位二进制,进行SDES加密后再转换回整数,即可实现字符串的加解密。

@staticmethod

def encrypt\_asc(p, k):

mi = ""

for char in p:

asc\_in = ord(char)

in\_binary = format(asc\_in, '08b')

out = SDES.encrypt(in\_binary, k)

asc\_out = int(out, 2)

mi += chr(asc\_out)

return mi

@staticmethod

def decrypt\_asc(c, k):

ming = ""

for char in c:

asc\_in = ord(char)

in\_binary = format(asc\_in, '08b')

out = SDES.decrypt(in\_binary, k)

asc\_out = int(out, 2)

ming += chr(asc\_out)

return ming

**三、算法分析**

**3.1 安全性**

相比DES,SDES具有以下安全性缺陷:

较短的密钥长度,容易通过暴力破解得到密钥。

数据块较短,存在块Padding oracle attacks。

迭代轮数仅为2轮,可以通过穷举破解。

S盒较少,相关密钥攻击可行。

所以SDES仅用于教学演示,不应在实际中使用。

**3.2 效率**

SDES仅有2轮迭代,相比DES的16轮来说,计算复杂度降低了近一个数量级,效率有大幅提升。

**3.3 可理解性**

SDES保留了DES的基本结构,但各个组件规模减小,更容易理解原理和实现。所以常作为教学演示示例,以解释现代对称密钥密码体制的基本思想。