DOI:10.13229/j.cnki.jdxbgxb2012.06.018

改进的国际数据加密算法的子密钥扩展算法

胡 亮[1],闫智佳[2],初剑峰[1],袁 巍[1],徐小博[1]

(1.吉林大学 计算机科学与技术学院,长春 130022; 2.吉林大学 软件学院,长春 130022)

摘 要:提出了一种改进的国际数据加密算法(International data encryption algorithm, IDEA)子密钥扩展算法。该算法采用伪随机序列产生具有无序性的子密钥,令攻击者无法有效地分析子密钥中初始密钥位的位置,也无法确定弱密钥的位置。无序的子密钥破坏了针对性攻击的攻击条件,使这些攻击无效。在只有伪随机序列产生的子密钥中,对初始密钥使用频率的不同会导致新弱密钥类的产生,因此引入线性探测再散列来防止该现象的发生。对该算法进行的验证性攻击实验及安全性、效率性分析都表明该算法是安全高效的。

关键词:计算机应用;弱密钥;伪随机序列;线性探测再散列

中图分类号:TP309.7 **文献标志码**:A **文章编号**:1671-5497(2012)06-1515-06

Cryptanalysis and improvement on subkey extendable algorithm of IDEA

HU Liang[1], YAN Zhi-jia[2], CHU Jian-feng[1], YUAN Wei[1], XU Xiao-bo[1]

(1. *College of Computer Seience and Technology, Jilin University, Changchun* 130022, *China*; 2. *College of Software, Jilin University, Changchun* 130022, *China*)

Abstract: This paper improves the subkey extendable algorithm of International Data Encryption algorithm (IDEA). This algorithm employs the pseudo-random sequence to implement the randomness of the subkey. So the attacker can not analyze the position of the initial key in subkeys, and the attacker can not ensure the position of the weak key. In the process of subkey generation using pseudo-random sequence, the linear probing rescattering is imported to avoid the occurrence that different frequencies of initial key may cause new weak key. Testable attacking experiment and analysis of security and efficiency show that such improvement can enhance the safety and efficiency.

Key words: computer application; weak key; pseudo-random sequence; linear probing rescattering

在早期的分组加密中,数据加密算法(Data encryption standard, DES)能够有效地保证网络信息安全,但随着计算机系统能力的不断发展,56 bit DES 的安全性不断降低。随后人们提出了大量的替代算法,其中瑞士联邦技术学院 Lai 和 Massey[1] 于 1990 年提出了建议标准算法(Proposed encryption standard, PES)。然后于 1992 年对该算法进行了改进[2],强化了抗差分分析的能力,改称为国际数据加密算法(International data encryption algorithm, IDEA)。128 bit IDEA 提供了足够大的

收稿日期:2011-09-30.

基金项目:"973"国家重点基础研究发展计划项目(2009CB320706);"863"国家高技术研究发展计划项目(2011AA010101);国家自然科学基金项目(61103197, 61073009);吉林省重大科技攻关项目(2011ZDGG007).

作者简介:胡亮(1968-),男,教授,博士生导师. 研究方向:网格计算与网络安全. E-mail: hul@mail.jlu.edu.cn

通信作者:徐小博(1978-),男,博士研究生. 研究方向:网络安全. E-mail: xuxiaobo111@sohu.com

密钥空间,以现阶段计算机系统的能力,无法对IDEA采用暴破方法进行破解,但是IDEA子密钥扩展算法仍然存在很多缺点。文献[3]采用基于相关密钥和差分线性对IDEA算法进行攻击,发现了IDEA算法的一些弱密钥类。另外,近年来又出现了针对子密钥扩展算法特点的攻击,其中以对5轮IDEA算法的攻击最为显著,该种攻击导致IDEA算法安全性大大降低,因此,有人对IDEA算法提出了一些改进意见。例如,文献[4]提出了变长密钥的IDEA子密钥扩展算法,但是变长密钥在产生子密钥的过程中空间复杂度增多,同时带来计算量的增加。文献[5]提出了采用变长初始密钥和加长明文分组的方法来加强IDEA算法的依赖性,但并没有改变子密钥扩展算法,这样也不能从根本上解决IDEA的弱密钥分析问题。

在上述算法的基础上,本文提出了一种改进的IDEA子密钥扩展算法,采用伪随机序列确定初始密钥在子密钥中的位置,破坏子密钥中初始密钥的规律性,使针对性攻击无效,同时采用简单的数据操作防止空间复杂度及时间复杂度增加。

1 IDEA子密钥扩展算法

1.1 IDEA算法

IDEA算法采用基于相异代数群上的混合运算的设计思想,在其迭代加密运算中应用了3种核心运算:16位整数的模2^{16}加法、16位整数的模$2^{16}+1$乘法以及16位分组的按位异或运算[6-9]。

IDEA算法在加密时会将输入的64 bit的数据分成4个16 bit的子分组:X_1,X_2,X_3,X_4。这4个子分组即为第1轮的输入。每轮结束之后,第2和第3个子分组相互交换,此时的子分组作为输入进入到下一轮迭代运算中。图1为IDEA的第1个循环,其中,W_{12}表示经过第1轮加密后产生的第2个输出分组。

当8轮迭代运算结束之后,需要用剩余的4个子密钥对子分组进行输出变换:①X_1和第1个子密钥相乘;②X_2和第2个子密钥相加;③X_3和第3个子密钥相加;④X_4和第4个子密钥相乘。最后将这4个子分组重新连接即为输出密文。

1.2 子密钥扩展算法

之前提到了IDEA子密钥扩展算法存在问题,下面介绍标准IDEA子密钥扩展算法具体实现。

IDEA是根据初始密钥通过子密钥扩展算法

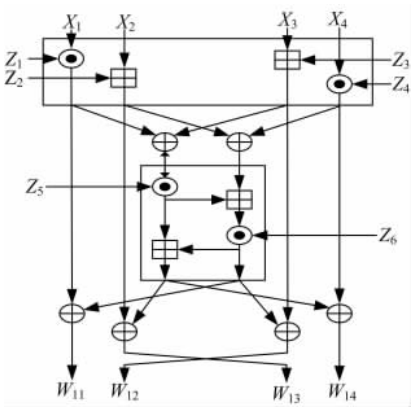

图1 IDEA的单个循环(第1个循环)

Fig. 1 A single loop of IDEA(first loop)

获得子密钥。取初始密钥的前96 bit为第1轮子密钥K_1~K_6(每个密钥16 bit)。第2轮首先使用第1轮没有使用的32(96~127)bit密钥,所以第2轮子密钥还需要64 bit。IDEA算法规定将对初始密钥进行密钥移位。密钥移位是指将初始密钥循环左移25位,即初始密钥的第24位移到第0位[8-9],再重新对移位的初始密钥进行取值。标准IDEA子密钥扩展算法产生的子密钥位置图如表1所示,其中,r为轮次数;Z_n表示该密钥分组为该轮次的第n个子密钥。

表1 标准IDEA子密钥位置图

Table 1 Standard IDEA sub-key location

r	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
1	0~15	16~31	32~47	48~63	64~79	80~95
2	96~111	112~127	25~40	41~56	57~72	73~88
3	89~104	105~120	121~8	9~24	50~65	66~81
4	82~97	98~113	114~1	2~17	18~33	34~49
5	75~90	91~106	107~122	123~10	11~26	27~42
6	43~58	59~74	100~115	116~3	4~19	20~35
7	36~51	52~67	68~83	84~99	125~12	13~28
8	29~44	45~60	61~76	77~92	93~108	109~124
9	22~37	38~53	54~69	70~85	—	—

从表1可以看出:通过标准IDEA子密钥扩展算法产生的子密钥中初始密钥位的位置是固定的,在知道加密轮次和明文的对应关系时,就可以确定明文与密文之间的关系。

2 对IDEA子密钥扩展算法的分析及攻击

2.1 IDEA弱密钥分析

通过标准IDEA子密钥扩展算法产生的子密

钥是存在问题的，因为对初始密钥进行的移位操作是线性的，子密钥中各个密钥位与初始密钥中的密钥位存在规律的对应关系，根据算法的轮次和迭代运算可以推断出密钥和密文、密文和明文及密钥和明文之间的相互关系，导致扩散和混淆的局限性[10-12]。

在文献[10]中也证实了大量弱密钥的存在。在 IDEA 中存在 3 种密钥类型：第 1 类(2^{23})是有一个线性因子的密钥；第 2 类(2^{35})是存在一个概率为 1 的环形阶；第 3 类(2^{51})是可以满足一组 12 个变量的 16 个非线性波尔等式的密钥。如果加密算法使用第 3 类密钥，则有高效的破解方法。子密钥的密钥空间减少，所以 IDEA 算法的实际抗强力破解能力不如理论上的估计。

2.2　针对 IDEA 子密钥扩展算法的攻击

在文献[13]中，作者依据 IDEA 子密钥扩展算法的特点，对 5 轮的 IDEA 进行攻击。根据 IDEA 的密钥扩展算法中，在 128 位的密钥中的第 18～24 位出现在子密钥 $Z_2^1, Z_4^3, Z_5^4, Z_5^5, Z_5^6, Z_6^6, Z_6^7, Z_1^9$ 中，而在前 5 轮加密过程中，只有 $Z_2^1, Z_4^3, Z_5^4, Z_5^5$ 中包含以上密钥位。

攻击方法如下：假设(K, K')是只有在 18～24 bit 不同的密钥对，通过明文对 (X, X') 加密，可以得到相同的中间值$(Y_1^1, Y_2^1, Y_3^1, Y_4^1)$ 和$(Y_1^{1'}, Y_2^{1'}, Y_3^{1'}, Y_4^{1'})$。根据密钥扩展算法可知，在第3轮子密钥 Z_4^3 之前没有非零差分，所以在经过该轮子密钥生成算法变换后$(Y_1^3, Y_2^3, Y_3^3, Y_4^3)$ 和$(Y_1^{3'}, Y_2^{3'}, Y_3^{3'}, Y_4^{3'})$ 的前 3 个子块相同，只有第 4 个子块不同。由 IDEA 的加密过程可知 $lsb(X_2^3 \oplus X_3^3)$ 和 $lsb(X_2^{3'} \varepsilon X_3^{3'})$ 相同，$lsb(X)$ 是指变量 X 的最低比特位。根据密钥的关系可知，猜测第 4 轮和第 5 轮的部分子密钥，解密对应的密文对 $(X^5, X^{5'})$。对于密钥猜测的正确性可以依据式(1)确定：

$$lsb\{X_2^5 \oplus X_3^5 \oplus [Z_5^5 \odot (X_1^5 \oplus X_2^5)] \oplus [Z_5^4 \odot (X_1^4 \oplus X_2^4)]\} = lsb\{X_2^{5'} \varepsilon X_3^{5'} \varepsilon [Z_5^{5'} \odot (X_1^{5'} \varepsilon X_2^{5'})] \varepsilon [Z_5^{4'} \odot (X_1^{4'} \varepsilon X_2^{4'})]\} \quad (1)$$

如果式(1)所示的等式成立，则猜测的部分子密钥是正确的。

文献[13]指出，在筛选过程中对每个密钥都要分析 m 个明文对，对错误的密钥，验证时计算结果是随机的，所以一个错误密钥被保留下来的概率是 2^{-m}，而攻击的 5 轮 IDEA 算法要猜测 64 bit 密钥。隐藏所有的错误密钥都排除的概率是 $(1-2^{-m})^{2^{64}} \approx e^{-2^{64-m}}$，取 $m = 72$，则以约为 1 的概率排除搜友的错误密钥，即可以得到正确密钥。攻击的数据复杂度为 $2^{23.2}$，计算复杂度为 $2^{70.5}$。

3　对 IDEA 密钥扩展算法的改进

3.1　预备知识

3.1.1　伪随机序列

伪随机序列不是真正的随机序列，通过一个确定的程序，对给定的随机种子进行操作，产生一个具有计算不可辨性的比特序列。伪随机序列的性质如下：①在有限长或者一个周期内，各个元素出现的个数相差不超过 1，即出现的概率相等。②在一个 P 元序列里出现 L 个相同值的概率为 $1/P^L$。③在一个序列中，任意两个序列数之间没有相关性。

3.1.2　线性探测再散列

线性探测再散列是处理哈希表冲突的一种方法，假设哈希表的地址集为 $0 \sim (n-1)$，冲突是指由关键字得到的哈希地址为 $j(0 \leqslant j \leqslant n-1)$ 的位置上已经存有记录，则处理冲突就是为该关键字找到一个“空”的哈希地址。

线性探测再散列的计算公式如下：

$$H_i = [H(key) + d_i] \bmod m \quad (2)$$
$$i = 1, 2, \cdots, k; k \leqslant m-1$$

式中：$H(key)$ 为哈希函数；key 为关键字；m 为哈希表表长；d_i 为增量序列；$d_i = 1, 2, \cdots, m-1$。

3.2　改进的子密钥扩展算法

在 IDEA 加密、解密的整个过程中，算法没有直接使用全部的 128 bit 初始密钥，而是通过子密钥扩展算法产生子密钥，同时在迭代运算过程中没有两个子密钥之间进行运算，这就说明 52 个子密钥是相对独立的。

要使 IDEA 中不存在弱密钥，就要改进子密钥扩展算法，使改进算法具有非线性特点，同时保证初始密钥使用频率的相同。随机序列不可能均匀地映射到初始密钥，需要对冲突初始密钥位进行线性探测再散列。要是使算法能判断出初始密钥是否已经加入到子密钥中，就要求初始密钥具有标志位。

对于上述描述，具体算法如下：

输入：随机序列 $R[832]$，128 bit 带有标志位初始密钥 $M[128][2]$。数组第 1 列为初始密钥；第 2 列为标志位，且标志位初始值为 0。使用随

机序列计数 $Count=0$,一轮初始密钥已使用计数 $N=0$,线性探测再散列中增量值 i。

输出:52 个 16 bit 子密钥 $Z[52][16]$。

Step1 取一位随机序列 $R[Count]$。

Step2 若 $M[R[Count]][1]=0$,执行 Step3。否则执行 Step4。

Step3 用初始密钥 $M[R[Count]][0]$ 将子密钥 $Z[Count/16][Count\%16]$ 赋值,将该初始密钥标志位置 1,执行 Step6。

Step4 线性探测再散列中增量值 $i=0$。执行 Step5。

Step5 增量值 $i+1$,若 $M[R[(Count+i)\mathrm{mod}127]][1]=0$,则用初始密钥 $M[R[(Count+i)\mathrm{mod}127]][0]$ 将子密钥 $Z[Count/16][Count\%16]$ 赋值,将该初始密钥的标志位置 1,执行 Step6,否则再执行 Step5。

Step6 一轮初始密钥中已使用计数 N 加 1,已使用随机序列计数 $Count$ 加 1。若 $N=127$ 则执行 Step7,否则执行 Step8。

Step7 将 128 位初始密钥的标志位均置为 0,再执行 Step8。

Step8 若 $Count=831$,则算法结束,否则执行 Step1。

当算法结束时,根据初始密钥(128 bit)产生了全部 52 个子密钥(832 bit 密钥位),初始密钥位使用频率也近似,初始密钥均匀地分布在子密钥中,且具有随机性,没有规律性。

4 安全性分析

4.1 抵抗已有攻击方法的分析

在改进子密钥扩展算法中加入随机序列,子密钥中各密钥位可能是任意一个初始密钥位,改变了密钥生成的调度,这样产生的子密钥中初始密钥的位子不固定,消除子密钥中密钥位的规律性,不会产生一个固定的子密钥位置图,而在文献[10]中对 IDEA 进行弱密钥分析的前提条件是有表 1 的存在,这样就不能采用文献[10]中方法进行弱密钥分析,避免弱密钥的出现。

下面用文献[13]中的攻击方法对改进子密钥扩展算法进行攻击,假设攻击前提条件相同,当采用改进的子密钥扩展算法时,初始密钥随机的加入到子密钥中,根据 18～24 bit 初始密钥会出现在不同的子密钥中,将产生如下几种情况。

(1)当 18～24 bit 初始密钥出现在第 2 轮子密钥中或者分布在第一轮前 4 个不同的子密钥中时,将不满足攻击条件:前 5 轮只在 Z_2^1,Z_4^3,Z_5^4,Z_5^5 中包含以上密钥位,这样在第 3 轮加密结束时将会出现多个差分子块,需要猜测的密钥位将增加,所以攻击方法对改进的子密钥扩展算法无效。

(2)当 18～24 bit 初始密钥出现在第一轮子密钥 Z_5^1,Z_6^1 中时,根据图 1 所示的 IDEA 加密过程可以看出 Z_5^1,Z_6^1 会影响两个输出块,这样在第 1 轮结束之后就会产生两个不同的字块,这样也不能满足攻击方法要求的条件,使攻击方法对改进的子密钥扩展算法无效。

(3)当 18～24 bit 初始密钥分布在第 3 轮前 4 个不同子密钥中时,经过第 3 轮加密操作之后将产生多个差分子块,这样也将破坏攻击方法的攻击条件,使攻击方法对改进的子密钥扩展算法无效。

由于改进的子密钥扩展算法采用了随机操作,要使攻击有效,18～24 bit 初始密钥必须同时出现在第 1 轮前 4 个子密钥中的一个子密钥中,同时在第 3 轮时,18～24 bit 初始密钥也要在前 4 个子密钥中的一个中,才能满足攻击条件,改进的子密钥扩展算法采用随机操作,这样每个初始密钥出现在子密钥中的位置是随机的,这就将是一个概率问题,即 18～24 bit 初始密钥出现在第一轮前 4 个子密钥中同一个的概率为 $4C_{16}^6/C_{128}^6$,经过计算可知为 5.9×10^{-6},18～24 bit 出现在第 3 轮前 4 个子密钥中的概率和上述概率相同,所以攻击方法对改进的子密钥扩展算法有效的概率仅为 3.481×10^{-11},因此改进的子密钥扩展算法可以有效地防范针对性攻击。

4.2 未引入新的攻击方法的分析

(1)改进的 IDEA 算法在子密钥扩展算法中加入了随机序列和线性探测再散列两个操作。由于改进的 IDEA 算法只对子密钥扩展算法进行了修改,并没有对加密过程进行修改,所以 IDEA 抵抗差分分析的强度没有改变。而加入了随机操作之后,破坏了规律性,消除了子密钥中的弱密钥,使得 IDEA 具有更好的安全性。对于现有针对标准子密钥扩展算法特点的攻击方式将没有效果,所以改进算法比标准算法更优秀。

(2)在改进的子密钥扩展算法中,子密钥的产生不再采用初始密钥移位,而是通过随机序列产生,将随机序列映射到初始密钥上,将映射的初始

密钥位加入到子密钥中。但是随机序列与随机种子有关，当算法运行过程中无法控制产生的随机序列值，在随机序列中不可避免地会出现重复的随机值或者取余相同的情况下进行映射时，会造成对初始密钥的使用频率不同，导致新弱密钥类型的出现。线性探测再散列的引入很好地解决了映射过程中产生的冲突问题和对初始密钥的使用频率问题。

一位子密钥位的选取与之前选取所有的子密钥有关，加入子密钥的初始密钥彼此之间相互影响，位置不再固定，同时保持了标准 IDEA 算法对初始密钥使用频率相同的优点，所以不再是简单的线性关系。因此无法应用文献[7]中提到的方法寻找弱密钥。所以改进的子密钥扩展算法将不会出现弱密钥，消除了 51 位弱密钥位，提高了 IDEA 算法的安全性。

(3)通过对 IDEA 子密钥扩展算法的改进，子密钥位可以是任意初始密钥，这样上述的攻击方法将没有作用。假设两组初始密钥中某些密钥位不同，在改进的子密钥扩展算法中，不同的初始密钥被随机分配到子密钥中，这样不同的初始密钥位就会分散出现在不同子密钥中，加密过程中就无法分析密钥与明文、密文的关系，从而也就无法应用文献[8]对密钥进行猜测，导致攻击方法的失效。

(4)改进的子密钥扩展算法通过随机序列产生子密钥，这样通过随机序列产生的子密钥也具有随机性。算法通过 3 种核心计算组合实现扩散和混淆的实现，对于标准 IDEA 子密钥扩展算法来说，初始密钥位在子密钥中的位置固定，根据加密算法流程可以推断出密钥作用的明文，影响混淆和扩散的作用。改进的子密钥具有随机性，对于不同随机序列将产生不同的子密钥，当攻击者对密钥与明文的关系进行分析时，可以简单地更换随机种子，更改随机序列，进而更改子密钥中初始密钥的位置，攻击者想取得密钥与明文的对应关系，就必须再次进行分析。

(5)随机序列可能多次映射到一个或者多个初始密钥位，这就会造成对初始密钥使用频率的不同，线性探测再散列可以处理映射产生的“冲突”，保证初始密钥使用频率的相同。而且线性探测再散列可以保证在一定范围内不会出现重复的初始密钥，当加入到子密钥中的初始密钥被再次映射到时，线性探测再散列就会顺序查找没有子密钥的初始密钥。子密钥位之间的关系更加复杂，无法推断初始密钥位出现在子密钥中的位置，防止了针对同一初始密钥位进行特殊明文测试攻击。线性探测再散列的加入解决了映射冲突问题，而且保证了初始密钥使用频率的相近。

改进的子密钥扩展算法在原有算法的基础上，只是增强了初始密钥的随机性，使在线性分析和非线性分析之间实现更好的平衡，所以改进的子密钥扩展算法不会引入新的未知的攻击。

5　效率分析

改进子密钥算法产生子密钥的步骤为：

(1)产生伪随机序列；

(2)将随机数对应的初始密钥位加入到子密钥中；

(3)如果初始密钥位已经加入到子密钥中，将进行线性探测再散列操作。

随机操作将会产生一个 832 bit 的随机序列，这个步骤的时间复杂度为 $O(n)$(标准子密钥扩展算法的时间复杂度)。将随机序列对应初始密钥加入到子密钥中是一个简单的赋值操作，这与算法中需要子密钥位数有关，因此该步骤的时间复杂度也为 $O(n)$。对于线性探测再散列操作的时间复杂度则与随机序列有关，这里讨论最坏和正常两种情况下的时间复杂度。在最坏情况下，随机序列都映射到同一个初始密钥位，要通过线性探测再散列操作将密钥位分配到各子密钥中，这样将执行 $(n^2-n)/2$ 次线性探测再散列操作，该步骤的时间复杂度为 $O(n^2)$，综合上面两个步骤的时间复杂度，可知改进算法最坏情况下的时间复杂度为 $O(n^2)$。在正常情况下，由伪随机序列性质(1)(2)可知，伪随机序列不会出现大量的重复值，这样每一位初始密钥需要进行线性探测再散列操作概率将趋近于一个常数 L_s，所以产生全部子密钥过程中线性探测再散列操作次数为 $L_s n$，该步骤的时间复杂度为 $O(n)$，因此在正常情况下，改进算法的时间复杂度为 $O(n)$。

与文献[4]提出的改进算法相比，本文算法虽然时间复杂度一致，但是不用对给定的变长初始密钥进行移位、增加或删除等操作，减小了时间复杂度，同时节省了物理空间，操作更简便。文献[5]中的改进算法也采用伪随机序列产生少量的随机数，但其采用的是对给定的初始密钥字符进行选择，这样不能避免弱密钥问题的产生。

通过上述对改进算法进行效率分析,对于正常情况下,改进的算法与标准 IDEA 算法的时间效率相同,但是改进算法却消除了弱密钥的出现,而且子密钥具有随机性和最大限度的均匀使用概率。由于只是对子密钥扩展算法进行改进,并没有对加解密部分进行修改,同时改进的子密钥扩展算法产生与标准子密钥算法相同格式的输出,所以不会对加解密过程中的时间复杂度产生影响,改进算法的效率是可以保证的。

6 结束语

通过对 IDEA 子密钥扩展算法的研究和对实际需求的考虑,找到了一种较好的子密钥生成算法。既可以避免针对 IDEA 子密钥的攻击,又可以保持算法的高效性,同时又保证了加解密运算的正确性。伪随机序列可以使子密钥具有随机性,这样子密钥就不再具有规律性,初始密钥对明文和密文有扩散的效果。通过线性探测再散列的加入将充分且均匀地利用全部初始密钥。因此在 IDEA 子密钥扩展算法中加入随机序列和线性探测,就可以使子密钥随机生成而不再是简单的顺序移位,防止了弱密钥的出现,破坏了初始密钥之间的位置关系,消除了 IDEA 算法中弱密钥的问题,而且改进的子密钥扩展算法可以有效地预防针对标准子密钥扩展算法对初始密钥位置规律性的攻击。

参考文献:

[1] Lai Xue-jia, Massey J L. A proposal for a new block encryption standard[C]// Eurocrypt'90 Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology, New York, USA 1991: 389-404.

[2] Lai Xue-jia, Massey J L. Hash function based on block ciphers[C]// Lecture Notes in Computer Science, 1993, 658: 55-70.

[3] Hawkes P. Differential-linear weak key classes of IDEA[J]. Lecture Notes in Computer Science, 1998, 1403:112-126.

[4] 杨维忠,李彤. 变长密钥的 IDEA 算法的研究与实现[J]. 计算机工程,2004,30(9):139-141.
Yang Wei-zhong, Li Tong. Study and implementation of IDEA of the variable length keys[J]. Computer Engineering,2004,30(9):139-141.

[5] 吴伟彬, 黄元石. IDEA 算法的改进及其应用[J]. 福州大学学报:自然科学版,2004(增刊 1):28-31.
Wu Wei-bin, Huang Yuan-shi. The improvement of IDEA algorithm and its application[J]. Journal of Fuzhou University(Natural Science),2004(Sup. 1): 28-31.

[6] 张青凤,殷肖川,李长青. IDEA 算法及其编程实现[J]. 现代电子技术,2006(1):69-71.
Zhang Qing-feng, Yin Xiao-chuan, Li Chang-qing. Principle and implementation of the IDEA algorithm [J]. Modern Electronics Technique, 2006(1): 69-71.

[7] Lai X J, Massey J L, Murphy S. Markov ciphers anddifferential cryptanalysis[C]// Proceedings of the 10th Annual International Conference on Theory and Application of Cryptographic Techniques, Heidelberg,1992:17-38.

[8] Schneier B. 应用密码学[M]. 吴世忠译. 北京:机械工业出版社, 2000.

[9] Stallings W. 密码编码学与网络安全:原理与实践[M]. 第 4 版. 孟庆树,王丽娜,傅建明,等译. 北京:电子工业出版社,2001.

[10] Daemen J, Govaerts R, Vandewalle J. Weak keys for IDEA[C]// Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology, New York, USA, 1993:224-231.

[11] Meier W. On the security of the IDEA block cipher [J]. Lecture Notes in Computer Science, 1994, 765:371-385.

[12] Hawkes P, O'Connor L. On applying linear cryptanalysis to IDEA[J]. Lecture Notes in Computer Science, 1996, 116:105-115.

[13] 鲁林真,陈少真. 对 5 轮 IDEA 算法的两种攻击[J]. 北京大学学报,2010,46(5):731-735.
Lu Lin-zhen, Chen Shao-zhen. Two attacks on 5-round IDEA[J]. Acta Scientiarum Naturalium Universitatis Pekinensis,2010,46(5):731-735.