DOI:10.13229/j.cnki.jdxbgxb2009.01.038
第39卷 第1期 吉林大学学报(工学版) Vol.39 No.1
2009年1月 Journal of Jilin University (Engineering and Technology Edition) Jan.2009

单向性策略与AES密钥生成算法的改进

胡 亮,袁 巍,于孟涛,初剑峰,刘 方

(吉林大学 计算机科学与技术学院,长春 130012)

摘 要:介绍了Mars和AES最终算法Rijndael的密钥生成算法。通过与DES比较,分析了新一代分组密码的密钥生成算法的设计思路。通过研究AES在密钥设计方面存在的不足,提出了一种新的设计策略,并应用该策略对AES的密钥生成算法进行了具体的改进,分析表明这种改进既可以提高原算法安全性,又不牺牲其效率。

关键词:计算机系统结构;单向性;密钥生成;Rijndael;AES
中图分类号:TP309.7 文献标识码:A 文章编号:1671-5497(2009)01-0137-06

One-way property strategy and improvement of key generation algorithm of Rijndael

HU Liang, YUAN Wei, YU Meng-tao, CHU Jian-feng, LIU Fang
(*College of Computer Science and Technology, Jilin University, Changchun* 130012, *China*)

Abstract: The key generation algorithm of Rijndael and Mars was introduced. By comparison with Data Encryption Standard (DES), the concept to design new key generation algorithm of block ciphers was analyzed. The weaknesses of the key generation design of Rijndael were investigated, and a new designing strategy was developed, which can be used to improve the key generation algorithm. Analysis shows that such improvement can enhance the safety of the original algorithm without reducing its efficiency.

Key words: computer systems organization; one-way property; key generation; Rijndael; AES

分组密码由于加解密速度快,可以传输大量的加密数据而在现实生活中有着广泛的应用。一个好的分组密码既要能提供高安全性,又要有很快的加密速度。这对密钥生成算法的设计提出了很高的要求。通过对现有的优秀的生成算法的研究,既可以提高现有算法的安全性,又可以总结规律,为设计新的密钥生成算法提供宝贵的经验。作者选择两个最有代表性的算法AES[1-2]和Mars[3]的密钥生成算法进行分析,通过分析它们如何克服要被取代的DES算法中存在的不足,找出优秀的算法在解决问题时的设计思路,总结它们的优点和不足之处,对现有的设计进行改进,以期获得具有代表性的结论。

1 算法介绍

1.1 AES算法的密钥生成

针对AES的密钥生成算法进行简要说明,以10轮为例(12、14轮原理相同)。AES初始密钥

收稿日期:2008-03-21.
基金项目:国家自然科学基金项目(60873235,60473099);教育部新世纪优秀人才支持计划项目(NCET-06-0300);吉林省科技发展计划项目(20080318).
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:袁巍(1984-),男,硕士研究生.研究方向:网络与信息安全.E-mail:yuanwei1@126.com

为 128 位,作为种子密钥填入前 4 个字 w_0、w_1、w_2、w_3。成为第一轮的轮密钥。然后,由 w_0 和 w_3 生成 w_4,由 w_1 和 w_4 生成 w_5,由 w_2 和 w_5 生成 w_6,由 w_3 和 w_6 生成 w_7。生成的 w_4、w_5、w_6、w_7 作为第 2 轮轮密钥。再以第 2 轮为基础继续下推,直到得到全部密钥。其中每轮轮密钥的第一个字要进行一次复杂化运算,由一个 S 盒函数 SubWord、一个移位运算函数 RotWord 和一个轮常量异或运算组成。每一轮的运算只依赖上一轮,无限下推可得到所需任意轮的密钥,故本文只介绍基本的 10 轮算法,其伪代码描述如下:

```
SubKeyGen(byte key[16], word w[44])
{   Word temp;
    for(i=0;i<4;i++) w[i]=key[4i..(4i+3)];
    for(i=4;i<44;i++)
    { temp=w[i-1];
      if(i mod 4=0) temp=SubWord(RotWord(temp))⊕Rcon[i/4];
      w[i]=w[i-4]⊕temp
    }
}
```

1.2 Mars 算法的密钥生成

Mars 共 32 轮,其中前 8 轮和最后 8 轮是专门针对差分攻击设计的,没有子密钥参加运算。中间 16 轮需要子密钥,每轮需要 2 个字,再加上开始和结束各需要 1 个 128 位密钥,共需要 40 个字的密钥。Mars 分组为 128 位,种子密钥为 128 ~1248 位。子密钥的生成算法如下:

(1)数组 k[0, n-1] 用来装主密钥,n 为主密钥字数,4≤n≤39,每个元素 32 位。

(2)数组 K[0, 39] 用来装子密钥。

(3)临时数组 T[-7, 39],每个元素长 32 位,T[-7, -1] = S[0, 6]。

(4)初始化 T[0, 38];

```
for i=0 to 38 do T[i]=((T[i-7]⊕T[i-2])≪3)⊕k[i mod n]⊕i
```

(5)T[39] = n。

(6)对 T 进行变换

```
repeat 7 times
  for i=1 to 39 do
    T[i]=(T[i]+S[T[i-1]∧0x01FF])≪9;
    T[0]=(T[0]+S[T[39]∧0x01FF])≪9;
end repeat
```

(7)子密钥装载

```
for i=0 to 39 do
  K[7i mod 40]=T[i]
```

(8)弱子密钥修正

设临时变量 M 为 32 位屏蔽字,M_x 表示 M 的第 x 位。设常量数组

B[ ] = {0xa4a8d57b, 0x5b5d193b, 0xc8a8309b, 0x73f9a978}

```
for i=5 to 35 do step 2
  j=K[i]∧3
  W=K[i]∨3
  M=0
```

当且仅当下列 3 个条件同时成立时,置 M_x 为 1,位 W_x 属于在 W 中连续的 10 个 0 或 1 中的某一个位。

$x>=2$,

$W_{x-1}=W_x=W_{x+1}$

```
r=K[i+3]∧0x1F
p=B[j]≪r
K[i]=W⊕(p∧M)
```

2 算法比较分析

2.1 AES 设计思想

在 DES 算法设计之初采用了把加密算法与密钥生成算法融为一体的设计思想,使得 DES 算法结构紧凑,运算快速。但随着计算机硬件计算能力的提高,DES 的 56 位密钥无法抵抗强力破解的攻击。而这种紧凑的结构又使得对算法任何修改都要破坏其原有的整体结构。

AES 在设计时,考虑到将来的扩展性,采取了如下 4 个设计原则[4]:

(1)简单性:为使 AES 可在对抗已知的密码分析方法的同时,不引入新的可能被攻击的弱点,除非有特别需要,一般不使用复杂的运算方法。

(2)模块性:每个步骤、每个模块都只是实现自身的功能,整体的 AES 按照定量的标准把各个模块组合起来。

(3)对称与平衡性:所有步骤都是可并行工作的,可以并行接受输入,并在硬件实现时对速度的差别进行折中。

(4)选择操作性:所有的步骤都可以仅用定义在 $GF(2^8)$ 上的异或基本操作完成。节约存储空间,使算法可以运行于多种平台之上。

由于 AES 具有良好的可扩展性,使我们可以修改其中的某一部分,而不影响其他部分的正常

工作。本文对 AES 密钥生成算法的改进建立在 AES 的设计思想之上。

2.2　AES 密钥生成分析

AES 采用了直接把种子密钥扩充，得到子密钥的方法。其中每个 32 位字 w_i 与它前面的第一个字 w_{i-1} 和前面的第四个字 w_{i-4} 相关，也就是说如果知道 w_{i-4} 和 w_{i-1} 就可以得到 w_i。同样地，如果知道 w_i 和 w_{i-1} 可以得到 w_{i-4}，知道 w_i 和 w_{i-4} 可以得到 w_{i-1}。虽然在密钥生成的每一轮都会把 4 的整倍数的字进行一次复杂化，但这不改变它们之间相关的特点。假设现在知道了 AES 某一轮的密钥 w_i、w_{i+1}、w_{i+2}、w_{i+3}，那么，可以通过 w_{i+2}、w_{i+3} 知道 w_{i-1}，通过 w_{i+1}、w_{i+2} 知道 w_{i-2}，通过 w_i、w_{i+1} 知道 w_{i-3}，再通过 w_{i-1}、w_i 知道 w_{i-4}，这样便得到该轮密钥前一轮的全部子密钥。也可以通过 w_i、w_{i+3} 知道 w_{i+4}，通过 w_{i+1}、w_{i+4} 知道 w_{i+5}，通过 w_{i+2}、w_{i+5} 知道 w_{i+6}，通过 w_{i+3}、w_{i+6} 知道 w_{i+7}，得到该轮密钥的后一轮子密钥。

分析易知 AES 密钥生成算法具有如下性质：

性质 **1**　密钥直接扩展，算法本身的执行具有高效性。

性质 **2**　每一轮密钥只依赖它上一轮的密钥即可生成，每生成一轮密钥即可参与加解密，对加解密算法具有即时性。

性质 **3**　如果攻击者得到一轮密钥即可得到包括种子密钥在内的全部子密钥，即子密钥与种子密钥在安全上具有等效性。

希望 AES 只具有性质 1 和性质 2，因为性质 3 降低了密钥生成算法的安全性。AES 之所以具有性质 1 的高效性，主要原因是它不借助中间变量，在种子密钥的基础上直接进行扩展。把种子作为第一轮子密钥，通过前一轮子密钥得到后一轮子密钥。一般来说，既然能够通过前一轮得到后一轮的子密钥，也就可以通过后一轮得到前一轮的子密钥，这就导致了性质 3 的出现。从前向后的推导是在正常工作中出现的行为，而从后向前的推导是进行密码分析时出现的行为。如果能够找到一种方法，既可以保持在从前向后推导时所具有的高效性，又使从后向前推导变得非常困难，就可以防止能量[5]、square[6] 等攻击方法利用 AES 密钥生成算法第 3 个性质，通过一轮密钥或一轮密钥的某些位而破解整个算法的威胁，称这种追求密钥生成过程不可逆的策略为单向性策略。由于密钥生成只是要得到在加解密过程中使用的密钥，而不需要像加密算法那样，要求必须有对应逆运算进行解密。所以，一个好的密钥生成算法就应该具有单向性。

再看 AES 要取代的 DES。DES 采用循环移位加查表操作的方式为每轮加解密运算提供轮密钥，也具有 AES 的三条性质。只是需要多存储一个表，它们几乎具有同样的安全性。由于在 AES 设计的竞赛中，当年设计 DES 的 IBM 公司也提出了一个 Mars 算法。很自然想到要参考 Mars 中的新密钥生成方法。

2.3　Mars 分析

通过表 1[7] 可以看到，与 AES 一样，Mars 也是一个具有高效性的算法。Mars 采用了用临时变量先经过步骤(1)～(6)得到 40 个子密钥，再经过步骤(7)乘 7 模 40 后装载到轮密钥中，最后用步骤(8)对密钥中可能出现固定结构的弱子密钥进行修正的方法。在临时变量中，某个字 K_i 与它之前的第 2 个字 K_{i-2} 和之前的第 7 个字 K_{i-7} 相关。但经过了装载以后，各个子密钥字之间失去了这种相关性，因为 K_{i-2}、K_{i-7}、K_i 已经被扩散到不同的轮次之中去了。如果再想通过这种关系进行递推 40 个字，就需要已知全部的 40 个字。即使得到子密钥中的某些字也无法推导出种子密钥。

表 **1**　**Smartcard** 性能比较(单位为 **1000** 指令周期)

Table 1　**Performance comparison of Smartcard**

	RAM	ROM	加密	密钥	解密
Mars	572	5468	45	21	67
Rrc6	156	1060	34	138	173
Rijndael	66	980	25	10	35
Serpent	164	3937	71	147	219
Ttwofish	90	2808	31	28	60

Mars 通过先对中间变量进行变换，然后对得到的结果进行调整，使得破解密钥所需的信息与已经知道密钥结果所掌握的信息同样多，从而让密码分析者无计可施。通过仔细观察，发现 Mars 通过对中间变量的调整打乱，使最终密钥失去了在密钥生成过程中原本的联系，从而具有了单向性。利用这个特点提出了一种改进，作为改进 1。

3　改进算法

3.1　改进算法 1

以 10 轮的 AES 算法为例，由初始 4 字密钥扩展到 44 字。其中的密钥依赖关系为：w_i 依赖于 w_{i-1} 和 w_{i-4}($i=4, 5, \cdots, 43$)。首先运行一遍

原始的密钥生成算法,然后对由种子生成的 40 个字作如下变换:

以第 2 轮和第 3 轮密钥为一组,交换 w_7 和 w_9;以第 4 轮和第 5 轮密钥为一组,交换 w_{15} 和 w_{17};以第 6 轮和第 7 轮密钥为一组,交换 w_{23} 和 w_{25};以第 8 轮和第 9 轮密钥为一组,交换 w_{31} 和 w_{33};以第 10 轮和第 11 轮密钥为一组,交换 w_{39} 和 w_{41}。

对交换后的 44 字密钥,仍以 4 个字为一轮,顺序排列,作为全部 11 轮的轮密钥。

以第 2 轮和第 3 轮密钥为例分析对换后每轮密钥之间的关系。

对换之前通过第 2 轮密钥可以推出第 3 轮密钥,根据第 3 轮密钥可以推出第 2 轮密钥。当把第 8 个字和第 10 个字对换以后,第 2 轮密钥是原来的 w_4、w_5、w_6、w_9,第 3 轮是原来的 w_8、w_7、w_{10}、w_{11}。这时再利用相关关系,通过第 2 轮只能得到第 3 轮中的 w_8,通过第 3 轮只能得到第 2 轮中的 w_4。如果攻击者得到其中的某一轮密钥,不能通过生成算法得到前一轮的全部子密钥。由于只是在原算法的基础上增加了 5 次对换,性能上基本保持了原算法的高速性,且部分提高了算法的安全性。

由于可以得到每轮的一个字,这种改进不能完全避免每轮密钥之间的相关关系,却提高了原算法的安全性。但是,这种方法最大的问题是密钥必须在全部生成以后才能进行对换,而不能像原来的生成算法那样,每生成一轮子密钥就可以同时用于加密,这就破坏了 AES 的性质 2(即时性)。从密钥生成的角度来看,虽然仍保持高效性,但却影响了加解密的速度。所以这样更改的算法不是一个完善的算法。

在充分分析了改进算法 1 的优点和缺点的基础上又提出了改进算法 2。

3.2 改进算法 2

仍以 10 轮为例,保持 AES 算法中的 SubWord、RootByte 和 Rcon 常量定义不变。首先对种子密钥进行直接扩展,然后从 4 个种子密钥开始,由 w_0 和 w_1 生成 w_4,由 w_2 和 w_3 生成 w_5,由 w_4 和 w_5 生成 w_6,由 w_5 和 w_6 生成 w_7,得到第 2 轮轮密钥 w_4、w_5、w_6、w_7,并对每轮的前两个字都使用 SubWord、RootByte 和 Rcon 进行复杂化。再由第 2 轮密钥生成第 3 个轮密钥,以此类推,可以得到需要加密的所有轮次的轮密钥。

算法的伪代码描述如下:

```
SubKeyGen(byte key[16], word w[44])
{
    for(i=0;i<4;i++) w[i]=key[4i..(4i+3)];
    for (i=4;i<44;i+=4)
    { w[i]=w[i-4]⊕w[i-3];
      w[i] = SubWord(RotWord(w[i]))⊕Rcon[i/4];
      w[i+1]=w[i-2]⊕w[i-1];
      w[i+1]=SubWord(RotWord(w[i+1]))⊕Rcon[i/4];
      w[i+2]=w[i]⊕w[i+1];
      w[i+3]=w[i+1]⊕w[i+2];
    }
}
```

算法分析:

在改进算法 2 中,轮密钥的生成算法已经具有了单向性,但同时也带来了一个问题,就是每轮密钥的内部,相关性比原算法提高了。所以,使用了两次复杂化函数,以降低每轮密钥内部的相关性。与原来的算法相比,复杂化函数运行了 2 次,效率要相对降低。

再看密钥的强度。假设某一轮的密钥已经被攻击者获取,计算攻击者已知 w_i、w_{i+1}、w_{i+2}、w_{i+3},可以猜到 w_{i-4}、w_{i-3}、w_{i-2}、w_{i-1}所花费的代价。由于 w_{i+3}只与 w_{i+1}、w_{i+2}相关,w_{i+2}只与 w_i、w_{i+1} 相关,所以它们不能用来向前推导。w_i 与 w_{i-4}、w_{i-3}相关,由于 w_i 已知,无论 w_{i-4}与之对应位为 1 或 0,可相应确定 w_{i-3},每个字长度为 32 位。可以通过 2^{32} 次穷举确定 w_{i-4}和 w_{i-3};同理,由 w_{i+1}反推 w_{i-2}、w_{i-1}也需要 2^{32} 次尝试。但是,由于当确定了 w_i 之后,聪明的攻击者可以利用算法的特点,通过 w_{i-4}和 w_{i-3}得到 w_{i-2}和 w_{i-1},就不必再利用 w_{i+1}反推 w_{i-2}、w_{i-1},所以通过一轮密钥反推上一轮密钥共需 2^{32} 次尝试。反推两轮则要再乘上 2^{32},以此类推。如果攻击者得到的是前 3 轮的轮密钥,则可以用比对 128 位初始密钥进行穷举攻击高的效率得到初始密钥,如果攻击者得到的是第 4 轮或者第 4 轮以后的轮密钥则用这种攻击方法还不如用暴力破解。

由于 Square 是利用第 4 轮之后平衡性的改变作为攻击的基础,能量攻击也很难直接确定开始一两轮的子密钥,设计的算法到第 4 轮之后猜

测的代价已经比暴力破解高。但改进算法 2 中还存在问题，就是在攻击者向前反推时，只需要确定前两个字即可用它们得到后两个字，而且算法的效率要比原算法低。于是，考虑在满足性质 1 和性质 2 的条件下，改进算法 2 能否再被改进。

对初始的 4 字密钥，原算法中的一个中间密钥字是通过本轮前一个字与前轮的一个字生成，故不具有单向性特征。改进算法 2 为了满足单向性要求而采用了用一轮的 4 个字生成下一轮的 2 个字，其余 2 个字用本轮已有的 2 个字再生成的方法，使得子密钥之间存在的是轮与轮之间的整体关系。

如果用一轮的 4 个字生成下一轮的 1 个字，则其余的 3 个字不可能由已经生成的这个字继续生成，而只能像原算法那样借助上一轮的至少一个字。这样的算法只是让攻击者反推上一轮的难度有所增加，但不具有单向性的特点。

如果用一轮的 4 个字生成下一轮的 3 个字，再通过已有的 3 个字生成该轮的第 4 个字。就会使得新生成的密钥与上一轮密钥之间存在过多的联系，使攻击者可以容易地找到一轮子密钥与其前一轮子密钥的关系。比如：有 w_i、w_{i+1}、w_{i+2}、w_{i+3}。生成算法是通过 w_i、w_{i+1}生成 w_{i+4}；w_{i+1}、w_{i+2}生成 w_{i+5}；w_{i+2}、w_{i+3}生成 w_{i+6}；再由 w_{i+4}、w_{i+5}、w_{i+6}生成 w_{i+7}得到新一轮的子密钥 w_{i+4}、w_{i+5}、w_{i+6}、w_{i+7}。则当攻击者知道 w_{i+4}、w_{i+5}、w_{i+6}、w_{i+7}时，就可以通过联立四元一次方程组求解 w_i、w_{i+1}、w_{i+2}、w_{i+3}。

同理，用 4 个字生成下一轮的 4 个字的方法也不能达到需要的单向性的算法。

所以，只能采用类似改进算法 2 中的方式，用一轮的 4 个字生成下轮的 2 个字，再由这 2 个字生成该轮的另外 2 个字的方式进行改进。改进算法 2 之所以在 4 轮之后才达到 128 位密钥强度，是因为当猜出上轮前两个字之后可以用它们确定后两个字。可能的再改进方式就只需要也只能是克服这个问题。于是，针对这个问题提出了改进算法 3。

3.3　改进算法 3

仍以 10 轮为例，保持 AES 算法中的 SubWord、RootByte 和 Rcon 常量定义不变。种子密钥先进行直接扩展，调整密钥使用顺序，从 4 个种子密钥开始，由 w_0 和 w_2 生成 w_4；由 w_1 和 w_3 生成 w_5，由 w_4 和 w_5 生成 w_6，由 w_5 和 w_6 生成 w_7，得到第 2 轮轮密钥 w_4、w_5、w_6、w_7，只对每轮的第一个字使用 SubWord、RootByte 和 Rcon 进行复杂化。再由第 2 轮密钥生成第 3 个轮密钥，以此类推，可以得到需要加密的所有轮次的轮密钥。算法的伪代码描述如下：

```
SubKeygen(byte key[16], word w[44])
{
    for(i=0; i<4; i++) w[i]=key[4i..(4i+3)];
    for(i=4; i<44; i+=4)
    { w[i]=w[i-4]⊕w[i-2];
      w[i] = SubWord(RotWord(w[i]))⊕Rcon[i/4];
      w[i+1]=w[i-3]⊕w[i-1];
      w[i+2]=w[i]⊕w[i+1];
      w[i+3]=w[i+1]⊕w[i+2];
    }
}
```

通过对改进算法 2 的分析可知，改进算法 3 从一轮子密钥反推前一轮子密钥需要猜测 2^{64} 次。同时，调整顺序后，一轮子密钥内部的相关性也大为降低，所以不必再对每轮使用 2 次复杂化函数，而是对每轮的第一个字使用一次即可。

经此改进后，算法减少了一次复杂化运算，效率与原始 AES 密钥生成算法基本相同。单向性强度从改进算法 2 的每轮 2^{32} 提高到每轮 2^{64}，使得算法运行 2 轮之后强度即可与暴力破解相同。

4　结束语

通过对密钥生成算法的研究和对需求的权衡，找到了一种较好的子密钥生成方法。既可以避免利用 AES 性质 3 的攻击，又可以保持高效性，同时又不影响加解密运算。Daemen 和 Rijmen 在设计加密算法时，为了抵抗差分[8]和线性分析[9]方法提出了宽轨迹策略[10]，使得每轮差分的传播效率最大，进而提供了加解密算法的安全性保障，却忽视了密钥生成算法的单向性。Mars 算法考虑到了单向性的要求，可在具体的实现上却无法使之与加解密的即时性相结合。通过对各种算法优点的提取，找到了改进算法 3 这样一种能够既提供单向性又不失高效性的方法。

在寻找这种密钥生成算法时，一直在追求一种一轮密钥只能用于推导下一轮密钥的方法，称之为生成子密钥的单向性策略。虽然这种策略是

在对AES的分析中提出来的,但是它也可以作为一种基本的设计思想应用到其他的分组密码的设计中。今后还将进一步研究这种生成子密钥的单向性策略在其他分组密码密钥生成算法中的应用。

参考文献:

[1] Daemen J, Rijmen V. AES proposal: Rijndael(Version2). [2005-07-26]. [EB/OL]. www.csrc.nist.gov/encryption/aes.

[2] Daemen J, Rijmen V. The Design of Rijndael: AES-The Advanced Encryption Stand[M]. Berlin: Springer-Verlag, 2002.

[3] 崔竞松, 张焕国. 高级加密标准AES候选之一—Mars算法[J]. 通信保密, 2000, 22(2): 59-66.
Cui Jing-song, Zhang Huan-guo. MARS algorithm—candidate of advanced encryption standard[J]. TONGXIN BAOMI, 2000, 22(2): 59-66.

[4] van Tilborg H C A. Encyclopedia of Cryptography and Security[M]. New York: Springer-Verlag, 2005.

[5] 吴文玲, 贺也平, 冯登国, 等. Mars和Rijndael的能量攻击[J]. 软件学报, 2002(4): 532-536.
Wu Wen-ling, He Ye-ping, Feng Deng-guo, et al. Power attack of MARS and Rijndael[J]. Journal of Software, 2002(4): 532-536.

[6] Daemen Joan, Knudsen Lars, Rijmen Vincent. The block cipher square[C] //Proceedings of Software Encryption Workshop. New York: Springer-Verlag, 1997.

[7] 赵富祥, 张福泰, 王育民. 分组加密算法设计原则及Rijndael[J]. 信息安全与通信保密, 2001, 23(10): 51-52.
Zhao Fu-xiang, Zhang Fu-tai, Wang Yu-min. The design principle of block cipher and Rijndael[J]. China Information Security, 2001, 23(10): 51-52.

[8] Biham E, Shamir A. Differential cryptanalysis of the data encryption standard[M]. New York: Springer-Verlag, 1993.

[9] Matsui M. Linear cryptanalysis method for DES cipher[C] //Proceedings of EURO-CRYPT'93, New York: Springer-Verlag, 1993.

[10] Daemen J, Rijndael V. The wide trail design strategy, cryptography and coding[C] // LNCS, 2001, 2260: 222-238.