文章编号:1001-5132（2008）01-0020-05

Rijndael 算法优化技术研究

刘　丽[1,2]，何加铭[1,3]，刘太君[1,3]，曾兴斌[1,3]

（1.宁波大学 通信技术研究所，浙江 宁波 315211；2.宁波教育学院 科技学院，浙江 宁波 315010；
3.宁波市通信芯片与射频技术重点实验室，浙江 宁波 315040）

摘要：通过深入分析 Rijndael 算法，改进了 AES 算法的几个有可能产生不安全隐患的步骤. 首先是对于最可能被攻击的混列进行优化，使该步骤变成简单的查表而不是域乘，增加了非线形安全性；其次对于子密钥的生成方面引入了随机函数使密钥的生成和选择变成随机性的，而不是原算法简单的线形选择，这样可以较完全抵抗线形密码分析的攻击.

关键词：Rijndael 算法；密钥；AES；S-盒

中图分类号：TP39　　文献标识码：A

1997 年 4 月 15 日美国国家标准技术研究所发起征集 AES(Advanced Encryption Standard)算法的活动[1,2]，并专门成立了 AES 工作组. 目的是为了确定一个非保密的、公开披露的、全球免费使用的分组密码算法，用于保护 21 世纪政府的敏感信息. AES 的要求是比三重 DES 快而且至少和三重 DES 一样安全，分组长度为 128 比特，密钥长度为 128 比特、192 比特、256 比特. 后来，NIST 从 15 个算法中筛选出 5 个 AES 候选算法，它们是 RC6、Mars、Rijndael、Serpent 和 Twofish. 最终，由比利时 Daemen J 和 Rijmen V 设计的 Rijndael 算法胜出.

1　算法实现

Rijndael 加密算法是分组长度可变、密钥长度也可变的分组密码. 分组长度、密钥长度彼此独立地确定为 128 比特、192 比特、256 比特.

1.1　字节转换

字节转换是一个以字节为单位的非线性取代运算，取代表(S-box)是经过 2 个运算过程而建立，并且是可逆的. 首先找出每个字节在 $GF(2^8)$ 中的乘法反元素；接着经过 1 个仿射(Affine)转换运算，定义如下：

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

字节取代(Subbytes)运算对 State 的影响，如图 1 所示，经过运算后形成的 S-box 如图 2 所示.

字节取代转换的反运算：计算仿射对应之后的相反运算可得到 S^{-1}-box，以此 S^{-1}-box 做字节取代

收稿日期：2007-07-14.　　宁波大学学报（理工版）网址：http://3xb.nbu.edu.cn
基金项目：国家自然科学基金（60372026）.
第一作者：刘　丽（1982－），女，重庆人，在读硕士研究生，主要研究方向：矩阵论. E-mail:cjmlx@nbu.edu.cn

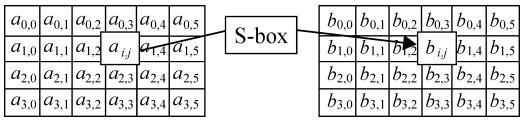

图 1　State 的字节取代运算

yx	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

图 2　字节取代运算后的 S-box

(Subbytes)即可.

1.2　移行转换

在这个转换中，State 的每一行以不同的偏移量做环状位移，第 0 行不动，第 1 行位移 C_1 个字节，第 2 行位移 C_2 个字节，第 3 行位移 C_3 个字节. 位移的偏移量 C_1, C_2, C_3 跟区块的数目(N_b)有关，其关系见表 1.

表 1　C_1, C_2, C_3 与区块数目(N_b)的关系

N_b	C_1	C_2	C_3
4	1	2	3
6	1	2	3
8	1	3	4

移行转换(Shift rows)运算对于 State 的影响，如图 3 所示. 移行转换的反运算：对第 2、第 3 及第 4 行做 N_b-C_1，N_b-C_2，N_b-C_3 个字节的环状位移即可.

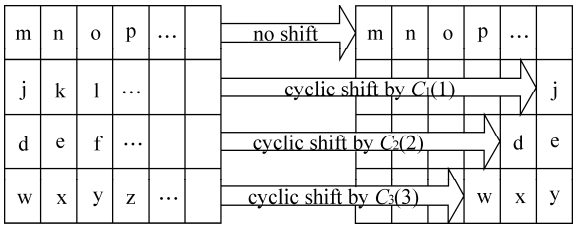

图 3　State 的移行转换

1.3　混列转换

在这个转换中，把 State 当作一个存在 GF(2^8) 中的多项式. 并且对一个固定的多项式 $c(x)$ 作乘法，如果发生溢位，则再模余 x^4+1. 表示如下：

$$c(x)='03'x3+'01'x2+'01'x+'02',$$

$c(x)$ 与 x^4+1 互质，令 $b(x) = c(x)\oplus a(x)$，以矩阵乘法表示如下：

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}.$$

State 经过混列(Mix columns)运算之后的变化如图 4 所示.

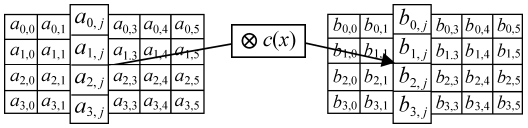

图 4　State 的混列运算

混列转换的反运算，则是乘上一个特殊的多项式 $d(x)$，

$$('03'x3+'01'x2+'01'x+'02')\oplus d(x)=$$
$$'01'\,d(x)='0B'x3+'0D'x2+'09'x+'0E'.$$

1.4　轮密钥加

这个运算主要是把每一个回合密钥(Round key)透过简单的 Bitwise exor 加入到每一个 State 中，如图 5 所示. Add round key 的逆是它自身.

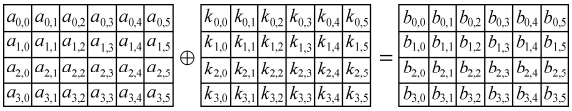

图 5　Bitwise exor 加入后的 State

2　Rijndael 算法优化

2.1　Mix columns 变换的优化技术

针对 AES 密码最可能成功的攻击来自一个允许时间选择攻击的弱实现[3,4]. 攻击者用不同的密钥并精确地测量出加密例程所需的时间. 如果加密例程被粗心编码，执行时间便依赖于密钥值，它就有可能推导出有关密钥的信息. 在 AES 中，这种事情最可能发生在 Mix columns 例程中，因为有域乘. 针对这种攻击的最稳妥的安全措施是将域乘

实现为一个查询表.

乘法所用的常量系数基于域论，并且是 0x01，0x02 或 0x03 中的任意一个值. 如给定某一列 c，其替代式如下：

State[0,c]=0x02*State[0,c]+0x03*State[1,c]+
0x01*State[2,c]+0x01*State[3,c]，
State[1,c]=0x01*State[0,c]+0x02*State[1,c]+
0x03*State[2,c]+0x01*State[3,c]，
State[2,c]=0x01*State[0,c]+0x01*State[1,c]+
0x02*State[2,c]+0x03*State[3,c]，
State[3,c]=0x03*State[0,c]+0x01*State[1,c]+
0x01*State[2,c]+0x02*State[3,c].

由以上公式计算出每一个可能的值，从而形成一张表，然后进行类似于 S-box 的砖匠置换. 所以当做 Mix columns 变换时只要查表即可，不需要用域乘，这样就增加了差分分析攻击的难度.

2.2 子密钥生成优化[5]

子密钥是由密钥导出的，这个过程包含 2 个组成部分，一个是密钥扩展，另一个是轮子密钥的选取，其基本原理为[6]：

(1) 轮子密钥的比特数总和等于分组长度乘以轮的数目加 1，即每分组长 128 比特，轮数等于 10 时，轮子密钥的比特数总和为 128×(10+1)=1 408 比特.

(2) 密钥扩充为扩展密钥(Expanded key).

(3) 轮子密钥是由这些扩展密钥中取出来的，第 1 轮子密钥由最先 N_b 个字组成，第 2 轮子密钥为其次 N_b 个字等.

2.2.1 密钥的扩充

扩充后的密钥是 1 个 4 比特的线性数组，表示为 $W[N_b \times (N_r+1)]$，前 N_k 个字组包含了加密密钥(Cipher key). 密钥扩充函式和 N_k 是息息相关的，分为 2 种情况运作，一是当 $N_k \leq 6$，另外则是当 $N_k > 6$，以伪码叙述如下：

当 $N_k \leq 6$ 时，

```
Key expansion(byte key[4×Nk] word W[Nb×(Nr+1)])
{
  for(i=0; i < Nk; i++)
  W[i]=(key[4×i], key[4×i+1], key[4×i+2], key[4×i+3]);
  for(i=Nk; i < Nb×(Nr+1); i++)
{
    temp=W[i-1];
    if (i % Nk == 0)
    temp=Subbyte(Rotbyte(temp)) ^ Rcon[i / Nk];
    W[i]=W[i－Nk] ^ temp;
}
}
```

在上面的子程序中，Subbyte(W)传回 1 个 4 比特的字组，这些字组是输入的字组经过 S-box 的转换所产生的相对字组. Rotbyte(W)则是传回经过旋转的字组.

当 $N_k > 6$时，

```
Key expansion(byte key[4×Nk] word W[Nb×(Nr+1)])
{
  for (i=0; i < Nk; i++)
  W[i]=(key[4×i], key[4×i+1], key[4×i+2], key[4×i+3] );
  for (i=Nk; i < Nb×(Nr+1); i++)
{
    temp=W[i－1];
    if (i % Nk == 0)
    temp=Subbyte(Rotbyte(temp)) ^ Rcon[i / Nk];
    else if (i % Nk == 4)
    temp=Subbyte(temp);
    W[i]=W[i－Nk] ^ temp;
  }
}
```

以上 2 种情况的相异处在于当 $N_k \leq 6$ 时，$(i-4)$ 是 N_k 的倍数时，对于 $W[i-1]$ 先执行 Subbyte，再执行 Exor. 上述回合常数定义如下：

Rcon[i]=(RC[i],‘00’,‘00’,‘00’)，

其中，RC[0]=’01’，$RC[i] = x(RC[i-1]) = x^{(i-1)}$.

由以上分析可得：

$RC[i] = randrom[]\ (RC[i-1]) = x^{(i-1)}$.

2.2.2 选择回合密钥

第 i 回合密钥是指在存在回合密钥缓冲区的字组 $w[N_b*i]$到 $w[N_b*(i+1)]$，如图 6 所示.

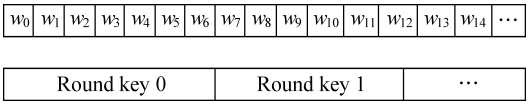

图 6　回合密钥选择过程

由以上分析可知，由于 Rijndael 算法的子密钥的前 N_k 个字完全由种子密钥填充而成，这样做虽然密钥的离散性非常好，减少了密钥间的线性关系，但是密钥的雪崩效应就减弱了，同时它的轮密钥是通过递归定义生成的，即对某一轮的密钥可以由前一轮或几轮的轮密钥推导而来. 当泄露的密钥信息达到一定程度时，其他的种子密钥或许就可以通过穷举法获得，进而获得全部轮子密钥. 对于公开的算法来说，如果密钥已知，我们轻而易举地就可以由密文译出明文，信息的保密就受到了威胁. 那么，怎样在优化线性关系和雪崩效应的同时提高密钥的保密性呢?从安全的角度分析，为了使攻击者更难于找到加密密钥特别是种子密钥，我们应该提高密钥生成算法的混淆性来有效抵抗密码攻击.

由此我们提出了在子密钥生成阶段和密钥选择阶段引入一个随机函数的方法来避免上述缺陷，具体方法如下：

第一次生成 RC[I]时，按照原来的方式利用线性同余法生成伪随机数 I，从而选择运算对象 rand[I]，具体的公式如下：

$RC[I] = (RC[I-1]) \cdot random(k)$，

其中 k 为小于 I 的任何数.

产生 random(k) 的函数：

```
Random (n, m, seed, a, b)
{
  r0=seed；
  for (i=1; i≤n; i++)
  ri=(a*ri－1+b)modm；
}
```

其中，将种子参数 seed 设为计算机当前的日期或者时间；m 是一个较大数，可以把它取为 $2w$，w 是计算机的字长；a 可以是 $0.01w$ 和 $0.99w$ 之间的任何整数；n 等于当前 $(I-1)$. 这样就可以方便地产生出随机的参与异或运算的对象 rand[I]. 密钥生成后把所有的子密钥形成一个表，而在密钥选择阶段同样可以使用随机函数，打乱子密钥选择的规律性，使生成各列子密钥的过程具有完全的随机性. 但是由于算法中用到了加密者的本地时间作为 seed，而解密者不一定与加密者时间统一，为了解决这个问题，我们决定用数字时间戳(Digital time-stamp)系统，该系统可以让通信双方遵守同样的时间，而算法中只要把取的系统时间加入到密文里的某个只有双方才知道的特殊位置(或者直接加入明文里加密也可)就可以使解密者很容易地对密文进行解密了.

3　结束语

Rijndael 算法已成为虚拟专用网、Sonet、远程访问服务器、高速 ATM／以太路由器、移动通信、电子金融业务等的加密算法，并逐渐取代 DES 在 IPSec、SSL 和 ATM 中的加密功能. 目前，IEEE 802.11i 草案已经定义了 AESJJI3 的 2 种不同运行模式，成功解决了无限局域网标准中的诸多安全问题. 在这种情形下，AES 算法的安全性及其快速实现问题显得格外突出.

本文在对 Rijndael 算法深入分析的同时，提出了子密钥生成阶段和密钥选择阶段引入一个随机函数的方法，在优化线性关系和雪崩效应的同时提

高密钥的保密性，并在生成各列子密钥的过程中，采用数值时间戳系统保证加密和解密时间同步. 当然，该方案只能应用在网络中，但是它给 Rijndael 带来的安全性是前所未有的，并且跟通信系统紧密联合在一起，在应用方面将会有很大的用武之地.

参考文献：

[1] Federal Information Processing Standards Publication 197. Specification for AES197 [EB/OL]. [2001-11-26]. http://csrc.nist.gov/publications/tips/.

[2] Daemen J, Rjjmen V. AES Proposal: Rijndael[EB/OL]. [2001-02-28]. http://csrc.nist.gov/cryptotoolkit/aes/rijndael/.

[3] Lipmaa H. Secure and efficient time stamping systems[D]. University of Tartu-estonia, 1999.

[4] 冯登国, 吴文玲. 分组密码的设训与分析[M]. 北京: 清华大学出版社, 2000.

[5] 马杏弛, 刘亮. 高级加密标准——Rijndael 算法介绍与探讨[J]. 装备指挥技术学院学报, 2003(1):96-100.

[6] 陈作新, 刘鸿雁. Rijndael 的一种改进算法及其实现研究[J]. 计算机工程与应用, 2003(2):76-80.

Optimization Study of Rijndael Algorithm

LIU Li[1,2], HE Jia-ming[1,3], LIU Tai-jun[1,3], ZENG Xing-bin[1,3]

(1.Communication Technology Institute, Ningbo University, Ningbo 315211, China;
2.Faculty of Science and Technology, Ningbo Institute of Education, Ningbo 315010, China;
3.The Ningbo Key Lab of Communication Chips & RF Technology, Ningbo 315040, China)

Abstract: Through analyzing the algorithm of Rijndael, some technical points are made in this paper with respect to improving several steps which may incur potential risks. By optimizing the most vulnerable MixColumns, the multiplication in field is replaced with using look-up tables so the security of nonlinearity is enhanced. In regard to creating the sub-key, instead of utilizing the simple linear selection as used in the original algorithm, this paper introduces the random function to generate and choose the keys so that the attacks of linear cryptanalysis can be resisted much more effectively.

Key words: Rijndael algorithm; cryptographic key; AES; S-box

CLC number: TP39 **Document code:** A

（责任编辑 史小丽）