变型的 Rijndael 及其差分和统计特性

冯国柱[1], 李 超[1,2], 多 磊[1], 谢端强[1], 戴清平[1]

(1. 国防科技大学数学与系统科学系, 湖南长沙 410073; 2. 中科院软件所计算机科学重点实验室, 北京 100080)

摘 要: 本文在原 Rijndael 算法基础上对其进行了变动和改进, 使得改动后的新算法在牺牲少许密钥装填速度的前提下, 抗差分攻击特性没有降低, 统计效果提高, 而且可以部分地抵抗 Square 攻击.

关键词: Rijndael; 差分概率; 统计; Square 攻击

中图分类号: TN911.2 文献标识码: A 文章编号: 0372-2112 (2002) 10-1544-03

Transmutative Rijndael with the Differential and Statistical Charactartics

FENG Guo-zhu[1], LI Chao[1,2], DUO Lei[1], XIE Duan-qiang[1], DAI Qing-ping[1]

(1. *Dept of Mathematic and System Science, NUDT, Changsha, Hunan 410073, China;*
2. *Laboratory of Computer Science, Institute of Software, CAS, Beijing 100080, China*)

Abstract: We improve and change the Rijndael primary algorithm. The new algorithm is better than the primary algorithm, its key setup is little slower, but is still secure against differential and truncated differential cryptanalysis. And the new algorithm's statistical effect is improved. It can counteract the square attacking to some extent.

Key words: rijndael; differential probability; statistic; Square-attacking

1 引言

1997 年 9 月, 美国国家标准技术研究所(NIST) 为了履行其法定职责, 发起了一场推选用于保护敏感的(无密级的) 联邦信息的对称密钥加密算法的活动. 1998 年 8 月, NIST 宣布接受十五个候选算法并提请全世界密码研究界协助分析这些候选算法, 包括对每个算法的安全性和效率特性进行初步检验. NIST 考察了这些初步的研究结果, 并选定 MARS、RC6、Rijndael、Serpent 和 Twofish 等五个算法作为参加决赛的算法, 经公众对决赛算法进行更进一步的分析评论, 2000 年 10 月, NIST 决定推荐 Rijndael 作为高级加密标准(AES).

Rijndael 是一种迭代分组密码, 它采用的是代替/置换网络(SPN). Rijndael 的圈函数由四层组成, 第一层(字节替换) 为非线性层, 一个 8×8 的 S 盒应用于每一个字节; 第二层(行移位变换) 和第三层(列混合) 是线性混合层, 4×4 的阵列按行位移, 按列混合; 在第四层(加圈密钥变换), 子密钥异或到阵列的每个字节.

Rijndael 密码[1] 是完全"自力更生"的, 没有使用其他密码的构成变换, 没有从声誉好的密码(如 DES, IDEA) 中借用 S 盒, 没有象 RC6 等密码算法中使用 π, φ 等数字, 同时 Rijndael 密码没有将其安全性建立在算术运算之间模糊的和不好理解的相互作用上. 但是 Rijndael 密码加解密是不一致的, 在软件实现时, Rijndael 密码及其逆密码使用不同的代码或表; 在硬件实现时, Rijndael 逆密码只能共同使用 Rijndael 密码的部分电路. 针对这一不足及 Rijndael 算法中 $m(x)$ 的选取的随意性, 我们修改了 Rijndael 算法中 $m(x)$、$c(x)$ 和 $d(x)$, 使得 $c(x)$ 和 $d(x)$ 取相同的多项式, 这样加密与解密具有更多一致性, 从理论上证明了这样的修改不影响其抗差分能力. 其次, 考虑到经典 S 盒中幂函数的抗差分能力, 用幂函数取代 Rijndael 算法中 S 盒的求逆运算, 并对修改后的算法进行 Square 攻击和统计测试, 结果表明新算法的抗 Square 攻击能力并未降低, 但统计性能更好.

2 变型的 Rijndael 算法

经过对原算法的研究以及对其安全性的分析, 我们按照不降低算法的安全性及其加解密速度的前提条件下对原算法进行了修改, 修改的原因和细节如下:

2.1 字节替换(ByteSub)

第一步: $GF(2^8)$ 中取元素 a 的乘法逆, 在所有可逆变换中, 我们认为它并不是最好的变换. 根据分组密码的设计原则[6], 并且参考 $E2$ 密码设计中的同构变换——a^e, $e=127$, 将有限域上的逆元映射变为幂函数映射, 即将求逆变换变成求 127 次幂(解密求 127 次根); 其次, 对仿射矩阵也做了实验性变换, 发现另一个含 5 个 1 的矩阵的性质并不比原算法中的矩阵差, 所以选择另一个矩阵作为仿射变换矩阵. 两个矩阵首行向量分别为: (1, 0, 0, 0, 1, 1, 1, 1, 1) 和 (0, 1, 0, 0, 1, 1, 1, 1, 1), 其余各行向量依次向右循环一位.

2.2 列混合变换(MixColumn)

列混合中, 为了使加解密共用部分电路或代码, 决定提供一种比较好的 $c(x)=d(x)$ 的算法, 而它没有明显的缺陷. 经过大量的实验仿真, 选取 $c(x)=d(x)=3+x+2x^2+x^3$, 它不仅简单, 并且性质好.

2.3 密钥扩展算法

Rijndael 算法的扩展密钥是通过密钥调度过程从含 N 个 32-bits 的字的密码密钥(种子密钥) 中获得的, 其中 $N=$ 密码密钥长度/32, 将 32-bits 的字记为 $W[\ ]$, 一共需要 $4\times(R+1)$ 个这样的字(其中 R 是加密轮数).

原算法的密钥扩展方案[1] 的性质已经很好了, 弱密钥极

收稿日期: 2001-06-28; 修回日期: 2002-01-14

基金项目: 中科院软件所计算机科学重点实验室开放基金(No. sysklf0201); 国防科技大学校基础研究基金(No. 0002)

少, 安装速度较快, 而且具有一个特殊的性质, 就是可以从任何一轮密钥推出全部的密钥(包括种子密钥). 这一性质有利也有弊, 好处在于这样可以在种子密钥变动频繁的情况下, 做到加解密同步或者并行处理; 弱点在于密钥的安全性大大降低, 极利于 Square 攻击, 只要攻击出其中一轮密钥, 就可以计算出密码密钥. 所以针对这种情况变动了密钥扩展方案, 采取了依赖数据的移位异或(非线性)方法, 使得逆推密钥变得相对困难, 一定程度上抵抗了 Square 攻击. 具体变动如下:

扩展密钥的前 W 个字 $W[0], \cdots W[N-1]$ 直接取密码密钥的 N 个字. 而对于 $i \in \{N, \cdots, 4\times(R+1)-1\}$, $W[i-1]$ 由以下方式获得:

If $(i \bmod N) = 0$

Then $W[i] := W[i-N] \oplus f(W[i-1]) \oplus rcon[i/N]$

Else if $((N>6) \text{and} (i \bmod N) = 4)$

Then $W[i] := W[i-N] \oplus g(W(i-1))$　　(1)

Else

$W[i] := W[i-N] \oplus (W[i-1] \ll W[i-N] \& 0x0F) \oplus (W[i-1] \gg W[0] \& 0x0F)$

那么容易看出, 即使已知 $W[i]$ 和 $W[i-1]$, 想得到 $W[i-N]$ 依赖于 $W[i-1]$ 和 $W[0]$, 这显然是比较困难的. 只是这样的变动将对密钥装填时间有少许影响.

3　变型算法的安全性

针对上一部分对算法的变动, 来考察变型算法的安全性.

3.1　变型算法的差分特性

文献[4]中给出了截断差分概率的计算过程. 注意到计算过程中, 最终的差分概率结果是由一个分块矩阵的秩决定的, 而分块矩阵的初等行变换不涉及到域 $GF(2^n)$ 上的乘法结构变化, 只有在计算矩阵 P 的某个子式的秩的时候才涉及到乘法结构的变化. 不失一般性, 下面讨论域 $GF(2^n)$ 上方阵 P 在不同的乘法结构下的秩.

定理　$GF(2^n)$ 上的阶 $m\times m$ 方阵 P, 当 $m=2^k$ 时 P 在不同的乘法结构下的秩相同, 其中 k 为正整数.

证明　设 $GF(2)[x]$ 中两个不同的 n 次不可约多项式 $m_1(x)$ 和 $m_2(x)$. 由这两个多项式构成两个乘法结构 $(+, \times)$ 和 $(\oplus, \otimes)$, 令 a, b 是 $GF(2^n)$ 中的两个元素, 对应的 $GF(2)$ 上的多项式为 $a(x)$ 和 $b(x)$. 设矩阵 P 的秩为 r.

当 $r=m$ 时, 不妨从简单的情况开始讨论 $m=2$, 此时可设矩阵 $P=\begin{pmatrix} a & b \\ c & d \end{pmatrix}$; 假设在乘法结构 $(+, \times)$ 下的 $|P| \neq 0$, 而在乘法结构 $(\oplus, \otimes)$ 下的 $|P|=0$, 也就是 $a(x)\times d(x) + b(x)\times c(x) \equiv 0 (\bmod m_2(x))$, 即

$$a(x)\times d(x) \equiv b(x)\times c(x) (\bmod m_2(x)) \qquad (2)$$

可以容易看出, 当将 $m_2(x)$ 换成与其互素的 $m_1(x)$, 式(2)仍然成立. 所以在乘法结构 $(+, \times)$ 下的 P 的行列式也为零. 那么假设错误, 即在乘法结构 $(\oplus, \otimes)$ 下的 $|P| \neq 0$. 因为 $m=2^k$, 一般的情况可以由归纳法证明.

当 $r<m$ 时, 在乘法结构 $(+, \times)$ 中必存在一个 r 阶的子式 Q 使得 Q 满秩, 那么在乘法结构 $(\oplus, \otimes)$ 中, 相应的子式 Q 由上一步的证明可知也是满秩的; 反之亦然.　　证毕

所以只要保证 $m(x)$ 或 $c(x)$ 是 $GF(2^n)$ 中的不可约多项式, 那么修改 Rijndael 算法中的 $m(x)$ 和 $c(x)$ 不会降低算法对截断差分分析和差分分析的抵抗性. 通过计算我们得到了 4 轮 Rijndael 的平均差分概率的上界为 $1.00 * p^{16} (= 1.065 * 2^{-128})$; 5 轮 Rijndael 的差分概率上界为 $0.940 * p^{16} (= 1.0007 * 2^{-128})$. 为了抗截断差分和差分分析, 一轮以上的 Rijndael 加密是必要的, 这时就可以避免对最后一轮的穷举搜索.

3.2　变型算法抗 Square 攻击能力

Square 攻击[2,3,5]是专门针对 SPN 结构的密码. 有关文献在理论上已经做到对 7 轮的 Rijndael 进行攻击. Square 攻击的原理参见文献[3], 这里就不赘述了. 攻击步骤如下[3]:

(1)for $X \in P_4$: $Y := MC^{-1}(X)$; $Z := SR^{-1}(Y)$;

记 2^8 个状态为 Z 集合 Q_4.

(2)for all $(i, j) \in \{0, 1, 2, 3\}^2$:

for $a \in \{0, 1\}^8$: $b(a) := \oplus_{Z \in Q_4} S^{-1}(Z_{i,j} \oplus a)$;

if $b(a) \neq 0$ then $L^4_{i,j} \neq a$. 其中 $L^r = SR^{-1}(MC^{-1}(K^r))$.

其中 P_4 为第四轮输出, L^r 是第 $r+1$ 轮密钥的 L 表示[3]. 选择 2^9 个明文组, 就可以将第四轮子密钥完全确定下来. 如果注意到原算法密钥扩展中, 如果知道了两个字 $W[i-1]$ 和 $W[i-N]$, 可以计算出字 $W[i]$. 那么可以反过来, 给定了 $W[i]$ 和 $W[i-1]$ 可以很容易计算出 $W[i-N]$. 因此, 如果知道了连续的 N 个字 $W[k]$, ..., $W[k+N-1]$ 的扩展密钥就足以计算出所有的扩展轮密钥, 也就很容易将最终的密码密钥确定出来. 而我们改进的变型的 Rijndael 算法中的密钥扩展方案采取了依赖数据的移位异或(非线性)方法, 使得从最后一轮扩展密钥计算出密码密钥的计算量大大增加. 以四轮 Rijndael 原算法和变型算法为例: 攻击得到密码密钥, 原算法需要 $2^{20}+5\times 2^6$ 次基本运算; 变型算法至少需要 $2^{20}+5\times 2^{70}$ 次基本运算. 这样就使得变型算法对 Square 攻击有一定的抵抗能力.

3.3 变型算法的统计特性

密码的统计效果是统计攻击和相关密钥攻击的主要依据, 让我们看一下变型算法的统计效果, 如表 1.

表 1 统计效果

名称	随机性测试		明密文独立性测试	明文扩散性测试	密钥扩散性测试
	频数检验	跟随性检验			
原算法	40.482089	37.23553	49.286151	46.853983	44.27460
变型算法	35.18907	22.99748	42.02816	24.73299	29.30333

可以看出, 变型算法的统计效果优于原算法.

3.4 变型算法的密钥装填速度

最后我们看一下变型算法在密钥装填以及加密解密速度上与原算法地比较, 结果见表 2.

表 2 8bits 算法测试

名称	密钥安装速度(M/s)	加密速度(M/s)	解密速度(M/s)
原算法	7.34	9.91	9.30
变型算法	6.06	9.91	9.30

从表中可以看出, 除了密钥装填速度略慢些之外, 其它没有区别. 如果用户所选用的算法要求安全性很高, 并且是用来加密大量数据的, 选择变型算法将更好一些.

参考文献:

[1] J Daemen, V Rijmen. AES Proposal: Rijndael (2nd version) [C]. AES submission.

[2] J Daemen, L Knudsen, V Rijmen. The block cipher square [C]. Fast software encryption 1997, Springer LNCS 1267, 149-165.

[3] Stefan Lucks, Attacking Seven Rounds of Rijndael under 192-bit and 256-bit Keys [Z].

[4] Makoto Sgita, Kazukuni Kobara, Kazuhiro Uehara, Shuji Kubata, Hideki Imai. Relationships mong differential, Truncated differential, impossible differential cryptanalysis against Word-Oriented block ciphers like rijndael, E2[C]: NTT Wireless Systems Innovation Laboratory, Network Innovation Laboratories.

[5] Henri Gilbert and Marine Minier, A collision attacks on 7 rounds of rijndael [C]. http://www.nist.gov/aes.

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

作者简介:

冯国柱 男, 1976 年 11 月生于内蒙古包头市, 2002 年 3 月在中国人民解放军国防科学技术大学获应用数学硕士学位, 现攻读应用数学博士, 感兴趣的领域是信息安全, 网络攻防以及编码密码等.