AES轮变换的代数正规型及其应用*

彭昌勇[1,2],祝跃飞[1],康绯[1],米顺强[3]
(1. 信息工程大学 信息工程学院,河南 郑州 450002;
2. 信息工程大学 理学院,河南 郑州 450000;
3. 95833部队,北京 100081)

摘 要:每个布尔函数的代数正规型(ANF)是唯一的,对于研究布尔函数有重要意义。利用Mathematica软件得到了高级加密标准的轮变换(Sbox, ShiftRow和MixColumn的复合)的128个分量函数的代数正规型。每个分量函数都是32元布尔函数,其项数在448~545,平均为496,远远小于随机32元布尔函数的平均项数2^{31}。这表明AES轮变换与随机置换有巨大偏差。得到这些ANF的时间复杂度在一个2GHz的PC机上只用几分钟。该方法优于通过真值表得到ANF的经典算法——其得到128个分量函数的时间复杂度为$O(128\times 32\times 2^{32}) = O(2^{44})$。作为应用,利用得到的ANF建立了一轮AES的一个方程系统,并用Cryptominisat 2.9.0进行求解。使用Guess-and-Determine的方法,利用一个已知明密对,可以在PC机上2^{33}h内得到全部128比特密钥。

关键词:高级加密标准;代数正规型;分组密码;符号计算;代数攻击; Cryptominisat软件
中图分类号:TN918.1 **文献标志码**:A **文章编号**:1001-2486(2012)02-0014-04

The ANFs of the component functions of AES round transformation and its application

PENG Changyong[1,2], *ZHU Yuefei*[1], *KANG Fei*[1], *MI Shunqiang*[3]
(1. Institute of Information Engineering, Information Engineering University, Zhengzhou 450002, China;
2. Institute of Science, Information Engineering University, Zhengzhou 450000, China; 3. Unit 95833, Beijing 100081, China)

Abstract: The Algebraic Normal Form (ANF) of a Boolean function is unique, which is very important in the research of Boolean function. The 128 ANFs of the component functions of the round transformation (the composition of Sbox, ShiftRow and MixColumn) of the Advanced Encryption Standard (AES) were obtained by using the Mathematica software. Each component function is a 32-variable Boolean function. The number of terms is between 448 and 545, and the average number is 496, which is much smaller than 2^{31}, the number of terms of a random 32-variable Boolean function. This shows a great deviation of the AES round transformation with a random permutation. The time complexity of getting the 128 ANFs of the AES round transformation is only a few minutes on a PC with a 2GHz CPU. The method is better than the classical one in [1] computing the ANF from truth table, with total time complexity of obtaining the 128 ANFs $O(128\times 32\times 2^{32}) = O(2^{44})$. As an application, an equation system for 1-full round of AES was obtained by using the ANFs. The equation system was solved by using Cryptominisat 2.9.0[2]. By the method of Guess-and-Determine, the 128 bits of keys can be recovered in less than 2^{33} hours on a PC with one known plaintext.

Key words: advanced encryption standard; algebraic normal form; block cipher; symbolic computation; algebraic attack; Cryptominisat software

2000年10月,NIST宣布由Joan Daemen和Vincent Rijmen设计的分组密码Rijndael作为最终的高级加密标准AES。众所周知,AES具有丰富的代数结构,因此很多论文[1-7]致力研究AES丰富的代数性质。尽管如此,我们未见到任何公开文献研究将AES轮变换的分量函数显式地表示为F_2上的非线性布尔函数。布尔函数的代数表示通常称为其代数正规型(Algebraic Normal Form:ANF,以下ANF都指代数正规型),每个布尔函数的ANF是唯一的。一个密码算法的分量函数的ANF表示,对于算法的实现和密码分析都是很有用的。如文献[8]中作者基于DES的S盒的布尔表示,得到了DES的最优硬件实现。同时,从一个分组密码的轮变换的代数表示式可以得到输入和输出的显式函数关系,进而可以从数学角度来更有效地分析研究该算法。文献[9]中

* **收稿日期**:2011-07-28
基金项目:国家863高技术计划项目(2007AA01Z471);郑州市科技创新团队项目(10CXTD150)
作者简介:彭昌勇(1974—),男,湖南永州人,博士研究生,E-mail:pengchangyong@tom.com;
祝跃飞(通信作者),男,教授,博士,博士生导师,E-mail:zyf0136@sina.com

还提到了所谓的“代数区分器”,作者期望从 AES 丰富的代数结构出发来找到可以将 AES 与真随机源区分开来的多项式时间的区分器。此外,一个分组密码的轮变换的代数表示,对于代数攻击中方程组的构造也有重要的作用。

求 n 元布尔函数的 ANF 的经典算法是通过其真值表得到,这在文献[1]中有详细描述,其时间复杂度为 $O(n\times 2^n)$。AES 的轮变换(这里指 Sbox, ShiftRow 和 MixColumn 的复合,下同)的每个分量函数都可以视为 128 元的布尔函数,用文献[1]中的算法得到其 ANF 计算上不可行。注意到 AES 的轮变换是 Sbox, ShiftRow 和 MixColumn 的复合,Sbox 变换是 16 个平行的 8×8 的小 S 盒变换,这些小 S 盒的分量函数的 ANF 是可以用文献[1]中的算法得到的,而 ShiftRow 和 MixColumn 都是 F_2^{128} 到其自身的线性变换,很容易将其用 F_2 上的 128×128 的矩阵表示出来。然后利用符号计算软件 Mathematica 将这些变换复合,就得到了 AES 轮变换的分量函数的 ANF,结果显示,这些分量函数与随机的布尔函数有显著的区别。

1　AES 轮变换简介

下面给出 AES 轮变换的简单描述。AES(详细描述见文献[10])是目前使用最广泛的分组密码之一,其分组长度为 128 比特,密钥长度为 128,192 和 256,本文研究的是密钥长度为 128 的 AES。AES 共有 10 轮,其中间状态可以写成下面的矩阵形式

$$\begin{pmatrix} b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \\ b_4 & b_8 & b_{12} & b_{16} \end{pmatrix}$$

其中每个 b_i 是一个字节。下面我们也将 AES 的状态用 128 比特的二进制向量$(x_1,x_2,\cdots,x_{128})$表示,其中 $b_1=(x_1,x_2,\cdots,x_8)$,$b_2=(x_9,x_{10},\cdots,x_{16})$,…,$b_{16}=(x_{121},x_{122},\cdots,x_{128})$。这里 AES 轮变换指在 F_2^{128} 上连续进行下述三个变换:对状态的每个字节使用 S 盒变换(Sbox),对状态矩阵进行行移位变换(ShiftRow),对状态矩阵进行列混合变换(MixColumn)。

S 盒变换是 16 个并行的 8×8 的小 S 盒变换(ByteSub),这是 AES 中唯一的非线性组件。小 S 盒变换(ByteSub)由有限域 F_{256}(通过模 F_2 上的不可约多项式 $x^8+x^4+x^3+x+1$ 得到)上的求逆运算和一个仿射变换复合而成。

行移位变换(ShiftRow)是将状态矩阵的第 i 行循环左移 i 个字节($i=0,1,2,3$)。

列混合变换(MixColumn)对状态矩阵逐列进行变换,它将每一列视为有限域 F_{256} 上的一个多项式,将其乘以 F_{256} 上的多项式 $03x^3+01x^2+01x+02$,再求乘积模 F_{256} 上的多项式 x^4+1 的余式,余式的系数作为变换后的状态矩阵的一列。

2　求 AES 轮变换分量函数的代数正规型

令 F_2 表示 2 元域,$f:F_2^n\to F_2$ 是一个 n 元的布尔函数,则布尔函数 f 的代数正规型(ANF)即 f 为如下形式的多项式表示:

$$f(x_1,\cdots,x_n)=a_0\oplus a_1x_1\oplus\cdots\oplus a_nx_n \oplus a_{n+1}x_1x_2\oplus\cdots\oplus a_{2^n-1}x_1x_2\cdots x_n \quad (1)$$

其中 $a_i\in F_2$,非零 a_i 的个数称为 f 的代数正规型的项数。

求一个 n 元布尔函数的代数正规型的快速方法是文献[1]中的算法,其时间复杂度为 $O(n\times 2^n)$。也可以使用解线性方程组的 Gauss 消元法[11],即将(1)式中的系数 a_i 都视为变元,将自变量$(x_1,\cdots,x_n)$从$(0,\cdots,0)$到$(1,\cdots,1)$的所有取值和其对应的函数值$f(x_1,\ldots,x_n)$代入(1)式就可以得到以 2^n 个系数 $a_0,a_1,\cdots,a_n,a_{n+1},\cdots,a_{2^n-1}$ 为变元的 2^n 个线性方程所构成的一个满秩方程组,求解该线性方程组就可以得到 $a_0,a_1,\cdots,a_n,a_{n+1},\cdots,a_{2^n-1}$ 的值。Gauss 消元法求解 2^n 个变元的 2^n 个线性方程组的时间复杂度为 $O((2^n)^\omega)$,$\omega=2.7\sim 3$。

AES 轮变换可视为 F_2^{128} 到自身的置换,每个分量函数可以视为 128 元的布尔函数,由轮变换的具体特点,轮变换的每个输出比特仅与 32 个输入比特有关,即可以看成 32 元的布尔函数。按照文献[1]中的算法,得到一个分量函数的代数正规型的时间复杂度为 $O(32\times 2^{32})$,故得到 128 个分量函数的代数正规型的时间复杂度 $O(128\times 32\times 2^{32})=O(2^{44})$。

用该方法的时间复杂度显然过高,在 PC 机上需要很长的时间才能得到结果。

本文基于数学软件 Mathematica 的符号计算功能,首先得到 AES 轮变换中 3 个分变换 S 盒变换(Sbox),行移位变换(ShiftRow),列混合变换(MixColumn)的代数正规型;然后利用符号计算得到 3 个变换复合后的代数正规型。

要得到 S 盒变换(Sbox)的代数正规型,只需要得到其 8×8S 盒的 8 个分量函数的代数正规型

即可。文献[12]通过解线性方程组的方法得到了 8 个分量函数的代数正规型。我们通过使用文献[1]中的算法也得到了相同的结论。令 S 盒变换的输入为 $X=(x_1,x_2,\cdots,x_{128})$,输出为 $S=(s_1,s_2,\cdots,s_{128})$。每个 s_i 都是 8 元布尔函数,其项数是 110, 112, 114, 131, 136, 145, 133, 132 中的一个,如:

$$s_1 = x_1 + x_3 + x_1x_3 + x_4 + x_2x_4 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 + x_1x_2x_3x_4 + x_3x_5 + x_1x_3x_5 + x_3x_4x_5 + x_2x_3x_4x_5 + x_1x_2x_3x_4x_5 + x_6 + x_2x_6 + x_1x_2x_6 + x_2x_3x_6 + x_1x_2x_3x_6 + x_4x_6 + x_2x_4x_6 + x_1x_2x_4x_6 + x_3x_5x_6 + x_1x_4x_5x_6 + x_2x_3x_4x_5x_6 + x_1x_7 + x_1x_2x_3x_7 + x_1x_4x_7 + x_1x_2x_4x_7 + x_1x_3x_4x_7 + x_1x_2x_3x_4x_7 + x_2x_5x_7 + x_1x_2x_5x_7 + x_3x_5x_7 + x_1x_2x_3x_5x_7 + x_1x_4x_5x_7 + x_1x_2x_3x_4x_5x_7 + x_6x_7 + x_2x_6x_7 + x_1x_2x_3x_6x_7 + x_2x_4x_6x_7 + x_1x_2x_4x_6x_7 + x_3x_4x_6x_7 + x_2x_3x_4x_6x_7 + x_5x_6x_7 + x_1x_5x_6x_7 + x_3x_5x_6x_7 + x_2x_3x_5x_6x_7 + x_4x_5x_6x_7 + x_1x_8 + x_2x_8 + x_1x_2x_8 + x_2x_3x_8 + x_1x_2x_4x_8 + x_3x_4x_8 + x_1x_5x_8 + x_2x_5x_8 + x_1x_2x_5x_8 + x_3x_5x_8 + x_1x_3x_5x_8 + x_2x_3x_5x_8 + x_1x_2x_3x_5x_8 + x_1x_4x_5x_8 + x_2x_4x_5x_8 + x_1x_2x_4x_5x_8 + x_2x_3x_4x_5x_8 + x_6x_8 + x_1x_6x_8 + x_3x_6x_8 + x_1x_4x_6x_8 + x_2x_4x_6x_8 + x_1x_2x_3x_4x_6x_8 + x_5x_6x_8 + x_1x_5x_6x_8 + x_2x_5x_6x_8 + x_3x_5x_6x_8 + x_1x_2x_3x_5x_6x_8 + x_1x_4x_5x_6x_8 + x_2x_4x_5x_6x_8 + x_1x_2x_4x_5x_6x_8 + x_3x_4x_5x_6x_8 + x_1x_3x_4x_5x_6x_8 + x_2x_3x_4x_5x_6x_8 + x_1x_2x_3x_4x_5x_6x_8 + x_2x_7x_8 + x_3x_7x_8 + x_1x_2x_3x_7x_8 + x_4x_7x_8 + x_1x_4x_7x_8 + x_1x_3x_4x_7x_8 + x_2x_3x_4x_7x_8 + x_2x_5x_7x_8 + x_1x_2x_3x_5x_7x_8 + x_4x_5x_7x_8 + x_1x_4x_5x_7x_8 + x_1x_2x_4x_5x_7x_8 + x_3x_4x_5x_7x_8 + x_1x_3x_4x_5x_7x_8 + x_1x_2x_3x_4x_5x_7x_8 + x_1x_6x_7x_8 + x_1x_2x_6x_7x_8 + x_3x_6x_7x_8 + x_1x_3x_6x_7x_8 + x_4x_6x_7x_8 + x_2x_4x_6x_7x_8 + x_1x_3x_4x_6x_7x_8 + x_5x_6x_7x_8 + x_2x_5x_6x_7x_8 + x_1x_2x_5x_6x_7x_8 + x_4x_5x_6x_7x_8$$

行移位变换(ShiftRow)、列混合变换(MixColumn)都是线性变换,都很容易用 F_2 上的 128×128 的矩阵表示(具体实现可以通过解线性方程组的方法得到)。行移位变换和列混合变换的复合也可以用一个矩阵表示,记为 P(为了节省篇幅,P 的具体形式这里不给出),则 AES 轮变换的输出可以用 $Y=(y_1,y_2,\cdots,y_{128})=P\cdot S^{\mathrm{T}}$ 表示,其中 S^{T} 是 S 盒变换的输出 $S=(s_1,s_2,\cdots,s_{128})$ 的转置。

我们使用 Mathematica 的命令 PolynomialMod[Y, 2]得到 AES 轮变换最终的代数正规型。PolynomialMod[Y, 2]给出一个多项式 mod 2 的约简型。最终我们得到 128 个分量函数的代数正规型的时间复杂度为:在一个 2GHz CPU 的 PC 机上只用几分钟,这远远小于文献[1]中经典算法的时间复杂度 $O(128\times32\times2^{32})=O(2^{44})$。

最终结论如下:

AES 轮变换的 128 个分量函数所含项数依次为 448,452,494,523,518,545,497,492(这 8 个数重复 16 次)。每个输出比特都是 32 元的次数为 7 的布尔函数。

1 ~ 32 比特是 x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_{41}, x_{42}, x_{43}, x_{44}, x_{45}, x_{46}, x_{47}, x_{48}, x_{81}, x_{82}, x_{83}, x_{84}, x_{85}, x_{86}, x_{87}, x_{88}, x_{121}, x_{122}, x_{123}, x_{124}, x_{125}, x_{126}, x_{127}, x_{128} 的函数。

33 ~ 64 比特是 x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30}, x_{31}, x_{32}, x_{33}, x_{34}, x_{35}, x_{36}, x_{37}, x_{38}, x_{39}, x_{40}, x_{73}, x_{74}, x_{75}, x_{76}, x_{77}, x_{78}, x_{79}, x_{80}, x_{113}, x_{114}, x_{115}, x_{116}, x_{117}, x_{118}, x_{119}, x_{120} 的函数。

65 ~ 96 比特是 x_{17}, x_{18}, x_{19}, x_{20}, x_{21}, x_{22}, x_{23}, x_{24}, x_{57}, x_{58}, x_{59}, x_{60}, x_{61}, x_{62}, x_{63}, x_{64}, x_{65}, x_{66}, x_{67}, x_{68}, x_{69}, x_{70}, x_{71}, x_{72}, x_{105}, x_{106}, x_{107}, x_{108}, x_{109}, x_{110}, x_{111}, x_{112} 的函数。

97 ~ 128 比特是 x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{49}, x_{50}, x_{51}, x_{52}, x_{53}, x_{54}, x_{55}, x_{56}, x_{89}, x_{90}, x_{91}, x_{92}, x_{93}, x_{94}, x_{95}, x_{96}, x_{97}, x_{98}, x_{99}, x_{100}, x_{101}, x_{102}, x_{103}, x_{104} 的函数。

为节省篇幅,每个输出比特的完整代数表示式这里不具体给出。我们验证所得到的代数正规型的正确性,以文献[13]第 216 页的 128 比特串 0x193de3bea0f4e22b9ac68d2ae9f84808(16 进制)作为输入,得到了与其相同的输出 0x46681e5e0cb199a48f8d37a2806264c。

3 结论及一轮 AES 的代数攻击

我们得到了 AES 轮变换 128 个分量函数的代数正规型,其项数在 448 和 545 之间,平均为 496,远远小于随机布尔函数的平均项数 2^{31}。这表明 AES 轮变换与随机置换有巨大偏差。分量函数的代数正规型可以用来建立 AES 的方程系统。此外,对于某些变元数较多($\geqslant40$)的布尔函数,其代数正规型可以用来计算 Walsh 谱[14],这也是本文要进行的后续工作。

作为一个应用,我们利用得到的 128 个分量函数的代数正规型建立了一轮 AES(Key addition0, Sbox, ShiftRow, MixColumn, Key addition1)的 F_2 上的非线性方程组,使用猜测和确定(Guess-and-Determine)的方法求解该方程组。众所周知,

Guess-and-Determine 攻击通过先猜测部分密钥的值,再求解剩下的密钥的值。本文中具体方法如下:我们先猜测主密钥($k_1, k_2, \cdots, k_{128}$)的低 32 比特,即($k_{97}, k_{98}, \cdots, k_{128}$)。根据 AES 密钥调度算法的特点,密钥调度中也用到了 S 盒变换,而其代数表达式我们已经求得,于是就可以将第二个轮子密钥的全部 128 比特用主密钥的高 96 比特($k_1, k_2, \cdots, k_{96}$)显式表示出来。这样利用一个已知明密对,我们就可以建立以($k_1, k_2, \cdots, k_{96}$)为变元的 128 个方程的非线性方程组。利用 Cryptominisat 2.9.0[2] 来求解该非线性方程组。我们对($k_{97}, k_{98}, \cdots, k_{128}$)猜对和猜错各进行了 5 次实验,实验环境为:CPU 为 2GHz 的 PC 机,内存 2G,操作系统为 Windows XP。在($k_{97}, k_{98}, \cdots, k_{128}$)猜对的情况下,Cryptominisat 2.9.0 平均用时 5635s 可以得到($k_1, k_2, \cdots, k_{96}$)的正确解。在($k_{97}, k_{98}, \cdots, k_{128}$)猜错的情况下,Cryptominisat 2.9.0 平均用时 5849s 可以确定关于($k_1, k_2, \cdots, k_{96}$)的方程组无解。即 Cryptominisat 2.9.0 给出方程组有解(猜对时)或无解(猜错时)的时间都小于 1h,这样,对于 1 轮 AES,利用一个已知明密对,我们可以在 2^{33}h 内获得 128 比特主密钥。

参考文献(References)

[1] Englund H, Johansson T, Turan M S. A framework for chosen IV statistical analysis of stream ciphers [C]// Proc of INDOCRYPT 2007, 268 – 281.

[2] Cryptominisat[CP/OL]. [2011 – 06 – 15]. http://www.msoos.org/cryptominisat2.

[3] Murphy S, Robshaw M J B. Essential algebraic structures within the AES [C]// Proc of Advances in Cryptology CRYPTO 2002, LNCS 2442, Springer-Verlag, 2002:1 – 16.

[4] Song B, Seberry J. Further observations on the structure of the AES algorithm [C]// Proc of Fast Software Encryption Workshop 2003, LNCS 2887, Springer-Verlag, 2003: 223 – 234.

[5] Le T V, Sparr R, Wernsdorf R, et al. Complementation like and cyclic properties of AES round functions[C]// Proc of Fourth Conference on the Advanced Encryption Standard-AES4, LNCS 3373, Springer-Verlag, 2004:128 – 141.

[6] Murphy S, Robshaw M J B. New observations on Rijndael[EB/OL]. [2000 – 08 – 12]. http://csrc.nist.gov/encryption/aes.

[7] Ferguson N, Shroeppel R, Whiting D. A simple algebraic representation of Rijndael[C]// Proceedings of Selected Areas in Cryptography, Springer-Verlag, 2001:103 – 111.

[8] Kwan M. Reducing the gate count of bitslice DES[EB/OL]. [2011 – 06 – 12]. http://eprint.iacr.org/2000/051.

[9] Cid C, Murphy S, Robshaw M J B. Computational and algebraic aspects of the advanced encryption standard[C]// Proceedings of the Seventh International Workshop on Computer Algebra in Scientific Computing-CASC 2004, St. Petersburg, 2004: 93 – 103.

[10] National Institute of Standards and Technology (U.S.): Advanced Encryption Standard (AES), FIPS Publication 197, November 26, 2001[S/0L]. [2011 – 06 – 11]. http://csrc.nist.gov/publication/_psI/_ps197/_ps – 197.pdf.

[11] Strassen V. Gaussian elimination is not optimal [J]. Numerische Mathematik, 1969, 13:354 – 356.

[12] Gligoroski D, Moe M E G. On deviations of the AES S-box when represented as vector valued boolean Function [J]. International Journal of Computer Science and Network Security, 2007, 7(4): 156 – 162.

[13] Daemen J, Rijmen V. The design of rijndael: AES-the advanced encryption standard (information security and cryptography)[M]. Springer, 2002.

[14] Gupta C, Sarkar P. Computing partial walsh transform from the algebraic normal form of a boolean function [J]. IEEE Transactions on Information Theory, 2009, 55(3): 1354 – 1359.