ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.7, July 2007, pp.1738−1745
DOI: 10.1360/jos181738

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

Dixon 结式在密码学中的应用*

唐棵瑾+, 冯 勇

(中国科学院 成都计算机应用研究所 自动推理实验室,四川 成都 610041)

Applying Dixon Resultants in Cryptography

TANG Xi-Jin+, FENG Yong

(Laboratory for Automated Reasoning and Programming, Chengdu Institute of Computer Applications, The Chinese Academy of Sciences, Chengdu 610041, China)

+ Corresponding author: Phn: +86-10-88256069, Fax: +86-28-85249933 ext 8207, E-mail: tangxij@mails.gucas.ac.cn

Tang XJ, Feng Y. Applying Dixon resultants in cryptography. *Journal of Software*, 2007,18(7):1738−1745.
http://www.jos.org.cn/1000-9825/18/1738.htm

Abstract: Based on extended Dixon resultants, a new algorithm DR (Dixon resultants) is proposed to solve the system of multivariate polynomial equations. The basic idea of DR is to apply the extended Dixon resultants method to the system of multivariate polynomial equations, by taking $x_1,x_2,\ldots,x_{n-1}$ as variables and x_n as parameter. The time complexity of DR technique is evaluated. It seems to be polynomial when the system is sparse and $m=n$, and the mixed volume is polynomial. Moreover, DR technique is compared with Buchberger's algorithm and XL technique in this paper. It is shown that DR is far more efficient than Buchberger's algorithm and XL when $m=n$. DR is a quite efficient algorithm and makes a good use of the sparsity of the sparse system. Besides its efficiency, another advantage of DR is that its complexity is easy to determine.

Key words: multivariate cryptography; polynomial equations over finite field; algebraic attack; Dixon resultants; DR (Dixon resultants)

摘 要: 针对密码学中的多变元多项式二次方程系统求解问题,基于扩展 Dixon 结式提出了一种求解算法 DR(Dixon resultants).基本思想为对于 MQ(multivariate quadratic)问题,把 $x_1,x_2,\ldots,x_{n-1}$ 当作变元,而把 x_n 当作参数,然后利用和改进扩展 Dixon 结式方法求解该类系统.分析了该算法对于一般情况的复杂度,并且基于实验证据猜测:对于某些稀疏问题,新算法的复杂度很有可能也是多项式的.实验结果表明,对于 $m=n$ 的一般和稀疏的问题,DR 效率优于已有的两种算法.除了高效性,新算法还具有复杂度容易度量、计算时间可以预测的优点.

关键词: 多变元密码学;有限域上的多项式方程;代数攻击;Dixon 结式;DR(Dixon resultants)
中图法分类号: TP309 文献标识码: A

代数攻击是现代密码学中的一种攻击方法,其主要手段就是利用密码学系统的良好代数性质及现有代数系统求解方法来攻击密码学系统,目前被认为是最具潜力的攻击方法之一.求解有限域上的多变元二次方程是代数攻击的热点之一,因为该类方程在密码学中有着众多应用,AES(advanced encryption standard,高级加密标准)可以用稀疏的、超定的多变元二次方程来描述[1],而且,近年来提出了很多基于多变元二次方程的密码学系

* Supported by the National Basic Research Program of China under Grant No.2004CB318003 (国家重点基础研究发展计划(973))
Received 2005-10-09; Accepted 2006-04-17

统,如 HFE(hidden field equations)公钥密码系统[2]等.

在国内外求解该类方程一般有两种方法:一种就是 Groebner 基方法[3]以及它的变种,据我们所知,最高效的一个变种为 F4[4];另一种就是 XL(extended linearization)[5]方法以及它的变种,像 XSL(extended sparse linearization),FXL(fixing and XL).但是,这两种方法都有着明显的缺点:首先是效率差.当用上述两种方法求解多变元二次方程问题时,Groebner 基方法的时间复杂度理论上为双指数,实际运行时间为单指数;XL 方法的时间复杂度对于一般的情况是双指数的,对非常超定的问题才是多项式的;另一个缺点是时间复杂度难以度量.由于上述两种方法都要达到一定条件才能终止算法,但是何时能达到条件不能预先判断,造成时间复杂度难以度量.对于稀疏的系统,Groebner 基方法能够受益于系统的稀疏性,但是复杂度难以度量而且效率也不理想;XL 更是不能利用系统的稀疏性,因为它没有乘积因子的选择策略.

在现代计算机代数领域中,求解多项式的非线性方程组一般有 Groebner 基方法、吴方法和结式方法.Dixon 结式[6]作为结式方法中最为有效的一种,受到了众多学者的关注,它被广泛地应用于代数几何和几何定理机器证明中.在使用过程中,Dixon 结式显示了效率高、时间复杂度容易度量及充分利用系统稀疏性的优点.但是,这样一种优秀的算法却没有被引入备受关注的代数攻击领域.基于这种现状,本文把 Dixon 结式引入到代数攻击领域中来,并提出了一种新的算法 DR(Dixon resultants)来求解有限域上多变元多项式二次方程系统.

本文第 1 节介绍 Dixon 结式、KSY(Kapur-Saxena-Yang)方法以及相关的概念.第 2 节介绍新算法 DR.第 3 节给出实验数据.第 4 节对算法进行复杂性分析.第 5 节给出 DR 与已有算法的实验比较.最后是本文结论.

1 背景知识

1.1 Dixon结式和KSY方法

早在 l779 年,Bezout 就创立了一种计算两个单变元多项式结式的方法.Dixon 在 1908 年将其推广到 3 个二变元的多项式系统.该方法通过逐步引入 k 个异于多项式中的变元,推广到 k+1 个 k 变元多项式系统.一般多项式系统的 Dixon 结式的整个计算过程如下:

给定 k+1 个 k 变元多项式组 PS:

$$PS=\begin{cases}p_1(x_1,x_2,\ldots,x_k)=0\\ p_2(x_1,x_2,\ldots,x_k)=0\\ \vdots\\ p_{k+1}(x_1,x_2,\ldots,x_k)=0\end{cases}.$$

设$\alpha_1,\alpha_2,\ldots,\alpha_k$是 k 个新变元,则如下行列式:

$$\Delta(x_1,x_2,\ldots,x_k,\alpha_1,\alpha_2,\ldots,\alpha_k)=\begin{vmatrix}p_1(x_1,x_2,\ldots,x_k) & p_2(x_1,x_2,\ldots,x_k) & \ldots & p_{k+1}(x_1,x_2,\ldots,x_k)\\ p_1(\alpha_1,x_2,\ldots,x_k) & p_2(\alpha_1,x_2,\ldots,x_k) & \ldots & p_{k+1}(\alpha_1,x_2,\ldots,x_k)\\ \ldots & \ldots & \ldots & \ldots\\ p_1(\alpha_1,\alpha_2,\ldots,\alpha_k) & p_2(\alpha_1,\alpha_2,\ldots,\alpha_k) & \ldots & p_{k+1}(\alpha_1,\alpha_2,\ldots,\alpha_k)\end{vmatrix}$$

是关于 $x_1,x_2,\ldots,x_k,\alpha_1,\alpha_2,\ldots,\alpha_k$ 的多项式,并且当 $x_i=\alpha_i(i=1,\ldots,k)$时,

$$\Delta(x_1,x_2,\ldots,x_k,\alpha_1,\alpha_2,\ldots,\alpha_k)=0.$$

这说明:$(x_1-\alpha_1)(x_2-\alpha_2)\ldots(x_k-\alpha_k)$是$\Delta(x_1,x_2,\ldots,x_k,\alpha_1,\alpha_2,\ldots,\alpha_k)$的因子,故

$$\delta(x_1,x_2,\ldots,x_k,\alpha_1,\alpha_2,\ldots,\alpha_k)=\frac{\Delta(x_1,x_2,\ldots,x_k,\alpha_1,\alpha_2,\ldots,\alpha_k)}{(x_1-\alpha_1)(x_2-\alpha_2)\ldots(x_k-\alpha_k)}$$

是一个多项式,并称其为 $p_1,p_2,\ldots,p_{k+1}$ 的 Dixon 多项式.

将 Dixon 多项式看作$\alpha_1,\alpha_2,\ldots,\alpha_k$的多项式,设共有 m 项,并将$\alpha_1,\alpha_2,\ldots,\alpha_k$的各不同幂积的系数记为

$$c_i(x_1,x_2,\ldots,x_k)=0,i=1,\ldots,m.$$

它们都是关于 $x_1,x_2,\ldots,x_k$ 的多项式,我们称这些 c_i 为方程组 $p_1=0,p_2=0,\ldots,p_{k+1}=0$ 的 Dixon 导出方程组.易知,原方程组的解必定是 Dixon 导出方程组的解.

在导出方程组中,把其中出现的关于变元$(x_1,x_2,\ldots,x_k)$的所有幂积按字典序由大到小写为 $e_1,e_2,\ldots,e_n$.这样,Dixon 导出方程组就可以写成这样的标准形式:

$$\delta \equiv D\begin{pmatrix} e_n \\ \vdots \\ e_2 \\ e_1 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix},$$

其中,D 为 Dixon 导出方程组关于幂积 $e_1,e_2,\ldots,e_n$ 的系数矩阵,称为多项式 $p_1,p_2,\ldots,p_{k+1}$ 的 Dixon 矩阵.当 Dixon 矩阵为方阵时,该行列式的值成为 Dixon 结式.Dixon 结式为 0,是多项式组 PS 有公共零点的必要条件.

但是,Dixon 矩阵很多时候是奇异的,它的行列式的值总等于 0,就不能提供任何关于公共零点的信息;有时,Dixon 矩阵甚至不是方阵,连 Dixon 结式都不能计算.为了解决这个问题,Deepak Kapur,Tushar Saxena 和杨路提出了 KSY 方法,也称为扩展 Dixon 结式.

给定任意的含有 $k+1$ 多项式方程的含参数多项式方程组 F,按上述方法计算它的 Dixon 矩阵 D,设它的大小为 $s_1\times s_2$,它的秩为 r.然后,设有一些对 $x_1,x_2,\ldots,x_k$ 的约束 C,有如下形式 $x_1\neq 0\wedge x_2\neq 0\wedge\ldots\wedge x_k\neq 0$.设 m_i 为 Dixon 矩阵的 i 列,它所对应的单项式为 $monom(m_i)$.给定约束 C,设 $nvcol(C)$代表那些在约束下不为 0 的列,即

$$C\Rightarrow monom(m_i)\neq 0.$$

设 N_i 为所有通过删除 D 中属于 $nvcol(C)$中一列而得到 $s_1\times(s_2-1)$的子矩阵的集合.设$\phi(\alpha_1,\alpha_2,\ldots,\alpha_k)\to Q$ 为一个映射,即把参数的实际数值代入,获得有理数域的代数闭域 Q 上的一个值.$\phi(F)$,$\phi(D)$和$\phi(N_1)$分别表示把参数的值代入到 F,D 和 N_1 而得到的结果.最后,让 $R=\{Y|Y$ 是 D 的一个 $r\times r$ 非奇异子矩阵$\}$.

定理 **1**. 如果$\exists X\in N_1(rank(X)<rank(D))$,那么对所有的 $Y\in R$,$\phi(\det(Y))$等于 0,如果$\phi(F)$有一个公共仿射零点满足约束 C.

我们把定理 1 的前提条件$\exists X\in N_1(rank(X)<rank(D))$称为 RSC(rank submatrix construction)前提.如果前提成立,那么 R 的任何成员就称为 KSY Dixon 矩阵;KSY Dixon 矩阵行列式的值称为扩展 Dixon 结式.但是,直接检查前提条件效率是不能令人满意的,在文献[2]中提供了另一种算法来检验前提条件.

算法 **1**. 计算扩展 Dixon 结式.

1. 计算 F 的 Dixon 矩阵 D.
2. 求解方程 $D\overline{w}=\overline{0}$, $\overline{w}=(w_1,\ldots w_{s_2})$.
3. 在 $\overline{w}$ 中是否存在这样的 w_i,它满足 $w_i=0$ 而且 $C\Rightarrow monom(m_i)\neq 0$.如果存在这样的 w_i,那么对 D 进行高斯消元,对角线上非零元素的乘积即为扩展 Dixon 结式.
4. 否则,返回失败.

1.2 混合体积(mixed volume)

一个多项式 f 的支撑集凸包被称为它的牛顿多胞体,用 $N(f)$来表示.

定义 **1**[7,8]. 混合体积函数$\mu(Q_1,Q_2,\ldots,Q_d)$(其中,Q_i 为凸包)是一个根据 Minkowski 积及扩展操作的独一无二的多线性函数,具有多线性性质:

$$\mu(Q_1,\ldots,aQ_k'+bQ_k'',\ldots,Q_d)=a\mu(Q_1,\ldots,Q_k',\ldots,Q_d)+b\mu(Q_1,\ldots,Q_k'',\ldots,Q_d).$$

为了确保唯一性,$\mu(Q_1,\ldots,Q)=d!Vol(Q)$,其中,$Vol(Q)$为凸包的欧几里德空间体积.

1.3 MQ问题

有如下的一个有限域上的多变元二次方程系统,它含有 m 个方程,n 个变元,所有的元素取值范围在 $GF(q)$上,所有的方程都是二次的.

$$A=\begin{cases} l_1(x_1,x_2,\ldots,x_n)=0 \\ \quad\vdots \\ l_m(x_1,x_2,\ldots,x_n)=0 \end{cases}.$$

给定一个如上的系统,我们把找到该系统的一个解,并不需要找到全部解,称为 MQ 问题.

2 DR 算法

DR 算法的主要思想是通过把 $x_1,x_2,\ldots,x_{n-1}$ 当作变元,而把 x_n 当作参数,然后应用扩展 Dixon 结式方法来求解 MQ 问题.因为 Dixon 结式方法要求方程数等于变元数加 1,所以,DR 会对 $m=n$ 的 MQ 问题效率较高,同时,DR 适用于超定问题.

设 A 是一个 $m=n$ 的 MQ 问题,所有的元素都在域 $GF(q)$上.设 $a_1,a_2,\ldots,a_k$ 是 A 的一个公共零点,$V(A)$是 A 的所有公共零点.然后定义一个投影算子$\pi_k((a_1,a_2,\ldots,a_k))=a_k$.

算法 **2**. DR 主算法.

输入:一个 $m=n$ 的 MQ 问题 A.

输出:至少一个 A 的公共零点.

1. 把 $x_1,x_2,\ldots,x_{n-1}$ 看作变元,x_n 看作参数,计算 A 的 Dixon 结式,
2. 运行子算法 RSC 来检查 RSC 前提,然后选择需要行和列来构造 KSY Dixon 矩阵,
3. 构造 KSY Dixon 矩阵,
4. 计算 KSY Dixon 矩阵行列式的值,
5. 在 $GF(q)$上,用 Berlekamp 算法求解第 4 步所得的方程,可能得到数个解,设解的集合为 s,
6. 把第 5 步得到的每一个解代入到 KSY Dixon 矩阵,然后求解线性方程组来得到其他变元的值,
7. 如果第 6 步没有得到一组 A 的公共零点,则令 $s=(0$,子算法 RSC 中 p 的取值),然后执行第 6 步.

算法 **3**. 构造 KSY Dixon 矩阵的子算法 RSC.

输入:一个 $s_1\times s_2$ 的 Dixon 矩阵.

输出:构造 KSY Dixon 矩阵需要的行和列.

1. 把矩阵中的 x_n 用 $GF(q)$上的一个随机数 p 来代替,
2. 对上一步得到的矩阵进行高斯消元,设得到的矩阵为 M',且该矩阵的秩为 r,
3. 如果 M'是一个满秩方阵,则返回矩阵所有的行和列,
4. 对 M'的每一列作如下运算:
   (a) 通过删除该列来构造 M'的一个大小为 $s_1\times(s_2-1)$的子矩阵 M_s,
   (b) 如果 M_s 的秩小于 r,则终止循环,
5. 如果第 4 步找到一个子矩阵 M_s 的秩小于 r,则
   (a) 选择那些可以用来构造 M'的一个秩为 r 的子矩阵的列,
   (b) 对 M'转置,然后进行高斯消元,再选择那些可以用来构造 M'的一个秩为 r 的子矩阵的行,
   (c) 返回选择的行和列,
   否则转第 1 步.

2.1 几点说明

(1) RSC 前提一般情形下都能满足,在文献[6]中也提到了这一点,在我们大量的测试例子中,没有遇到过不成立的情形;

(2) 我们改进传统的 KSY Dixon 矩阵构造方法,新算法充分利用了数字计算的高效性;

(3) 在算法 RSC 中的第 4 步,我们是在高斯消元后的矩阵上来构造子矩阵的,所以,子矩阵高斯消元的计算量远远小于 $O(s_1\times(s_2-1)\times(s_2-1))$;

(4) 如果只想得到一个公共零点,那么,当算法 RSC 中的第 4 步失败时,我们可以立即回到主算法 DR 中,跳过第 3~5 步,让 $s=\{p\}$,然后执行第 6 步.这是因为 RSC 前提总是成立的,所以,失败时说明 $p\in\pi_n(V(A))$;

(5) 实践中我们发现,用以下的步骤代替主算法 DR 中的第 4~6 步能节约计算时间.

把 $GF(q)$中的每一个值代入到 KSY Dixon 矩阵中求行列式的值,如果为 0,则求解其他变元.

2.2 DR算法的正确性

在我们的实验中,RSC 前提总是成立,虽然存在不成立的情况,不过情形极为稀少.在以下的讨论中,我们假设 RSC 前提成立. $x_1^0, x_2^0, \ldots$ 对应于 Dixon 矩阵中的列 1,该列很多情况不是其余列的线性组合.这样,对于公共零点就没有约束.如果列 1 是其余列的线性组合,且还有很多列满足 RSC 前提,这样,我们就可以最小化约束.通常情况下,我们就可以达到对 x_n 没有约束.如果连这一点都做不到,我们就把 0 添加到 s 中来解决这个问题.

在子算法 RSC 的第 4 步,如果 $p \in \pi_n(V(A))$,检查 RSC 前提的过程可能失败,也可能成功:如果我们检查失败了,那么,我们赋上另一个随机值继续程序,可以得到$\pi_n(V(A))$的全集;如果没有失败,那么我们可能漏掉了 p,可以把 p 加入到 s 中来解决这个问题.

综上所述,在绝大多数情况下,DR 算法可以得到所有的公共零点.

3 实验结果

根据 MQ 的稀疏程度,我们把问题分为一般和稀疏两种,然后根据混合体积不同的增长方式,我们又把稀疏问题分为两类.

3.1 一般的MQ问题

这部分实验中,DR 求解的方程是由 Maple 中的 *randpoly* 命令随机产生的,我们确保它至少有一个公共零点.所有元素都在 $GF(127)$上,见表 1.其中,n:变元数;m:方程数;Terms:项数;Mixed volume:混合体积; M:Dixon 矩阵大小;M':KSY Dixon 矩阵大小.

Table 1 DR over $GF(127)$ for general problem

表 1 DR 算法对一般 MQ 问题的实验结果

	General problem					
n	3	4	5	6	7	8
m	3	4	5	6	7	8
Terms	9	14	16	22	28	30
Mixed volume	8	14	26	54	124	254
M	(5,5)	(14,14)	(41,38)	(113,109)	(236,222)	(406,430)
M'	(5,5)	(12,12)	(29,29)	(71,71)	(150,150)	(278,278)

3.2 稀疏的MQ问题

根据混合体积的增长方式,我们把稀疏问题分为两类:一类为多项式方式增长的,称为 A 型;另一类是指数方式增长的,称为 B 型.

3.2.1 稀疏 A 型

这部分实验中,我们给出一系列 A 型的例子,每一个方程具有如下的形式:$l_i=x_i+x_{(i \bmod n)+1}\times x_{((i+1) \bmod n)+1}+b_i$.当 n=3 时,求解的系统则可能为$\{x_1+x_2\times x_3+4, x_2+x_3\times x_1+114, x_3+x_1\times x_2+106\}$,见表 2.其中,表中各项说明与表 1 相同.

Table 2 DR over $GF(127)$ fro sparse MQ problem of type A

表 2 DR 算法对于稀疏 A 型的实验结果

	Sparse problem of type A									
n	3	4	5	6	7	8	9	10	11	12
m	3	4	5	6	7	8	9	10	11	12
Terms	7	9	11	13	15	17	19	21	23	25
Mixed volume	5	6	12	17	30	46	77	122	200	321
M	(2,2)	(4,4)	(7,7)	(12,12)	(20,20)	(33,33)	(54,54)	(88,88)	(143,143)	(232,232)
M'	(2,2)	(4,4)	(7,7)	(12,12)	(20,20)	(33,33)	(54,54)	(88,88)	(143,143)	(232,232)

3.2.2 稀疏 B 型

这部分实验中,我们给出一系列 B 型的例子,每一个方程具有如下的形式: $l_i = x_i + x_{(i \bmod n)+1} + x^2_{((i+1) \bmod n)+1} + b_i$.当 n=3 时,求解的系统则可能为$\{x_1+x_2+x_3^2+25, x_2+x_3+x_1^2+119, x_3+x_1+x_2^2+116\}$,见表 3.其中,表中各项说明与表 1

相同.

Table 3 DR over *GF*(127) for sparse MQ problem of type B

表 **3** DR 算法对稀疏 B 型的实验结果

	Sparse problem of type B								
n	3	4	5	6	7	8	9	10	11
m	3	4	5	6	7	8	9	10	11
Terms	7	9	11	13	15	17	19	21	23
Mixed volume	8	16	32	64	128	256	512	1024	2048
M	(4,4)	(8,8)	(16,16)	(32,32)	(64,64)	(128,128)	(256,256)	(512,512)	(1024,1024)
M′	(4,4)	(8,8)	(16,16)	(32,32)	(64,64)	(128,128)	(256,256)	(512,512)	(1024,1024)

4 DR 算法的复杂度分析

DR 算法的运算量取决于 Dixon 矩阵的大小.在子算法 RSC 中,把 Dixon 矩阵中的 x_n 用一个随机数 p 代替.如果 $p\notin\pi_n(V(A))$,那么,我们需要对 Dixon 矩阵进行两次高斯消元,对 KSY Dixon 矩阵进行若干次高斯消元,次数取决于我们上一步求得的解的个数;如果 $p\in\pi_n(V(A))$,那么,我们通常需要进行 3 次高斯消元,因为连续两次选择的 p 都属于 $\pi_n(V(A))$的概率很小,小于 $\frac{5}{127}\times\frac{4}{127}$ (对于 MQ 问题,通常只有少于 5 个的公共零点).我们知道,对于一个大小为 $n\times n$ 的矩阵,一般高斯消元的复杂度为 n^3,改进算法的复杂度可以降为 $n^{2.3766}$.所以,DR 算法的复杂度至少为 $\min(s_1,s_2)^{\omega}$,其中:在一般的高斯消元算法中,ω=3;在改进算法中,ω=2.3766.

当 MQ 问题是满的,即所有可能出现的项出现在每一个方程中时,根据 Dixon 矩阵的构造过程,我们可以发现:Dixon 矩阵的大小恰好等于多项式 $\prod_{i=1}^{n-1}\left(\sum_{j=1}^{i}(1+x_j)\right)$ 中项数,为 $\frac{(2\times n)!}{n!\times(n+1)!}$.所以,对于满的 MQ 问题,Dixon 矩阵的大小为 $\frac{(2\times n)!}{n!\times(n+1)!}$.设 $f(n)=\frac{(2\times n)!}{n!\times(n+1)!}$,那么 $\lim_{x\to\infty}\left(\frac{f(x+1)}{f(x)}\right)=4$.所以,Dixon 矩阵的大小总是小于 4^n.但是,求解的 MQ 问题一般都不是满的,因为如果是满的,我们可以进行一个高斯约旦消元,这样就可以在每个方程中消去 $n-1$ 项.

对于一般的 MQ 问题,Dixon 矩阵的大小等于或小于 $\frac{(2\times n)!}{n!\times(n+1)!}$,把这个式子写成较为直观的形式,可以得到 Dixon 矩阵的大小为 C^n,$C\to4$,C 随着 n 的增大而增大.

$$C\approx\begin{cases}2, & n=1\ldots8\\ 3, & n=9\ldots22\\ 3.5, & n=22\ldots88\\ 4, & n>80\end{cases}.$$

当 MQ 问题是稀疏时,Dixon 矩阵的大小取决于系统的结构.大量实验表明:当混合体积是以多项式方式增长时,Dixon 矩阵的大小极有可能也是以多项式方式增长;如果混合体积是以指数方式增长时,Dixon 矩阵的大小也是以指数增长,并且大小为 2^n.

总结上面的讨论,我们可以得到 DR 算法的复杂度为:

- 多项式的,当 $m=n$,系统是稀疏 A 型的.
- $2^{\omega\times n}$,当 $m=n$,而且系统是稀疏 B 型的.
- $C^{\omega\times n}$,当 $m=n$,一般的 MQ 问题,$C\to4$.

5 DR 算法与已有算法的比较

5.1 DR与Groebner基算法的比较

因为 Groebner 基算法的复杂度难以度量,在这部分的比较中,我们比较求解同一个问题 DR 和 Groebner 基算法所需的时间.我们采用了两个版本的 Groebner 基算法的实现:一个是 Maple 9.5 的实现,命令为

GroebnerBasis,所采用的算法为经典的 Buchberger 算法;另一个是 Singular 3.0[9]的实现,命令为 slimgb,所采用算法是 F4 的一个变种.我们的算法是在 Maple 9.5 中实现的.所有的计算都是在一台配置为 Intel P4 1.4G,256 兆内存的机器上进行的,见表 4~表 6.其中,*n*:变元数;*m*:方程数;Terms:项数;DR:DR 算法所用秒数;G1:Maple 9.5 所用秒数;G2:Singular 所用秒数.

Table 4 Comparisons between DR and Buchberger's algorithm (1)

表 **4** DR 与 Groebner 基算法的比较(1)

	General problem					
n	3	4	5	6	7	8
DR	1.32	1.61	6.5	43.39	118.6	557.75
G1	0.29	269.52	>3600	?	?	?
G2	<0.01	0.13	971.02	?	?	?

Table 5 Comparisons between DR and Buchberger's algorithm (2)

表 **5** DR 与 Groebner 基算法的比较(2)

	Sparse problem of type A									
n	3	4	5	6	7	8	9	10	11	12
DR	0.20	0.28	0.30	0.35	0.86	1.80	4.22	9.25	21.51	51.05
G1	0.03	0.03	0.25	1.36	1.39	114.42	>7200	?	?	?
G2	<0.01	<0.01	<0.01	<0.01	0.02	0.07	3.80	1489.23	>18000	?

Table 6 Comparisons between DR and Buchberger's algorithm (3)

表 **6** DR 与 Groebner 基算法的比较(3)

	Sparse problem of type B								
n	3	4	5	6	7	8	9	10	11
DR	0.19	0.31	0.45	1.42	4.65	15.86	94.45	406.20	1593.26
G1	0.03	0.03	0.25	1.36	1.39	114.42	>7200	?	?
G2	<0.01	<0.01	0.01	0.05	0.75	320.56	4421.42	?	?

5.2 DR与XL的比较

DR 和 XL 的复杂度都取决于各自生成的矩阵的大小,我们这里用了 XL 的一个变种 FXL(固定 x_n).当 FXL 生成矩阵的秩等于列数时,我们终止算法,因为这时可以得到一个关于 x_n 的方程,然后可以求解该方程,进而求解其他变元,见表 7~表 9.其中,*n*:变元数;*m*:方程数;Terms:项数;DR:DR 生成矩阵的大小;FXL:FXL 生成矩阵的大小.

Table 7 Comparisons between DR and XL (1)

表 **7** DR 和 XL 的比较(1)

	General problem					
n	3	4	5	6	7	8
DR	(5,5)	(13,14)	(32,32)	(65,72)	(224,179)	(406,330)
FXL	(18,15)	(80,56)	(127,125)	(1512,792)	?	?

Table 8 Comparisons between DR and XL (2)

表 **8** DR 和 XL 的比较(2)

	Sparse problem of type A									
n	3	4	5	6	7	8	9	10	11	12
DR	(2,2)	(4,4)	(7,7)	(12,12)	(20,20)	(33,33)	(54,54)	(88,88)	(143,143)	(232,232)
FXL	(9,9)	(16,15)	(75,57)	(126,100)	(588,392)	(960,661)	(?)	(?)	(?)	(?)

Table 9 Comparisons between DR and XL (3)

表 **9** DR 和 XL 的比较(3)

	Sparse problem of type B								
n	3	4	5	6	7	8	9	10	11
DR	(4,4)	(8,8)	(16,16)	(32,32)	(64,64)	(128,128)	(256,256)	(512,512)	(1024,1024)
FXL	(18,15)	(80,56)	(350,210)	(1512,792)	(?)	(?)	(?)	(?)	(?)

6 对 Sflash 新的攻击

Sflash 是一种快速的多变元密码学签名算法,特别适用于智能卡等计算能力比较差的应用.这个方案推出后,又更新了两遍,现在最高版本是 v3,在 Sflash-v2 中签名长度为 259 bits,而公钥大小为 15 Kbytes.具体的签名方法可以参考文献[10].

在 Sflash 中,n=37,m=26,所有的方程都是二次的,所有的元素都在 $GF(2^7)$上.把 11 个任意变元赋值为 0,一般情况下,我们也还能得到数量大于 1 的公共解,那么,我们就只剩 26 个未知数.我们选择 27 个方程,然后用 DR 算法求解.把 n=26 代入到公式 $\frac{(2\times n)!}{n!\times(n+1)!}$ 中,得到矩阵大小为 18367353072152^2,那么,我们可以得到 DR 求解 Sflash 需要的计算量最多为 $18367353072152^{2.3722}\approx 2^{104}$.

7 结 论

本文基于扩展 Dixon 结式,提出了一种新的算法用来求解 MQ 问题.DR 算法是相当高效的一种算法,尤其是对于稀疏的 MQ 问题,因为它能充分利用 MQ 问题的稀疏性.对于那些稀疏而且混合体积增长是以多项式方式增长的问题,DR 算法的复杂度可能是多项式的.据我们所知:DR 是在比较大的域上对于某些稀疏的 MQ 问题效率可以超过穷举搜索的算法.但是综合起来看,对于很多实际系统,最好的求解方法还是穷举.

DR 算法除了高效性以外,它的另外一个优点是其复杂度容易度量.与 Groebner 和 XL 算法不同的是,在计算之前,我们可以预计它大概所需花费的时间.

References:

[1] Murphy S, Robshaw MJB. Essential algebraic structure within the AES. In: Moti Y, ed. Proc. of the 22nd Annual Int'l Cryptology Conf. on Advances in Cryptology. London: Springer-Verlag, 2002. 1–16.

[2] Patarin J. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. LNCS 1070, Springer-Verlag, 1996. 33–48. http://www.minrank.org/#courtois/hfe.ps

[3] Wadams W, Loustaunau P. An Introduction to Grobner Bases. New York: American Mathematical Society, 1994.

[4] Faugére J-C. A new efficient algorithm for computing Gröbner bases F_4. Journal of Pure and Applied Algebra, 1999,139(1-3): 61–88.

[5] Courtois N, Klimov A, Patarin J, Shamir A. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Prened B, ed. Proc.of the EUROCRYPT 2000. LNCS 1807, Berlin, Heidelberg: Springer-Verlag, 2000. 392–407.

[6] Kapar D, Saxena T, Yang L. Algebraic and geometric reasoning using dixon resultants. In: Proc. of the ISSAC. 1994. 99–107.

[7] Gelfand IM, Kapranov MM, Zelevinsky AV. Discriminants, Resultants, and Multidimensional Determinants. Boston: Birkhauser, 1994.

[8] Cox D, Little J, O'Shea D. Using Algebraic Geometry. New York: Springer-Verlag, 1998.

[9] Greuel GM, Pfister G, Schonemann H. Singular 3.0. Centre for Computer Algebra A Computer Algebra System for Polynomial Computations: University of Kaiserslautern, 2001. http://www.singular.uni-kl.de/

[10] Courtois N, Goubin L, Patarin J. Second updated version of sflash specification (sflash-v2). 2001. http://www.minrank.org/sflash

唐樨瑾(1981－),男,浙江富阳人,工程师,主要研究领域为信息安全,密码学,符号计算.

冯勇(1965－),男,博士,研究员,博士生导师,主要研究领域为计算机代数.