An Efficient Algorithm for Computing Branch Gröbner Systems and Its Applications in Algebraic Cryptanalysis

Yao Sun, Zhenyu Huang, Dongdai Lin
SKLOIS, Institute of Information Engineering, CAS, (China)

Dingkang Wang
KLMM, Academy of Mathematics and Systems Science, CAS, (China)

dwang@mmrc.iss.ac.cn

Abstract

Solving systems of boolean polynomial equations is a kernel problem in algebraic computations and Gröbner basis is one of the most important tools to solve such systems.

In 2009, Sun and Wang proposed an algorithm for computing a branch Gröbner system [8, 9] based on the matrix version of the F5 algorithm [4]. For a set of boolean polynomials, their algorithm uses the F5 algorithm to compute a Gröbner basis, and creates branches before constructing huge matrices, such that the computing complexity for each branch can be controlled in a relative low level. Their algorithm uses zero-suppressed binary decision diagrams (ZDD) to store Boolean polynomials and has a good performance for a class of stream cypher generated by linear feedback shift registers. The ZDD data structure is also used in PolyBoRi [1] and Chai et al.'s characteristic set algorithm [3, 5].

In this talk, a new algorithm for computing branch Gröbner systems is presented. Some new techniques for manipulating boolean polynomials are used to build Gröbner bases for all branches. ZDD is again used as the basic data structure to store boolean polynomials. The implementation of this new algorithm in C performs very well for many examples. The ideas used in this new algorithm can also be extended to compute branch Gröbner systems in a more general form, which will be studied in our future work.

Let $B := \mathbb{F}_2[x_1, \ldots, x_n]/\langle x_1^2 - x_1, \ldots, x_n^2 - x_n\rangle$ be a boolean polynomial ring over the binary field $\mathbb{F}_2 = \{0, 1\}$ with n variables $\{x_1, \ldots, x_n\}$. Let F be a set of boolean polynomials in B, an ideal generated by F over B is defined as $\langle F\rangle = \{p_1 f_1 + \cdots + p_m f_m \mid p_1, \ldots, p_m \in B, f_1, \ldots, f_m \in F\}$.

Let $\prec$ be an order on B deduced from a monomial order in $\mathbb{F}_2[x_1, \ldots, x_n]$, and F be a set of boolean polynomials in B. A set $G \subset \langle F\rangle$ is called a **Gröbner basis** of $\langle F\rangle$, if for any $f \in \langle F\rangle$, there exists $g \in G$ such that $\mathrm{lm}(g)$ *divides* $\mathrm{lm}(f)$.

In this talk, we will consider a variant of Gröbner bases.

Definition 1 (Branch Gröbner system) *Let $\prec$ be an order on B deduced from a monomial order in $\mathbb{F}_2[x_1, \ldots, x_n]$, and F be a set of boolean polynomials in B. A finite set $\mathcal{G} = \{G_1, \cdots, G_l\}$ is called a* **branch Gröbner system** *of the ideal $\langle F\rangle$, if*

1. *G_i is a Gröbner basis for the ideal $\langle G_i\rangle \subset B$, and*
2. *$V(F) = V(G_1) \cup \cdots \cup V(G_l)$,*

where $V(F) = \{\alpha \in \mathcal{F}_2^n \mid f(\alpha) = 0, \forall f \in F)\}$ and similarly for $V(G_i)$. Particularly, each G_i is called a branch of this branch Gröbner system $\mathcal{G}$.

Please note that a general Gröbner basis of $\langle F\rangle$ directly constructs a branch Gröbner system. A branch Gröbner system will be easier to be computed than a general Gröbner basis, because in each branch the corresponding system is simpler. In current talk, instead of computing a branch Gröbner system in its general form, we present an efficient algorithm for computing a special branch Gröbner system defined below.

Definition 2 (Linear branch Gröbner system) *A branch Gröbner system $\mathcal{G} = \{G_1, \cdots, G_l\}$ is called a* **linear branch Gröbner system** *of the ideal $\langle F\rangle$, if for any $g \in G_i$, we have $\mathrm{lm}(g) \in \{x_1, \ldots, x_n\}$ where $i = 1, \ldots, l$, i.e. each polynomial appearing in this branch Gröbner system has a linear leading monomial.*

Linear branch Gröbner systems are similar to the characteristic sets discussed in [3, 5]. But the algorithm presented in this paper can be extended to compute other branch Gröbner systems with a small adaption.

Clearly, a linear branch Gröbner system is sufficient to find all points in $V(F)$ directly.

The new algorithm has been implemented in C based on the CUDD package [7]. Our implementation is tested by the famous Bivium stream cipher after guessing several bits. Examples are from [6], and the input of examples all include 176 variables and 160 polynomials. The timing below is obtained from a PC (Core i7-2600, 4GB memory) running Windows 7 (64 bit).

Table 1: Timings (sec.)

Bits guessed	Average Time	Max Time	Min Time
37	0.186	0.359	0.078
36	0.401	0.609	0.265
35	0.655	0.874	0.453
34	3.584	9.391	1.342

In the above table, the first column shows how many bits/variables are guessed in the Bivium system. Average Time is obtained from 10 times of *arbitrary* guesses. Max Time and Min Time give the largest and smallest time during these tests. The data in this table shows this new algorithm is efficient and guessing 35 variables leads to the best attack of Bivium system which is consistent with existing results.

Keywords

Gröbner basis, branch Gröbner system, boolean polynomial, algorithm, algebraic cryptanalysis.

References

[1] M. Brickenstein and A. Dreyer. PolyBoRi: A framework for Gröbner-basis computations with Boolean polynomials. J. Symb. Comp., vol. 44(9), 1326-1345, 2009.

[2] Raddum, H.: Cryptanalytic results on TRIVIUM. eSTREAM, ECRYPT Stream Cipher Project, Report 2006/039 (2006)

[3] F.J. Chai, X.S. Gao, and C.M. Yuan. A characteristic set method for solving Boolean equations and applications in cryptanalysis of stream ciphers. J. Syst. Sci. Complex., vol. 21(2), 191-208, 2008.

[4] J.-C. Faugère. A new effcient algorithm for computing Gröbner bases without reduction to zero (F_5). In Proc. ISSAC'02, ACM Press, 75-82, 2002. Revised version downloaded from fgbrs.lip6.fr/jcf/Publications/index.html.

[5] X.S. Gao and Z.Y. Huang. Characteristic set algorithms for equation solving in finite fields. J. Symb. Comp., vol. 47(6), 655-679, 2012.

[6] Z.Y. Huang and D.D. Lin. Attacking Bivium and Trivium with the Characteristic Set method. AFRICACRYPT 2011, 77-91, 2011.

[7] F. Somenzi. CUDD: CU Decision Diagram package release 2.3.0. University of Colorado at Boulder, 1998.

[8] Y. Sun and D.K. Wang. Branch Gröbner bases algorithm over Boolean ring (in Chinese). J. Syst. Sci. & Math. Sci., vol. 9, 1266-1277, 2009.

[9] Y. Sun and D.K. Wang. The implementation and complexity analysis of the branch Gröbner bases algorithm over Boolean ring. In Proc. ASCM 2009, 191-200, 2009.