Alexander Rostovtsev[1] and Anna Shustrova[2]
St. Petersburg state polytechnic university, Russia, St. Petersburg

[1] alexander.rostovtsev@ibks.ftk.spbstu.ru, [2] luminescenta@gmail.com

SIMPLIFICATION/COMPLICATION OF THE BASIS OF PRIME BOOLEAN IDEAL

Prime Boolean ideal has the basis of the form ($x_1 + e_1$, ..., $x_n + e_n$) that consists of linear binomials. Its variety consists of the point (e_1, ..., e_n). Complication of the basis is changing the simple linear binomials by non-linear polynomials in such a way, that the variety of ideal stays fixed. Simplification of the basis is obtaining the basis that consists of linear binomials from the complicated one that keeps its variety.

Since any ideal is a module over the ring of Boolean polynomials, the change of the basis is uniquely determined by invertible matrix over the ring.

Algorithms for invertible simplifying and complicating the basis of Boolean ideal that fixes the size of basis are proposed. Algorithm of simplification optimizes the choose of pairs of polynomials during the Groebner basis computation, and eliminates variables without using resultants.

Key words: block ciphers, Boolean functions, cryptanalysis, characteristic set, Groebner basis, hash functions, varieties.

Notations

Vectors are denoted by bold font (**x**, **f**, **g**) (the case of the row and the column vector is clear from context);

G$_n$[**x**], **x** = (x_1, …, x_n) is the ring of Boolean polynomials;

A, B, C, ... are ideals of polynomial ring;

P is a prime ideal;

A Å B is the sum of ideals;

V(A) is the variety (the set of common zeroes) of ideal;

$\overline{x} = 1 + x$ is the inverse of *x*;

DNF is the disjunctive normal form;

LM(*f*) is the leading monomial of polynomial *f*;

f is the order of monomials.

1. Introduction

Finite Boolean rings are widely used in cryptography. Many cryptographic problems such as hash-function inverting and computing the key of block or stream cipher under known plaintext/ciphertext take solving systems of Boolean equations. The set of variables contains the key bits and the bits of intermediate texts. Since a hash function and a ciphertext are easy to compute, the equations are sparse (each polynomial depends on relatively small number of variables comparatively to total number of variables).

The ring $\mathsf{G}_n[\mathbf{x}]$, $\mathbf{x} = (x_1, \ldots, x_n)$, of n-bit Boolean functions was introduced by I. Zhegalkin [18]. Solving the system of Boolean equations means the computing the common zeroes of sparse polynomials of $\mathsf{G}_n[\mathbf{x}]$. These polynomials form the ideal in $\mathsf{G}_n[\mathbf{x}]$ and their common zeroes form the variety of ideal. The problem of computing the variety of ideal of sparse polynomials is known as satisfiability problem which is NP-complete [1].

If there are one or few known plaintext/ciphertext pairs, the encryption key is determined uniquely with high probability. If the key (input of hash function) is determined uniquely, then the ideal is prime.

There are few different methods for computing the variety of Boolean ideal. The variety can be obtained by eliminating the variables. The common method of elimination takes computing of the resultant in polynomial ring [16]. Unfortunately, this method takes extremely large memory and since it is impractical.

N. Courtois suggested the extended linearization method for solving the overdefined system of square equations [4]. Nonlinear monomials are considered as new variables, and the technique of solution is similar roughly to Gaussian elimination method. The method is equivalent to Groebner basis method.

The most popular methods are based on Groebner bases technique[1] [7, 8]. These methods take the computation of syzygy as linear combination of two polynomials over the monoid of monomials such that least monomial are reduced. This takes the ordering of monomials and variables. The syzygies are to be computed for many pairs of polynomials of the basis and are joined to the basis. Since procedure of obtaining the syzygy is not invertible, the size of basis of ideal growths during the initial half of computation, and the size of syzygy growths too. The dependence of the memory size of the basis is near to exponent. Hence this method stays impractical too. Theoretically, this method gives the basis of prime ideal in the form $(x_1 + e_1, ..., x_n + e_n)$ if the solution is $(e_1, ..., e_n)$.

The Groebner basis computation is hard in practice due to some lacks.

[1] Initially the Groebner bases were applied for division of polynomials of several variables and were not directly connected with solving systems of polynomial equations. Computing common zeroes of polynomial ideals is in the area of algebraic geometry.

1. Computation of many syzygies leads to zero reduction (notice that this is possible only if the joining of the syzygy increases the size of the basis of ideal). Hence the choose of pairs of polynomials is an important problem.
2. Number of polynomials of the basis of ideal increases.
3. The average size of a syzygy increases.

If we can choose the pairs of polynomials in such a way that the size of the basis of ideal stays fixed (i.e. one of the basis polynomials is changed by the syzygy), the first two lacks are neglected.

Quite different method (agreeing and gluing) was proposed in [14]. This method deals with the set of zeroes of polynomials and has exponential time and memory complexity.

The characteristic set method [9, 17] uses the ordering of variables and uses the ordering of variables. The result of computation is triangle set of polynomials that form the basis of ideal: the last polynomial depends on one variable, the last but one polynomial depends on the two variables, etc. If the ideal is prime, then characteristic set is equivalent to Groebner basis computation. Indeed, if the triangle set of polynomials of the basis is obtained, then we can obtain the Groebner basis. The last polynomial of triangle system is the required polynomial. Substituting he zero of last polynomial in last but one polynomial we obtain the zero for next variable, etc. Back, if the basis ($x_1 + e_1, ..., x_n + e_n$) is given, then the triangle set can be easily obtained.

The goal of this paper is suggesting the method of invertible transformation of prime Boolean ideal. It is shown that any invertible transformation of the basis of ideal that fixes the size of the basis is determined by the action of invertible matrix. Hence for arbitrary system of n Boolean polynomials that have the unique zero ($e_1, ..., e_n$), there exists an invertible matrix over the polynomial ring that simplifies initial basis to the form ($x_1 + e_1, ..., x_n + e_n$). Back, for basis ($x_1 + e_1, ..., x_n + e_n$) there exists an invertible matrix that complicates the basis in such a way that computing of the zero of the transformed ideal seems hard.

Simplification allows eliminating of the variables without using resultants and obtaining the triangular basis similarly to characteristic set method. The proposed method is invertible and it admits variations. Also the simplification admits optimizing the choose of some polynomial pairs during the Groebner basis computation. Complication of the basis can be used in public-key cryptology, for example, in hidden field equations and isomorphism of polynomials.

2. Finite Boolean rings, their ideals and varieties

Boolean ring consists of idempotent elements that satisfy equality $x^2 = x$ [3]. Then $x^3 = x \times x^2 = x \times x = x$ and by induction $x^n = x$ for n ³ 1.

Boolean ring has characteristic 2 due to equalities

$$a + a = (a + a)^2 = a^2 + 2a^2 + a^2 = a + a + 2a,$$

hence $2a = 0$. Boolean ring is commutative due to equalities

$$a + b = (a + b)^2 = a^2 + ab + ba + b^2 = a + b + ab + ba,$$

hence $ab = ba$.

Finite Boolean ring is isomorphic to the ring of subsets of finite set with operations of symmetric difference (addition) and intersection (multiplication). If the finite set has n elements, then the set of its subsets has 2^n elements. Similarly the finite Boolean ring is isomorphic to the ring of n-dimensional binary vectors with bitwise addition and multiplication.

Boolean function is the map Å$_2$ ´ ... ´ Å$_2$ ® Å$_2$.

Define the ring G$_n$[**x**] = Å$_2$[x_1, …, x_n]/($x_1^2 + x_1$, …, $x_n^2 + x_n$) of Boolean polynomials. Since $f^2 + f = f(f + 1) = 0$ for any f Î G$_n$[**x**] each non-constant polynomial divides the zero. Since zero divisor cannot be invertible, the unique invertible element in G$_n$[**x**] is 1.

There is a bijection between Boolean functions and Boolean polynomials, #G$_n$[**x**] = 2^{2^n}.

Ideal of a ring is subset of the ring such that sum of any two elements of the ideal is in the ideal, and product of element of the ideal and element of ring is element of the ideal.

According to Hilbert's basis theorem any ideal of polynomial ring has finite basis [10]. Hence each ideal of a polynomial ring is a finite-dimensional module over this ring, and back, each finite-dimensional module is an ideal.

The set of common zeroes of a polynomial ideal A form the variety V(A). Ideals A, A^2, ... have the same variety, so there exists the largest ideal that has given variety, it is known as radical.

The Krull dimension of a ring is the maximal length of increased prime ideals. Since the ring G$_n$[**x**] has zero divisors, the zero ideal is not prime. Since the ring G$_n$[**x**] is finite, it is Noetherian and Artinian, and any its prime ideal is maximal [2]. Hence the dimension of G$_n$[**x**] is 0. The dimension of ideal A Ì G$_n$[**x**] is the dimension of the ring G$_n$[**x**]/A. So all ideals of G$_n$[**x**] have dimension 0.

Ideals of polynomial ring admit commutative and associative operations of addition Å, intersection and multiplication. In Noetherian ring any ideal admits unique primary decomposition as intersection of primary ideals. In G$_n$[**x**] any ideal is idempotent, so a primary ideal is prime, similarly any ideal is radical. Hence there is bijection between varieties and ideals.

In Artinian ring the intersection of ideals coincides with its product, hence any ideal of G$_n$[**x**] has unique factorization as product of different prime ideals. The variety of a prime ideal P Ì G$_n$[**x**] consists of single point (e_1, ..., e_n), and P is the set of polynomials that take zero at this point. If V(P) = (e_1, ..., e_n), then P = ($x_1 + e_1$, ..., $x_n + e_n$) = ($x_1 + e_1$) Å ... Å ($x_n + e_n$). The prime ideal P can be also given as principal ideal by one polynomial $\mathsf{P} = \left(1 + \prod_{i=1}^{n} (x_i + e_i + 1)\right)$. Hence any ideal as product of prime ideals can be considered as a principal one (given by a polynomial). Hence there is the bijection between the set of ideals and the set of polynomials of $\mathsf{G}_n[\mathbf{x}]$.

Any automorphism of $\mathsf{G}_n[\mathbf{x}]$ permutes prime ideals and back, any such permutation is an automorphism [15].

Quotient ring $\mathsf{G}_n[\mathbf{x}]/\mathsf{P}$ is isomorphic to Å$_2$ (has 2 elements). Similarly $\mathsf{G}_n[\mathbf{x}]/(\mathsf{P}_1\mathsf{P}_2)$ consists of 4 elements, $\mathsf{G}_n[\mathbf{x}]/(\mathsf{P}_1\mathsf{P}_2\mathsf{P}_3)$ consists of 8 elements, etc. So finite Boolean ring is isomorphic to $\mathsf{G}_n[\mathbf{x}]$ or to its quotient ring.

Addition and multiplication of ideals induces corresponding operations for their varieties: $V(\mathsf{A} \oplus \mathsf{B}) = V(\mathsf{A}) \cap V(\mathsf{B})$, $V(\mathsf{AB}) = V(\mathsf{A}) \cup V(\mathsf{B})$. If ideals are principal, then $(f) \oplus (g) = (f + g + fg)$, $(f)(g) = (fg)$.

Unique factorization of an ideal determines the (exact) division of ideals: ideal A divides ideal B if $V(\mathsf{A}) \subseteq V(\mathsf{B})$.

If $\mathsf{A} = \prod_{i \in I} \mathsf{P}_i$, $\mathsf{B} = \prod_{i \in J} \mathsf{P}_i$ — prime factorizations, then their greatest common divisor $\mathrm{GCD}(\mathsf{A},\mathsf{B}) = \mathsf{A} \oplus \mathsf{B} = \prod_{i \in I \cap J} \mathsf{P}_i$. If the right-hand product is empty, then A and B are relatively prime and $\mathsf{A} \oplus \mathsf{B} = (1)$. For relatively prime principal ideals $\mathsf{A} = (f)$, $\mathsf{B} = (g)$ there exist polynomials h_1, h_2 such that $fh_1 + gh_2 = 1$.

For example, ideals (f) and $(1 + fg)$ are relatively prime for arbitrary g. Here $fg + (1+ fg) = 1$.

Similarly to prime factorization we can represent additive decomposition of an ideal as a sum of ideals. Analog of prime ideal is additively irreducible ideal that takes zero in all but one points. So if P is prime ideal, then $\overline{\mathsf{P}} = 1 + \mathsf{P}$ (complement to P) is additively irreducible ideal; $\overline{\mathsf{AB}} = \overline{\mathsf{A}} \oplus \overline{\mathsf{B}}$. So an ideal can be uniquely represented as the sum of additively irreducible ideals.

There is a great number of representations of a prime ideal $\mathsf{P} \subset \mathsf{G}_n[\mathbf{x}]$ as a sum of n principal ideals. Each such representation corresponds to a system of n polynomials with the same varieties.

Consider division of ideals: the binary operation that for given ideals A, B gives A (mod B) so that inequalities hold

$$V(\mathsf{A}\overline{\mathsf{B}}) \subseteq V(\mathsf{A}(\mathrm{mod}\,\mathsf{B})) \subseteq V(\mathsf{A}). \tag{1}$$

In [15] it is shown that in $\mathsf{G}_n[\mathbf{x}]$ there are two divisions: the commonly used polynomial division in the domain Å$_2[x_1, \ldots, x_n]$: $f = gh$, $\deg(f) = \deg(g) + \deg(h)$, and algebraic-geometrical (AG) division: $f = gh$, if $V(f) = V(g) \cup V(h)$. Similarly we can determine the two divisions for ideals. If *V*(A) Ê *V*(B), then there exists ideal C such that A = BC and *V*(A) = *V*(B) È *V*(C). For example in G$_2$[*x*, *y*] ideal (*x* + *y*) is divided by ideal (*x* + *y* + *xy*) for AG-division: (*x* + *y*) = (*x* + *y* + *xy*)(1 + *xy*). In the case of monomials, polynomial exact division and AG exact division coincide.

In the case of AG-division the cardinality of *V*(A (mod B)) is minimal, $V(\mathrm{A}(\mathrm{mod}\,\mathrm{B}))=V(\mathrm{A}\overline{\mathrm{B}})$. In the case of polynomial division A (mod B) is not uniquely determined. For the unique computation of A (mod B) the monomial ordering and the computing Groebner basis of B are needed.

Though AG-division transforms the ring G$_n$[**x**] to Euclidean ring, the operation A (mod B) usually is hard to compute. But sometimes it admits effective computation, for example, if ideal B has monomial basis. To reduce polynomial *f* (mod B) it is sufficient to delete the monomials that are divided by some monomial of B.

Two kinds of division are caused by different considerations of field ideal F = (x_1^2 + x_1, …, x_n^2 + x_n). In the case of common polynomial division this ideal is external with respect to infinite integral domain of polynomials. To compute the variety of ideal over Å$_2$ the field ideal F must be joined to the polynomial ideal. In the case of AG-division the field ideal F is internal, it determines the ring G$_n$[**x**].

Computation of Groebner basis of finite Boolean ideal and Courtois's XL method use polynomial division. Ideal F here is used for deleting squares from the syzygies. Computation of the triangular basis(characteristic set) uses AG-division.

Theorem 1. In the ring G$_n$[**x**] equality holds A Å B = A (mod B) Å B both for polynomial division and AG-division.

Proof. If *V*(A) Ê *V*(B), then A Å B = A and using equality (1) we obtain $\mathrm{A}\overline{\mathrm{B}}$ Å B = A Å B. Hence we have A Å B = A (mod B) Å B for both divisions.

If *V*(A) Í *V*(B), then A Å B = B and A (mod B) = (0), so A Å B = A (mod B) Å B for AG-division. The same is true for polynomial division.

If *V*(A) Ç *V*(B) = Æ, then $V(\overline{\mathrm{B}})$ Ê $V(\mathrm{A})$, $V(\mathrm{A}\overline{\mathrm{B}})=V(\mathrm{A})$, $\mathrm{A}\overline{\mathrm{B}}=\mathrm{A}$. Hence for AG-division we have A Å B = A (mod B) Å B. In the case of polynomial division *V*(A (mod B)) Í *V*(A), hence A Å B = A (mod B) Å B.

The general case of ideals A, B can be reduced to three considered above cases by additive decomposition of ideal A. **n**

Corollary 1. A_1 Å A_2 Å … Å A_k = A_1 Å A_2 (mod A_1) Å A_3 (mod (A_1 Å A_2)) Å … Å A_k (mod (A_1 Å … $\mathsf{A}_{k\text{-}1}$)).

The proof is carried out by induction. **n**

In the ring $\mathsf{G}_n[\mathbf{x}]$ any polynomial has degree at most 1 for each variable and any non-constant polynomial is zero-divisor. Hence "naive" elimination of monomials (variables), when the coefficients of leading monomials (variables) are equalized by multiplication with some polynomials, leads to appearance of wrong solutions. In practice on some step of elimination we obtain equation $0*x_i = 0$, and x_i can be arbitrary.

Other obvious way of solving of the system $f_1 = 0, ..., f_k = 0$ that defines prime ideal, is the search of principal ideal defined by polynomial $1 + (1 + f_1)...(1 + f_k)$ or the (non-constant) monomial of minimal degree of it. Direct computation of the polynomial has exponential time and memory complexity. Obviously there exists some change of variables of kind x_i ® $1 + x_i$, that gives the prime ideal polynomial without constant term. Then the solution is (0, ..., 0). But search of such change of variables is equivalent to enumeration.

The most popular method of solving system of Boolean equations is the Groebner basis computation.

3. Monomial and binomial ideals

Define monomial ideal of $\mathsf{G}_n[\mathbf{x}]$ as ideal, which basis can be given by monomials, and binomial ideal, which basis can be given by binomials. Notice that recognition monomial or binomial ideal is nontrivial problem [6].

Monomial ideal without linear elements cannot be prime. For example, ideal of $\mathsf{G}_n[\mathbf{x}]$ generated by all monomials of degree 2 has $n + 1$ zeroes in points (0, ..., 0), (1, 0, ..., 0), (0, 1, 0, ..., 0), (0, ..., 0, 1).

No monomials are relatively prime. For example common divisor of monomials x_1, x_2x_3 is nonzero ideal $(x_1 + x_2x_3 + x_1x_2x_3)$.

Theorem 2. Ideal has monomial basis iff the Boolean function that determines the principal ideal can be represented by disjunctive normal form (DNF) where no conjunction contains an inversion.

Proof. Let $(m_1, ..., m_k)$ is the basis of monomial ideal. Any monomial is conjunction without inversions. The Boolean function that gives the principal ideal takes zero iff each conjunction takes zero. Hence the principal ideal is given by Boolean function m_1 Ú ... Ú m_k. Back, let Boolean function is the DNF without inversions. Any conjunction is monomial, and the Boolean function determines monomial ideal. **n**

Any prime ideal is binomial by definition.

Binomial ideal ha important property: all syzygies of the basis elements are binomials. So the Groebner basis can be computed relatively easy, and the system of Boolean binomial equations has effective solution.

Horn DNF is the disjunction of conjunctions where each conjunction contains at least one inverse variable [3].

Theorem 3. An ideal is binomial iff it is generated by Horn DNF.

Proof. Let the basis of ideal is given by binomials ($m_1 + n_2$, $m_2 + n_2$, ..., $m_k + n_k$), and m_i, n_i are monomials. Then binomial $x_{i_1}x_{i_2}...x_{i_r} + x_{j_1}x_{j_2}...x_{j_s}$ is the Horn DNF of kind

$$x_{i_1}x_{i_2}...x_{i_r}(\overline{x_{j_1}} \vee \overline{x_{j_2}} \vee ... \vee \overline{x_{j_s}}) \vee x_{j_1}x_{j_2}...x_{j_s}(\overline{x_{i_1}} \vee \overline{x_{i_2}} \vee ... \vee \overline{x_{i_r}}).$$

Sum of binomials in the sense of ideal addition also give Horn DNF.

Back, let $m_1\overline{x_{i_1}} \vee m_2\overline{x_{i_2}} \vee ... \vee m_r\overline{x_{i_r}}$ is Horn DNF, where m_i are monomials. Then $m_j\overline{x_{i_j}} = m_j(1+x_{i_j}) = m_i + m_i x_{i_j}$ is binomial. So any Horn DNF determines corresponding binomial ideal. ■

Satisfiability problem for Horn DNF has effective solution [3].

4. Transformation of the basis of Boolean ideal

Let the length of the input **x** and the output **y** of random binary map **y** = F(**x**) is n. Estimate the probability that for given output $\mathbf{y}^*$ there exists unique input under the condition that such input exists.

There are $N = 2^n$ possible inputs and outputs. The map F is defined by the string of N output n-bit words for all possible N inputs, so there are N^N possible maps. The number of inputs for given $\mathbf{y}^*$ is the number of words $\mathbf{y}^*$ in the string. The number of strings without $\mathbf{y}^*$ is $(N - 1)^N$. The probability that random map has no input **x** such that $\mathbf{y}^* = F(\mathbf{x})$ is $((N - 1)/N)^N = e^{-1}$, $e = 2.71828$. The number of maps that have at least one solution is $((e - 1)/e)N^N$.

Estimate the number of maps that have unique solution $\mathbf{y}^* = F(\mathbf{x})$. If the word $\mathbf{y}^*$ is in the first position of the string, there are $(N - 1)^{N-1}$ such strings. Similarly there are $(N - 1)^{N-1}$ strings that have $\mathbf{y}^*$ in the second position, etc. So there are $N(N - 1)^{N-1} \approx (N - 1)^N$ strings that contain only one word $\mathbf{y}^*$. Hence the probability of event that a random map F has unique solution of equation $\mathbf{y}^* = F(\mathbf{x})$ under the condition that at least one solution exists, is

$$\left(\frac{N-1}{N}\right)^N \frac{e}{e-1} = \frac{1}{e-1} = 0.582.$$

We can consider the hash function or the block cipher with given input as the random map. Usually it is required to compute the input of hash function (to compute the key) under the condition that desired input (the key) exists. Hence the obtained estimation gives the probability that polynomial ideal defined by Boolean polynomials that describe the hash function transformation or the block cipher encryption is prime.

4.1. The fixed size of the basis

Ideal of $\mathsf{G}_n[\mathbf{x}]$ has finite basis and by definition is the module over $\mathsf{G}_n[\mathbf{x}]$, and back, any module over $\mathsf{G}_n[\mathbf{x}]$ is the ideal. The bases of the ideal and of the module coincide. Hence changing the basis of ideal that fixes the size of the basis can be determined as invertible matrix over $\mathsf{G}_n[\mathbf{x}]$ [5].

An automorphism of $\mathsf{G}_n[\mathbf{x}]$ permutes prime ideals and hence usually changes the ideal. But multiplication by invertible matrix L fixes the ideal.

Theorem 4. Two bases $\mathbf{f} = (f_1, ..., f_k)$ and $\mathbf{g} = (g_1, ..., g_k)$ generate the same ideal in $\mathsf{G}_n[\mathbf{x}]$ iff there exists invertible matrix L of size k over $\mathsf{G}_n[\mathbf{x}]$ such that $\mathbf{f} = L\mathbf{g}$. Then $\mathbf{g} = L^{-1}\mathbf{f}$.

Proof. Any ideal is the module over $\mathsf{G}_n[\mathbf{x}]$ and any module is the ideal, the bases of the module and the ideal coincide. If $\mathbf{f}$ is the basis of the module (ideal) and L is invertible matrix over $\mathsf{G}_n[\mathbf{x}]$ then $\mathbf{g} = L\mathbf{f}$ is also the basis of the module (ideal) and the transformation is given by matrix L. Inverse transformation L^{-1} gives $\mathbf{f} = L^{-1}\mathbf{g}$.

Back, let $\mathbf{f}$, $\mathbf{g}$ are the bases of the ideal. Since the basis if ideal is the basis of the module, then $\mathbf{f}$, $\mathbf{g}$ are the module bases. Then there exists invertible matrix L such that $\mathbf{f} = L\mathbf{g}$.

Proof that multiplication by L fixes the ideal. Indeed, let $\mathbf{f} = L\mathbf{g}$. Then by definition $f_i \in (g_1, ..., g_k)$ and hence $(f_1, ..., f_k) \subseteq (g_1, ..., g_k)$. On the other hand $\mathbf{g} = L^{-1}\mathbf{f}$ and $(f_1, ..., f_k) \supseteq (g_1, ..., g_k)$. So $(g_1, ..., g_k) = (f_1, ..., f_k)$. ■

The theorem 4 is true for any k. If $k = 1$, then the onliest invertible matrix is the constant 1. So principal ideal is determined uniquely.

Let $\mathbf{f}$ is the basis of prime Boolean ideal P that has variety $(e_1, ..., e_n)$ and $\mathbf{f} = (f_1, ..., f_n)$. Then the basis of P can be written in simplified form $\mathbf{g} = (x_1 + e_1, ..., x_n + e_n)$ and there exists an invertible matrix L such that $\mathbf{g} = L\mathbf{f}$. Hence the problem of simplification of the basis of prime Boolean ideal is closely related with the problem of representing matrix L as product $L = L_k...L_2L_1$ such that $\mathbf{f}_i = L_i...L_1\mathbf{f}$ is less then $\mathbf{f}_{i-1}$ (in some sense).

The common technique takes minimization of leading monomial (in Groebner basis algorithms) or the number of variables (in characteristic set algorithms).

In [12] it is shown that for any set of relatively prime polynomials $(f_1, ..., f_k)$ there exists an invertible matrix with first row $(f_1, ..., f_k)$.

Sometimes during the computation the Groebner basis one computes syzygy of two polynomials and tries to change one of the initial polynomials by the syzygy.

This is equivalent to multiplication the initial 2-element basis by matrix $L = \begin{pmatrix} a & b \\ 0 & d \end{pmatrix}$, $a, b, d \in \Gamma_n[\mathbf{x}]$. Matrix L is invertible iff $a = d = 1$. So usually the element of the basis cannot be changed by the syzygy. This leads to increasing the size of the intermediate basis.

Number of invertible matrices 2*2 is very large. Many of them can be obtained as products of matrices $S_h = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}$ for arbitrary h and $T = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, i.e. $T^{e_1} S_{h_m} T S_{h_{m-1}} ... T S_{h_1}$ for $e_1 \in \{0, 1\}$. Matrices S, T satisfy following equalities:

$$S_gS_h = S_{g+h} = S_hS_g,\ \ S_h^{-1} = S_h,\ \ T^{1} = T,$$

$$S_{h_2}TS_{h_1} = \begin{pmatrix} h_2 & 1+h_1h_2 \\ 1 & h_1 \end{pmatrix},\quad S_{h_3}TS_{h_2}TS_{h_1} = \begin{pmatrix} 1+h_2h_3 & h_1+h_3+h_1h_2h_3 \\ h_2 & 1+h_1h_2 \end{pmatrix},$$

$$S_h\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a+ch & b+dh \\ c & d \end{pmatrix},\ \ \begin{pmatrix} a & b \\ c & d \end{pmatrix}S_h = \begin{pmatrix} a & b+ah \\ c & d+ch \end{pmatrix}, \qquad (2)$$

$$T\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} c & d \\ a & b \end{pmatrix},\ \ \begin{pmatrix} a & b \\ c & d \end{pmatrix}T = \begin{pmatrix} b & a \\ d & c \end{pmatrix}.$$

Invertible matrices with constant elements form group of order 6.

Define monomial matrix as a square matrix whose elements are monomials. The equalities above show that product of monomial matrices usually is not a monomial one. Usual computation of syzygies takes monomial matrices.

4.1.1. Application for Groebner basis computing

It is known that the complexity of Groebner basis computation strongly depends on the sequence of polynomial pairs that give a syzygy [6]. The influence of the choose on the complexity is caused by different reasons. One of the reasons is the incrementing of the size of the ideal basis. Hence we can distinguish unsuccessful choose of pair of polynomials that increases the size of the basis, and successful choose fixes the size of the basis.

Using matrix S_h one can obtain easily some pairs that simplifies the polynomial system (in the sense of Groebner basis) and hence determine the optimization of Groebner basis computation.

Computation of Groebner basis takes the linear ordering of monomials. Each polynomial f has a leading monomial LM(f). Similarly a set of polynomial has a leading monomial.

Let f_1, f_2 are polynomials of the basis of an ideal basis such that LM(f_1) $\equiv$ 0 (mod LM(f_2)) and.[2] Then $\mathrm{LM}(f_1) \succ \mathrm{LM}(f_2)$. Let monomial h is determined by equation LM(f_1) = h*LM(f_2). Then multiplication of (f_1, f_2) by the monomial matrix

[2] Remember that exact division of monomials coincides both for polynomial division and for AG-division.

S_h deletes the LM(f_1) and the size of ideal does not increase. In this case the reduction of intermediate syzygies can be rejected. Here $\mathrm{LM}(\mathbf{f})\ \mathbf{f}\ \mathrm{LM}(S_h\mathbf{f})$.

Algorithm 1. Optimizing the sequence of pairs of polynomials during Groebner basis computation.

Input. Ideal basis (f_1, ..., f_k) and monomial order.

Output: Ideal basis (f_1, ..., f_k) with reduced leading monomials in some polynomials.

Method.

1. Order the polynomials (f_1, ..., f_k) according to monomial order.
2. For i = 1 to k, j = i + 1 to k define a pair (f_i, f_j) such that LM(f_i) is divided by LM(f_j).
3. Compute monomial h from equation LM(f_i) = h*LM(f_j).
4. Change f_i ® f_i + hf_j, that deletes the leading monomial, test that $\mathrm{LM}(f_i + hf_j)\ \mathbf{p}\ \mathrm{LM}(f_i)$.
5. If no such pairs (f_i, f_j) exist, stop. **n**

Algorithm 1 computes **f** ® S_h**f** for appropriate pairs **f** = (f_i, f_j). This algorithm can be applied after any computation of a syzygy.

Usually a syzygy of two polynomials f_1, f_2 is computed in such a way that the sum h_1LM(f_1) + h_2LM(f_2) = 0 for appropriate monomials h_1, h_2. But this is not necessary indeed, it is sufficient to obtain LM(h_1f_1) + LM(h_2f_2) = 0.[3] After multiplication by monomial h, two monomials of hf_i can coincide, hence their sum becomes zero. This allows to modify algorithm 1 that probably eliminates the LM(f_1) and does not increase the number of basis polynomials.

Algorithm 2. Optimizing the sequence of pairs of polynomials during Groebner basis computation.

Input. Ideal basis (f_1, ..., f_k) and monomial order.

Output: Ideal basis (f_1, ..., f_k) with reduced leading monomials in some polynomials.

Method.

1. Order the polynomials (f_1, ..., f_k) according to monomial order.
2. For i = 1 to k, j = i + 1 to k define a pair (f_i, f_j) such that $hf_j + f_i\ \mathbf{p}\ f_i$ for some $h\ \mathbf{p}\ \mathrm{LM}(f_i)$ including h = LM(f_i).
3. Change f_i ® f_i + hf_j, that deletes the leading monomial.
4. If no such pairs (f_i, f_j) exist, stop. **n**

[3] Notice that generally hLM(f) and LM(hf) are different in the Boolean ring.

If the degree of f_i is near to the number of variables of f_i, f_j, then the sum of some monomials of hf_j become zero. If h = LM(f_i), this technique decreases the length of the f_i + LM(f_i)f_j comparatively to the length of f_i and to algorithm 1.

Example 1. Elimination of the leading monomial without increasing the size of the ideal basis and the size of polynomial f_1.

Let $f_1 = x_1x_2x_3 + x_1x_3 + x_2$; $f_2 = x_1x_2x_4 + x_1x_2 + x_1 + x_3x_4 + 1$. We use the order: x_i **f** x_{i+1}, mx_i **f** mx_{i+1} for monomial m. LM(f_1) = $x_1x_2x_3$, LM(f_2) = $x_1x_2x_4$, LM(f_1) ¹ 0 (mod LM(f_2)), LM(f_1) **f** LM(f_2). Let $L=\begin{pmatrix}1 & \mathrm{LM}(f_1)\\ 0 & 1\end{pmatrix}$. Then first element of new basis $f_1 + hf_2 = x_1x_3 + x_2$ is more short then the polynomial f_1. We get new basis (f_1 + hf_2, f_2), the leading monomial of the first polynomial is deleted and LM($f_1 + hf_2$) **p** LM(f_1). **n**

Invertible 2*2 matrix with all nonzero elements is defined by the next theorem.

Theorem 5. Invertible matrix $L=\begin{pmatrix}a & b\\ c & d\end{pmatrix}$ with all nonzero elements is uniquely defined by the quadriple of arbitrary elements a, d, g_1, g$_2$ Î G$_n$[**x**], where $b = 1 + adg_1$, $c = 1 + ad + adg_1 + adg_1g_2$.

Proof. Matrix L is invertible if $ad + bc = 1$, so polynomial ad is the complement to polynomial bc. Then a, d can be arbitrary, but (b) Ê (1 + ad), (c) Ê (1 + ad), and (bc) = (1 + ad). Those conditions can be rewritten as

$$b = 1 + adg_1,$$

$$c = (1 + ad)(1 + g_1 + g_1g_2) = 1 + ad + adg_1 + adg_1g_2.$$

Then $bc = 1 + ad$ for arbitrary g_1, g_2. Elements of L are determined uniquely. Notice that the inverse statement is not true: for given matrix L elements g_1, g_2 generally are not unique. If $g_2 = 0$, then $g_1 = (1 + b)/(ad)$ (it is a polynomial in the sense of AG-division). In this case if the quadriple (a, d, g_1, 0) for given L exists (i.e. if $c = 1 + ad + adg_1$), it is unique.

In the conditions of the theorem the inverse matrix is $L^{-1}=\begin{pmatrix}d & b\\ c & a\end{pmatrix}$. **n**

The invertible matrix L with given first row exists iff ideals (a) and (b) are relatively prime. Indeed, if $a = a_1h$, $b = b_1h$, then $ad + bc = h(a_1d + b_1c)$, and matrix L cannot be invertible. If ideals (a), (b) are relatively prime, then we can choose arbitrary polynomial d relatively prime to b, and corresponding polynomial c, relatively prime to ad that gives the matrix.

Let **f** = (f_1, f_2) and LM(**f**) = max(LM(f_1), LM(f_2)). Consider the general case when multiplication by invertible matrix L decreases the leading monomial of the basis: LM(L**f**) **p** LM(**f**). As it is shown above, if LM(f_1) ¹ 0 (mod LM(f_2)), then matrix L has no zero elements and is not a monomial one. It is sufficient to consider the case when LM(f_1), LM(f_2) have no common variables.

Theorem 6. Let the monomials LM(f_1), LM(f_2) have no common variables, LM(f_i) = GCD(LM(f_1, f_2))h_i, and h_1, h_2 are relatively prime. Then

$$\mathrm{GCD}(\mathrm{LM}(f_1), \mathrm{LM}(f_2)) = \mathrm{LM}(f_1) + \mathrm{LM}(f_2) + \mathrm{LM}(f_1)\mathrm{LM}(f_2),$$

$$h_1 = \frac{\mathrm{LM}(f_1)}{\mathrm{GCD}(\mathrm{LM}(f_1), \mathrm{LM}(f_2))} = 1 + \mathrm{LM}(f_2) + \mathrm{LM}(f_1)\mathrm{LM}(f_2),$$

$$h_2 = \frac{\mathrm{LM}(f_2)}{\mathrm{GCD}(\mathrm{LM}(f_1), \mathrm{LM}(f_2))} = 1 + \mathrm{LM}(f_1) + \mathrm{LM}(f_1)\mathrm{LM}(f_2).$$

Proof follows from the direct computation. **n**

If the monomials LM(f_1), LM(f_2) have common variables, then we represent them as products of common variables and the monomials without common variables. Hence the greatest common divisor is to be multiplied by the product of common variables. But h_1, h_2 do not change if the leading monomials have common variables.

Further in theorem 7 it is shown that h_1, h_2 are relatively prime to each other and to f_1, f_2. Hence we can obtain the invertible matrix $L = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, $a = h_2$, $b = h_1$. Choose arbitrary d relatively prime to b and compute c using theorem 5. Then the leading monomials of $af_1 + bf_2$ will be rejected. But vector $L\mathbf{f}$ can be more then the initial vector **f** (in the sense of monomial ordering). Computing such pairs seems to be hard.

4.1.2. Application for eliminating of variables

Invertible transformation of the basis of prime ideal allows eliminating of variables without incrementing the size of the basis and without using resultants. This method is similar to computation of the characteristic set according to [8, 11] but it has additional degrees of freedom.

Theorem 7. For ideals A, B Ì G_n[**x**] next equalities hold: A = GCD(A, B)A$_1$, B = GCD(A, B)B$_1$, where ideals A$_1$, B$_1$ are relatively prime to each other and to GCD(A, B), if AG-division is used.

Proof. Due to unique prime factorization there exists GCD(A, B) as product of all common prime ideal in prime factorization of A, B. Then A = GCD(A, B)A$_1$, B = GCD(A, B)B$_1$. If ideals A$_1$, B$_1$ are not relatively prime, they have common prime divisor, that is impossible. Since any ideal is squarefree, A$_1$ and B$_1$ are relatively prime to GCD(A, B). **n**

Now we can present algorithm that eliminates a variable and fixes the size of the basis. This algorithm is near to algorithm proposed in [11], but differs in methods and possesses additional freedom degrees.

Algorithm 3. Elimination of variable x for two polynomials, both containing x.

Input. Polynomial $f_1 = f_{10} + xf_{11}$, $f_2 = f_{20} + xf_{21}$.

Output: Matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ and polynomials F_1, F_2, where F_1 does not depend on x.

Method.

1. Factor (f_{11}), (f_{21}) as product of prime ideals.
2. Compute GCD(f_{11}, f_{21}).
3. Compute relatively prime polynomials (under AG-division) $h_1 = f_{11}/\mathrm{GCD}(f_{11}, f_{21})$, $h_2 = f_{21}/\mathrm{GCD}(f_{11}, f_{21})$.
4. Set $a = h_2$, $b = h_1$ and compute prime factorization of ideal (b).
5. Using factorization p. 4, choose arbitrary polynomial d relatively prime to b.
6. Choose a polynomial g_1 so that inequality holds $V(1 + b) \setminus V(ad) \subseteq V(g_1) \subseteq V(1 + b)$ (that gives $b = 1 + adg_1$). Choose arbitrary polynomial g_2 and compute the polynomial $c = 1 + ad(1 + g_1 + g_1g_2)$.
7. Compute $\begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$. ■

Appropriate choosing of auxiliary polynomials g_1, g_2 allows simplifying element F_2 of the output basis **F** in some proper sense (minimizing its length or degree, deleting the unnecessary monomials, etc.).\

This algorithm can be also explained in the language of linear algebra. Notice that $\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} f_{10} & f_{11} \\ f_{20} & f_{21} \end{pmatrix}\begin{pmatrix} 1 \\ x \end{pmatrix}$, $\begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} f_{10} & f_{11} \\ f_{20} & f_{21} \end{pmatrix}\begin{pmatrix} 1 \\ x \end{pmatrix}$. Variable x is eliminated if the right-up element in matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} f_{10} & f_{11} \\ f_{20} & f_{21} \end{pmatrix}$ is zero: $af_{11} = bf_{21}$.

Example 2. Elimination of a variable.

Let $f_1 = 1 + x_1 + x_4 + x_1x_2 + x_2x_3$, $f_2 = x_2x_4 + x_1x_4 + x_1x_2x_3$, $\mathbf{f} = (f_1, f_2)$. Eliminate variable x_1. Then $f_1 = f_{10} + f_{11}x_1 = (1 + x_4 + x_2x_3) + x_1(1 + x_2)$, $f_2 = f_{20} + f_{21}x_1 = x_2x_4 + x_1(x_4 + x_2x_3)$.

Polynomials f_{11}, f_{21} have next prime factorization in the ring with variables x_2, x_3, x_4: $(f_{11}) = P(1, 0, 0)P(1, 0, 1)P(1, 1, 0)P(1, 1, 1)$; $(f_{21}) = P(0, 0, 0)P(1, 0, 0)P(0, 1, 0)P(1, 1, 1)$, where $P(e_2, e_3, e_4)$ has the zero at point (e_2, e_3, e_4).

Compute GCD(f_{11}, f_{21}) = $P(1, 0, 0)P(1, 1, 1) = (1 + x_2(1 + x_3)(1 + x_4))(1 + x_2x_3x_4) = 1 + x_2 + x_2x_3 + x_2x_4$.

Compute factorization of f_{11}, f_{21} as product of relatively prime divisors: $f_{11} = (1+ x_2 + x_2x_3 + x_2x_4)(1 + x_2x_3 + x_2x_4)$; $f_{21} = (1 + x_2 + x_2x_3 + x_2x_4)(x_2 + x_4 + x_2x_4)$, second divisors of f_{11}, f_{21} are relatively prime.

For elimination of x_1 we must have $af_{11} + bf_{21} = 0$, where a, b are relatively prime. Let $a = x_2 + x_4 + x_2x_4$, $b = 1 + x_2x_3 + x_2x_4$ (these are the second divisors of f_{11}, f_{21}). Choose $d = x_2$ (d can be arbitrary but relatively prime to b), then $ad = x_2$. Compute $g =$ P(1, 0, 0)P(1, 1, 1) $= 1 + x_2 + x_2x_3 + x_2x_4$, $c = 1 + x_2 + x_2x_3 + x_2x_4$. We obtain the next matrix

$$L = \begin{pmatrix} x_2 + x_4 + x_2x_4 & 1 + x_2x_3 + x_2x_4 \\ 1 + x_2 + x_2x_3 + x_2x_4 & x_2 \end{pmatrix}.$$

Then $\mathbf{F} = L\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} x_2 + x_2x_3 + x_2x_4 + x_2x_3x_4 \\ 1 + x_1 + x_2 + x_4 + x_1x_2 + x_1x_2x_3 + x_1x_2x_4 \end{pmatrix}$. The variable x_1 is eliminated from the first polynomial of the basis.

Inverse matrix is $L^{-1} = \begin{pmatrix} x_2 & 1 + x_2x_3 + x_2x_4 \\ 1 + x_2 + x_2x_3 + x_2x_4 & x_2 + x_4 + x_2x_4 \end{pmatrix}$ and $L^{-1}\mathbf{F} = \mathbf{f}$. ■

Notice that algorithm 3 gives additional degree of freedom (choosing of polynomials d, g_1, g_2). This allows optimizing of polynomial F_2.

In order to obtain the triangular basis in the case of n equations of n variables it is desirable (but not necessary) to order the variables.

We choose arbitrary equation as the first basic one. This equation must contain the variable that will be eliminated first.

Next we consider $n - 1$ pairs of equations, where the first one is the first basic equation, and apply algorithm 3 to each pair. This gives $n - 1$ polynomials without first variable. Similarly we eliminate second variable, etc. If the ideal is prime, then the last equation must be of kind $x + e = 0$, and we obtain the solution for last variable. Substituting this value to last but one equation we obtain the solution for last but one variable, etc.

So computing the basis of prime ideal that consists of linear binomials takes $O(n^2)$ applications of algorithm 3. In order to get polynomial complexity of the solution computing, each application of algorithm 3 must have polynomial complexity.

Since the initial basis is sparse, the complexity of algorithm 3 for eliminating initial few variables can be polynomial. Also final part of triangular basis can be computed comparatively easy. But the large part of polynomials of the triangular basis "in the middle part" of computation may have very large length.

The farther reduction of complexity can be obtained if we initially compute the monomial (binomial, trinomial) ideal ˆº and consider elements of triangular basis as residues modulo ˆº. The reduction of a polynomial modulo monomial ideal is very easy — it is sufficient to delete the monomials that are divided by the monomials of the ideal. Similarly one can reduce a polynomial modulo binomial or trinomial ideal. In this case initial (probably prime) ideal P is represented as a sum P = ˆº Å (P (mod ˆº)). The solution is searched initially for ideal (P (mod ˆº)), that may not be prime, and then it is lifted to ideal P by joining polynomials of ˆo to the basis.

In [15] it is shown that reduction of the random polynomial modulo monomial approximation of the ideal of AES *S*-box decreases the length of the polynomial about 30%. The average degree of a polynomial is decreased too.

4.2. Changing the size of the basis

The size of the basis can be reduced by deleting linearly dependent elements or by joining some polynomials in one principal ideal.

Increasing the size of the basis can be obtained by additive decomposition of some polynomials of the basis as sum of two or more principal ideals. The number of such additional decompositions usually is very large.

5. Complication of ideal

Consider the problem that is the inverse to problem of solving equations.

Assume we have an ideal, for example prime ideal of the form P $= (x_1 + e_1, ..., x_n + e_n)$ or other ideal that has a basis of short polynomials. How can we complicate it so that the point of variety is hard to compute?

The answer is given by theorem 4: it is sufficient to choose invertible matrix and to multiply the basis as vector by the matrix.

It is known that invertible matrix can be factored as product of upper and lower triangular matrices and permutation matrix [1]. It is known also that both upper and lower triangular matrices (a_{ij}) with property $a_{ii} = 1$ for all i are invertible. This gives the algorithm that complicates the ideal.

Choose at random upper and lower invertible triangular matrices and permutation matrix. Compute the basis of ideal by those matrices and obtain the transformed basis.

Invertible matrix L of size n has $\mathrm{O}(n^2)$ elements, but the basis transformation is defined by $O(n)$ equations. Hence there are many invertible matrices L that satisfy equality $L\mathbf{f} = \mathbf{g}$ for given $\mathbf{f}$, $\mathbf{g}$.

Usually in symmetric cryptography the basis $\mathbf{g} = (x_1 + e_1, ..., x_n + e_n)$ and the basis $\mathbf{f}$ is a sparse one — it corresponds to encryption equations. More precisely, each polynomial of $\mathbf{f}$ corresponds to some encryption round, hence it depends on input and output variables for given round and on key bits. In this case t is useful to consider matrix L as a block one. Similarly decomposition of matrix L should be considered for block matrices.

Such complicated prime ideal can be used in public key cryptosystems, for example in cryptosystem based on hidden field equations and isomorphism of polynomials [13].

References

1. Aho A., Hopcroft J., Ullman J. The design and analysis of computer algorithms. — Addison-Wesley, 1974.
2. Atiyah M., Macdonald L. Introduction to commutative algebra. — Addison-Wesley, 1969.
3. Crama Y., Hammer P. Boolean functions: theory, algorithms and applications. — Cambridge university press, NY, USA, 2011.
4. Courtois, N. T.; Pieprzyk, J. Cryptanalysis of block ciphers with overdefined systems of equations. — Advances in Cryptology – AsiaCrypt 2002, Lecture Notes in Computer Science 2501, Springer-Verlag, pp. 267–287.
5. Cox D., Little J., O'Shea D. Ideals, varieties and algorithms, — Springer, 2007.
6. Eisenbud D., Sturmfels B. Binomial ideals // Duke mathematical journal, v. 84 (1996), no. 1, pp. 1- 45.
7. Faugere J.-C. New Efficient Algorithm for Computing Groebner Basis without Reduction to Zero (F5). — Proceedings of the 2002 international symposium on Symbolic and algebraic computation, ACM Special Interest Group on Symbolic and Algebraic Manipulation, 2002.
8. Gao S., Guan Y., Volny F. A new incremental algorithm for computing Groebmer basis // ISSAC'10, ACM press, 2010, pp.13- 19.
9. Gao X.-S., Huang Z. Efficient characteristic set algorithms for equation solving in finite fields and application in analysis of stream ciphers. IACR e-print archive, 2009-637.
10. Harshorn R. Algebraic geometry. GTM, 1997.
11. Huang Z, Sun Y., Lin D. On the efficiency of solving Boolean polynomial systems with characteristic set method, http://arxiv.org/abs/1405.4596, 2014.
12. Lissner D. Matrices over polynomial rings // Transaction of the AMS, 98 (2), 1961, pp. 285- 305.
13. Patarin J. Hidden field equations (HFE) and isomorphism of polynomials (IP): two new families of asymmetric algorithms // Advances in Cryptology, EUROCRYPT'96, pp. 33- 48.
14. H. Raddum and I. Semaev. New technique for solving sparse equation systems. Cryptology e-print archive, report 2006/475, 2006 // Available at http: // e-print.iacr.org/2006/475.
15. Rostovtsev A., Mizyukin A. On Boolean ideals and varieties with application to algebraic attacks // IACR eprint archive, report 2012/151, see also "Problemy informacionnoi bezopasnosty. Computernye systemy, SPb, 2012, no 1, pp. 82- 89" (in Russian).
16. Sturmfels B. Solving systems of polynomial equations. — University of Barkley, California, USA, AMS, 2002.
17. Wu W. T. On the decision problem and the mechanization of the theorem proving in elementary geometry. — Sci. Sinica 21, pp. 159- 172, 1978.
18. Zhegalkin polynomials, Wikipedia article // http://en.wikipedia.org/wiki/Zhegalkin_polynomial