Abstract

In this paper we propose a new algorithm for computing Groebner basis for a system of multivariate polynomial equations describing a cryptosystem. The objective for designing this algorithm is to reduce the degree and number of polynomials resulting in a Groebner basis, which appears in the output of the algorithm. To attain this goal, a new division algorithm is proposed. The proposed algorithm, improved Buchberger and F4 algorithm have been applied to the system of algebraic equations extracted from the Courtois Toy Cipher and their efficiencies have been compared. The results show that the proposed algorithm has advantages over improved Buchberger and F4 algorithms from the view point of the number of polynomials within the obtained Groebner basis and computational (time) complexity.

Keywords: Groebner basis, algebraic cryptanalysis, Buchberger algorithm

1 Introduction

It is known that any cryptosystem can be described as a system of multivariate polynomial equations over GF(2). Notably, during the last decade algebraic cryptanalysis has been applied to public key cryptosystems, like the Matsumoto-Imai cryptosystem [1] and the primary version of HFE cryptosystem [2]. Unlike linear and differential cryptanalyses, the security of cryptosystem against algebraic cryptanalysis does not increase exponentially with increasing number of rounds. Once algebraic cryptanalysis attracted the attention of cryptographers, several methods were proposed to develop and generalize the linearization of a system of multivariate polynomial equations. While
algebraic cryptanalysis based on linearization was successful against stream ciphers, they did not manifest considerable success against block ciphers such as AES and Serpent. Generalizing and improving methods based on linearization are among solutions proposed for this issue, of which the XL [3] and XSL [4] algorithms are considered as examples. As the studies indicate, the XL algorithm is not as efficient as expected. On the other hand, the accuracy and precision of the XSL algorithm are not confirmed by the scientific community. So we are in need for better systematic algorithms. Groebner bases are validated tools for solving multivariate polynomial equations and, as a model, may replace the heuristic XL and XSL algorithms. The outline of the paper is as follows: Section 2 involves the necessary definitions and notations used in this paper. Section 3 is devoted to introducing Groebner basis. Section 4 deals with a new algorithm for computing Groebner basis. After a brief description of the Courtois Toy Cipher structure in Section 5, the authors extract its algebraic equations and compare the performance of the three algorithms in computing the Groebner basis.

2 Definitions and Notations

Definition 2.1 The monomial $x_1^{\alpha_1}...x_n^{\alpha_n}$ is shown with $x^\alpha$, where $x = (x_1, ..., x_n)$ and $\alpha = (\alpha_1...\alpha_n)$. Any polynomial $f(x)$ in $K[x_1...x_n]$, where $K[.]$ is a polynomial ring over the field, can be depicted as follows:

$$f(x) = \sum_{\alpha} c_{\alpha}x^\alpha : c_{\alpha} \in K$$  \hspace{1cm} (1)

If $\alpha = (\alpha_1, ..., \alpha_n)$ then $|\alpha| = \alpha_1 + ... + \alpha_n$ is called the degree of monomial $x^\alpha$. Also $T(f)$ is considered to be the set of all monomials of $f$.

Definition 2.2 Let $K$ be a field and $f_1, \ldots, f_n \in K[x_1, \ldots, x_n]$, by definition [5]:

$$V(f_1, \ldots, f_m) = (a_1, \ldots, a_n)|a_i \in K, f_i(a_1, \ldots, a_n) = 0 : 1 \leq i \leq m,$$  \hspace{1cm} (2)

where $V(f_1, \ldots, f_m)$ is called the variety of $f_1, \ldots, f_m$.

Definition 2.3 The subset $I$ of $K[x_1, \ldots, x_n]$ is called ideal if [5]:

- $\forall f, g \in I : f + g \in I$
- $\forall h \in K[x_1, \ldots, x_n], f \in I : h \cdot f = f \cdot h \in I$

The leading terms of polynomials are particularly important in the context of Groebner bases. Therefore, we define order relation for the set of all monomials of the ring $K[x_1, \ldots, x_n]$. Various orders may be defined. In this paper, we define one of such order relations, called the lexicographical order.
Algebraic attacks from a Groebner basis perspective

Definition 2.4 (lexicographical order) Let $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_n)$ belong to $\mathbb{Z}_{\geq 0}^n$. Then $\alpha \succ_{\text{lex}} \beta$ (read $\alpha$ under the lexicographical order is greater than $\beta$) when the first nonzero element of the vector $\alpha - \beta$ from the left is positive. Also $x^\alpha \succ_{\text{lex}} x^\beta$ when $\alpha \succ_{\text{lex}} \beta$ [5].

Definition 2.5 (leading term of a polynomial) Let $f(x) = \sum_c c_\alpha x^\alpha : c_\alpha \in K$ is non-zero and " $\succ$ " is the order relation defined for the monomials of the polynomial $f(x)$. The greatest monomial in $f(x)$, regarding to the order relation " $\succ$ ", is called the leading monomial for the polynomial $f(x)$ and is represented by $\text{LM}(f)$. Also the set $M(f)$ consists of all monomials of $f(x)$ and $T(f)$ denote the set of all terms of $f(x)$. The coefficient of the leading monomial is represented by $\text{LC}(f)$ and called the leading coefficient. The term containing both the leading coefficient and leading monomial is called the leading term, represented by $\text{LT}(f) = \text{LC}(f) \cdot \text{LM}(f)$ [5].

Remark. we say that $f(x)$ is greater than $g(x)$, if $\text{LT}(f - g) \in M(f)$, or the coefficient of monomial $\text{LM}(f - g) \in f$ is greater than that in $g$.

Definition 2.6 (Division algorithm for multivariate divisors) Consider the order " $\succ$ ". Let $F = f_1, \ldots, f_m$ be an ordered set of polynomials according to the order relation " $\succ$ " for the leading terms of polynomials. Thus, any polynomial $f \in K[x_1, \ldots, x_n]$ can be written as $f = a_1 \cdot f_1 + \ldots + a_m \cdot f_m + r$, where $a_i, r \in K[x_1, \ldots, x_n]$ and $r = 0$ or is a polynomials, none of which is divisible by $\text{LM}(f_i) : \forall i = 1 \ldots m$ where $r$ is called the residue of dividing $f$ by $F$ and is denoted by $f \rightarrow_r^F r$ [5]. For one stage division the residue of dividing $f$ by $f_i$ is denoted by $f \rightarrow_{f_i} f - k \cdot f_i$, where $k$ is a monomial. As it can be observed, the order relation plays an important role in the Division algorithm. Using various orders may result in various residues. Subsequently, we shall demonstrate that if $F = f_1, \ldots, f_m$ is a Groebner basis, the residue will be unique for any given order relation.

Lemma 2.7 The division algorithm is a finitely determined relation.

Proof. Any ideal of $K[x_1, \ldots, x_n]$ is finitely generated [5]. It is sufficient to show that the residue of each stage of division algorithm is smaller than dividend; that is:

$$f \rightarrow g \Rightarrow g \prec_{\text{lex}} f$$

where $G = [g_1, \ldots, g_n]$. It is proved that [5]: if

$$f \rightarrow g, LT(h \cdot g_i) \in T(f), 1 \geq i \geq m \Rightarrow \exists s \in T(f) \land \exists a \in K :$$

$$g = f - \frac{a}{LT(g_i)} \cdot s \cdot g_i; LT\left(\frac{a}{LT(g_i)} \cdot s \cdot g_i\right) \in T(f),$$

(4)
where \( f - \frac{n}{LT(g_i)} \cdot s = g_i \). Thus \( g_i \prec_{\text{lex}} f \). So it is proved that the proposed division algorithm is finitely determined. The uniqueness characterization is also derived from the order relation specifications [5].

\[ \]

3 Groebner Basis

The idea of Groebner basis was first proposed by Buchberger [6] to study the membership of a polynomial in the ideal of the polynomial ring. Let an ideal \( I \) be generated by \( G = g_1, \ldots, g_m \), where \( g_i, 1 \leq i \leq m \) is a polynomial. \( G \) is called the Groebner basis for the ideal \( I \), if \[ f \rightarrow G_0 : \forall f \in I(G). \]

**Theorem 3.1** Let \( G = g_1, \ldots, g_m \), where \( g_i, 1 \leq i \leq m \) is a polynomial. \( G \) is a Groebner basis for ideal \( I \), iff [5]: \[ \langle LT(I) \rangle = \langle LT(g_1), \ldots, LT(g_m) \rangle, \] where \( LT(I) = LT(f) : f \in I \) and \( \langle LT(I) \rangle \) denotes the ideal generated by the leading terms of the members in \( I \).

**Proof.** See [5].

According to the Theorem 1, the following definition may be used as Groebner basis equivalent: \[ \langle LT(I) \rangle = \langle LT(g_1), \ldots, LT(g_m) \rangle \Leftrightarrow G \text{ is a Groebner basis for ideal } I. \] Note that the Theorem 1 does not provide a practical criterion for computation of Groebner basis, since verifying the correctness of (5) is impossible in the case of infinite polynomial ring \( K[x_1, \ldots, x_n] \), where an infinite number of operations must be performed. Therefore, we have to seek an equivalent condition comprising finite number of computations. Hence, the following statement from Buchberger comes into effect. The proof is given in [6].

**Theorem 3.2** (Buchberger criterion) \( G \) is a Groebner basis for ideal \( I \), iff for every \( i, j; 1 \leq i < j \leq m \) the following assertion holds [6]:

\[ Spol(g_i, g_j) \triangleq \frac{X^\gamma}{LMg_i} \cdot g_i - \frac{X^\gamma}{LMg_j} \cdot g_j \rightarrow G_0, \] where \( X^\gamma = \text{lcm}(LT(g_i), LT(g_j)) \) (lcm: least common multiple of two terms) and \( Spol(g_i, g_j) \) is called the S-polynomial generated by polynomials \( g_i \) and \( g_j \).

Assume that \( G \) does not fulfill the Buchberger criterion, that is, there exist \( i \) and \( j \) such that the corresponding S-polynomial is not reduced to zero by \( G \). The algorithm below shows how \( G \) can be converted to a Groebner basis for the ideal generated by \( G \), that is, \( I(G) \).
3.1 The Buchberger Algorithm

The following algorithm first introduced by Buchberger in 1965. In this algorithm, the Buchberger criterion is checked for generator polynomials. If the residue of S-polynomials in $G$ does not reduce to zero, this residue is added to $G$ as a new polynomial and creates a new set $G'$. Thus, the ideal generated by $G$, that is, $I(G)$, is the same as that generated by $G'$. This algorithm may be presented as the steps below [6]:

- Input the set of polynomials of $G$ that generates $I(G)$,
- For every two polynomials $g_i$ and $g_j$ of $G$, create the corresponding S-polynomial,

$$Spol(g_i, g_j) = \frac{X^\gamma}{LM(g_i)} \cdot g_i - \frac{X^\gamma}{LM(g_j)} \cdot g_j X^\gamma = \text{lcm}(LT(g_i), LT(g_j))$$  \hspace{1cm} (7)

- Compute the residue of these S-polynomials reduced by $G$,

$$s = Spol(g_i, g_j)^G$$  \hspace{1cm} (8)

- If $s \neq 0$, add $s$ to $G$ as a new polynomial,

$$G' = G \cup s$$  \hspace{1cm} (9)

- Set $G = G'$ and repeat §2 through §4 until $s$ becomes void. Thus $G$ is a Groebner basis.

3.2 Improved Buchberger Algorithm

Any Groebner basis computing algorithm for an ideal can be divided into two major parts. In the first part, called the selection strategy, some of the existing S-polynomials are selected and enter the second part according to the strategy. In this algorithm, this will be done according to the algebraic degree at which the basis is computed. Thus, the degree of each S-polynomial is computed as $|X^\gamma|$ from the two polynomials $g_i$ and $g_j$ which make up the S-polynomial of (7) [5]. The Groebner basis is computed degree by degree, that is, selecting S-polynomials of degree 1, one can compute the corresponding Groebner basis. Next, set the obtained Groebner basis as an input to the algorithm and select those S-polynomials of degree 2 and compute the corresponding Groebner basis. Thus, the S-polynomials are computed with increasing degrees. In this way, this procedure results in a Groebner basis. In the second part, the selected S-polynomials are reduced by the generator polynomials.
3.3 The F4 Algorithm

The F4 algorithm first introduced by Faugere in 1999. The primary objective of the F4 algorithm is to propose a more powerful reduction algorithm. For that purpose the F4 designer reduces simultaneously several polynomials by a list of polynomials using linear algebra techniques which ensure a global view of the process [13]. The first part of the F4 algorithm is the same as the first part of improved Buchberger algorithm. The second part has two main stages. The input of the second part is the set \( F \) of the S-polynomials whose corresponding \(|X^\gamma|\) is equal to the degree for which the Groebner basis is computed [13]. Figure 1 shows the algorithmic scheme of F4. The two stages of the second part are as follows.

- Symbolic Preprocessing
  - \( \text{Done} := LT(F) \)
  - While \( T(F) \neq \text{Done} \) select \( m \in T(F) \) \( \text{Done} \) and \( \text{Done} := \text{Done} \cup m \).
  - If \( m \) is top reducible \( G \) and \( m = m' \cdot HT(f) \) for some \( f \in \text{Gandsomem}' \in T \)
  - \( F = F \cup m' \cdot f \)

- As it can be seen the set \( F \) includes the S-polynomials in the critical pair formation and some polynomials (reductors) which are produced in the first stage. By critical pair we mean the pair defined as \((X^\gamma \cdot \text{LM}(g_i) \cdot g_i, X^\gamma \cdot \text{LM}(g_j) \cdot g_j)\). In this stage the elements of \( F \) reduce each other in a row echelon matrix. Finally, if there were some polynomials whose leading terms are not in the set \( LT(F) \), then add these polynomials to \( G \).

4 New Algorithm

In the computation process of Groebner basis for an ideal, any Basis Computing Algorithm may be improved by changing the order [7]. Therefore, it is crucial to figure out the optimal order in the intermediate stages of computing the basis. Such algorithms can be developed heuristically [7]. Bearing this in mind, the proposed algorithm attempts to change the conditions of the problem from finding the optimal order to offering a new division algorithm, apart from the ordering used. This leads to a new improved Buchberger algorithm. Our objective is to reduce the degree and number of monomials within the polynomials of the basis. For practical purposes, we desire the polynomials representing a Groebner basis for a system of multivariate polynomial equations, that is, the polynomials generating the ideal, consist of monomials with
smaller degree regarding the presumed order. In solving a system of multivariate polynomial equations, the less the number of equations resulting from a Groebner basis, the less computational complexity of finding the solution. The proposed algorithm for computing a Groebner basis, described in the next part, reduces the dimension of the Groebner basis, as well.

### 4.1 The Proposed Division Algorithm

We start this section with an example. Assume that in the Basis Computing Algorithm within the reduction stage, we need to reduce the following polynomial by the set of polynomials $g_i, g_j, g_k$.

\[
s = x_1x_2x_4x_5 + x_1x_3x_4 + x_1x_5 + x_2x_4,
\]

where the polynomials $g_i, g_j$ and $g_k$ are arranged in the lexicographical order:

Let $x_5 \prec_{lex} x_4 \prec_{lex} x_3 \prec_{lex} x_2 \prec_{lex} x_1$ and $K = GF(2)$.

\[
\begin{align*}
g_i &= x_1x_2x_4 + x_1x_3 + x_2x_3 \\
g_j &= x_1x_2x_5 + x_1x_3 \\
g_k &= x_1x_2.
\end{align*}
\]

Note that in division algorithm we have two kinds of orders, one between the monomials of a polynomial and the other between divisors. In this article,
the authors focus on the order between devisors and try to change this order in division algorithm exposed in definition 6. In the Buchberger algorithm the stages of division algorithm are performed according to the presumed order. Thus, considering the fact that \( g_i \) is the largest polynomial, the S-polynomial is reduced by \( g_j \), that is,

\[
s - x_5 \cdot g_i = r_1 = x_1 x_3 x_4 + x_1 x_3 x_5 + x_1 x_5 + x_2 x_3 x_5 + x_2 x_4 \tag{12}\]

Now, let us consider another order in this division algorithm. For example, let \( g_j \) be the divisor; as a result, the residue of the division algorithm will be different from that of the previous one, where \( g_i \) is the divisor. Therefore, the residue of the division algorithm is:

\[
s - x_4 \cdot g_j = r_2 = x_1 x_5 + x_2 x_4 \tag{13}\]

The ultimate goal is finding a basis with the least possible number of monomials and also the least possible degree of polynomials with respect to the presumed ordering, hence resulting in the least possible dimension of the computed Groebner basis. Consider the residues of the above divisions for all three choices of divisor:

\[
\begin{align*}
  s - x_5 \cdot g_i &= x_1 x_3 x_4 + x_1 x_3 x_5 + x_1 x_5 + x_2 x_3 x_5 + x_2 x_4 \\
  s - x_4 \cdot g_j &= x_1 x_5 + x_2 x_4 \\
  s - x_4 x_5 \cdot g_k &= x_1 x_3 x_4 + x_1 x_5 + x_2 x_4. \tag{14}
\end{align*}
\]

As it is observed, despite the fact that according to the presumed order, \( g_j \) is smaller than \( g_i \), the residue of dividing the given S-polynomial by \( g_j \) yields a smaller residue (i.e. a residue with smaller leading term), smaller degree, and also smaller number of monomials than dividing it by \( g_i \). Therefore, \( g_j \) fulfills our goal rather than \( g_i \) and \( g_k \) and so it is the selected divisor in the approach used in this stage. In this way the proposed division algorithm selects a divisor in each step. According to the abovementioned fact, the major difference between the proposed division algorithm and the algorithm described in definition 6 is the following. In each stage of the former, we choose the divisor whose corresponding residue yields the smallest leading term (according to the presumed order), in contrast to the latter case in which the largest divisor is selected. Thus we render the division in each stage of the algorithm independent from the order between divisor polynomials, and select the divisor according to the proposed division algorithm.

Remark. As explained above, the sole difference between the definition 6 and the proposed algorithm is the choice of reductors in each step. Since the reductors do not play any role in proving Lemma 1 (i.e. the choice of \( h, g_i, h \cdot g_i \) and consequently have no role in the proof of Lemma 1).
Remark. Remark 1 shows that the proposed division algorithm for computing Groebner basis is finite. Since in each step of Groebner basis procedure a few polynomials are added to the previous set, we shall have an ascending chain of the ideal of leading terms of polynomials; considering the fact that $K[x_1, \ldots, x_n]$ is Noetherian, the chain of its ideals is finite. Therefore, the algorithm for computing Groebner basis based on the proposed division algorithm is a finitely determined algorithm.

Remark. The authors have shown that the proposed division algorithm results in a unique residue for a specified order relation. Hence it is finitely determined.

So, replacing the division algorithm, used in the improved Buchberger’s, with the proposed one, improves the previous Groebner basis computation algorithms.

5 The Courtois Toy Cipher (CTC)

CTC is designed to be vulnerable against algebraic attack while any other attack is infeasible on this cipher [9]. However, this cipher is not resistant against linear cryptanalysis [10]. The notation $CTC_{3,b,n_r}$ denotes a CTC equations system with $B = b$ and $N_r = n_r$, where $B$ denotes one third of the block size and $N_r$ denotes the number of rounds as explained below [9].

5.1 A brief description of CTC

This cipher is an iterated block cipher which operates on block sizes of multiple 3. The block size is $B \cdot s$ where $s = 3$ and $B$ can be chosen arbitrarily. The number of rounds is denoted by $N_r$. Each round consists of parallel applications of $B$ S-boxes, a linear diffusion layer, and a round-key addition. Also, a round-key $k_0$ is added to the plaintext block before the first round. The plaintext bits $p_0 \cdots p_{B \cdot s - 1}$ are identified with $z_{0_0} \cdots z_{0_{B \cdot s - 1}}$ and the ciphertext bits $c_0 \cdots c_{B \cdot s - 1}$ are identified with $x_{N_r + 1_0} \cdots x_{N_r + 1_{B \cdot s - 1}}$ to have a uniform notation.

5.1.1 The S-box

The S-box is defined over $F_{2^3}$ as the non-linear random permutation $7, 6, 0, 4, 2, 5, 1, 3$. The transformation from $(F_2)^3$ to $F_{2^3}$ is the "natural" mapping $x = 4 \cdot x_3 + 2 \cdot x_2 + x_1$ and $y = 4 \cdot y_3 + 2 \cdot y_2 + y_1$ where $x$ and $y$ are the input and the output of the S-box respectively and $x_1, x_2, x_3$ and $y_1, y_2, y_3$ are the input and output bits respectively. This S-box gives $r = 14$ quadratic equations in $t = 22$ terms over $F_2$ [9]. This S-box has also been used in [11] by Courtois to describe a toy cipher. Later Biryukov and Canniere in [12] described a method
to construct a basis for a quadratic equation system from a given S-box using CTC S-box as an example.

5.1.2 The Diffusion Layer
The diffusion layer is defined as:

\[ z_{i,(257 \mod B)} = y_{i,0}, i = 1, \ldots, N_r \]
\[ z_{i,(j \cdot 1987 + 257 \mod B \cdot s)} = y_{i,j} + y_{i,(j+137 \mod B \cdot s)}, j \neq 0, i = 1, \ldots, N_r, \quad (15) \]

where \( y_{i,j} \) and \( z_{i,j} \) represent input bits and output bits respectively.

5.1.3 Key Addition
A bit-wise key addition is performed as follows.

\[ x_{i+1,j} = z_{i,j} + k_{i,j}, i = 0, \ldots, N_r, j = 0, \ldots, B \cdot s - 1, \quad (16) \]

where \( z_{i,j} \) represents output bits of the previous diffusion layer, \( x_{i+1,j} \) the input bits of the next round and \( k_{i,j} \) the bits of the current round-key. The round keys are generated by a key schedule as follows.

\[ k_{i,j} = k_{0,(j+i \mod B \cdot s}) \quad (17) \]

5.2 Extracting the algebraic equations of Courtois Toy Cipher
S-boxes are the nonlinear parts of every block cipher which play an important role in nonlinearity of the equations representing the block cipher. The equations given in [9] representing the CTC S-box are written as polynomials of the S-box input and output. The number of these equations is \( r = 14 \). As discussed in Section 4, finding a Groebner basis for a system of multivariate polynomial equations, the dimension and also the number of monomials in the computed Groebner basis are especially important. The dimension of the Groebner basis depends on that of the input of the basis computing algorithm. That is, the smaller the dimension of the input of the algorithm, the higher the probability of having a basis with lower dimension. Thus, it is crucial to find the normal algebraic form for the CTC S-box with minimal number of monomials and algebraic degree. As an example, we compute a Groebner basis for the CTC S-box. First, we obtain a Groebner basis for the system of equations in [9]. In the following it can be shown that the system of equations in [9] can be replaced by the system of equations composed of its first, second and fifth equation:

\[ y_1 = x_1 x_2 + x_3 + x_2 + x_1 + 1 \]
\[ y_2 = x_1 x_3 + x_2 + 1 \]
\[ y_3 = x_1 x_2 + x_1 x_3 + x_2 x_3 + x_3 + x_2 + 1 \quad (18) \]
Note that these three equations can be represented explicitly on the contrary to the other equations in [9]. The Groebner basis computed for the system of equations (18) has the dimension smaller than that of the system of equations in [9]. It is noteworthy that the system of equations (18) has unique solution, similar to the equations in [9]. Therefore, we utilize the system (18) to extract the system of equations representing the CTC. In the next section we apply the proposed algorithm along with the improved Buchberger and F4 to the extracted equations representing the CTC to find a Groebner basis and compare these three algorithms from the efficiency point of view.

Table 1: Comparing the algorithms for computing the Groebner basis for $x = 44$ and $k = 42$

<table>
<thead>
<tr>
<th>Cipher</th>
<th>$CTC_{3,2,1}$</th>
<th>$CTC_{3,2,2}$</th>
<th>$CTC_{3,2,3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCB</td>
<td>13 14 13</td>
<td>19 21 18</td>
<td>51 54 44</td>
</tr>
<tr>
<td>DC</td>
<td>2.07 1.64 2.60</td>
<td>12.79 3.24 10.73</td>
<td>123.75 21.54 62.96</td>
</tr>
<tr>
<td>NA</td>
<td>21352 25132 37525</td>
<td>236055 266188 346698</td>
<td>2117538 3211244 2875527</td>
</tr>
<tr>
<td>NMU</td>
<td>16366 12670 27503</td>
<td>169120 57932 207648</td>
<td>1365833 395990 1197539</td>
</tr>
<tr>
<td>NMO</td>
<td>52 53 50</td>
<td>159 190 162</td>
<td>624 799 630</td>
</tr>
</tbody>
</table>

DCB: Dimension of Computed Basis, DC: Duration of Computation (Sec), NA: Number of Additions, NMU: Number of Multiplications, NMO: Number of Monomials.

5.3 Comparison and Discussion

In this section, we consider the system of equations representing the 1-round, 2-round, and 3-round CTC cipher with $B = 2$, key $k = 42$, and $x = 44$ as the input for the algorithms of computing the Groebner basis. The results of the comparison are summarized in table 1:

As depicted in Table 1, the new algorithm is superior to both improved Buchberger and F4 in terms of the dimension of the computed basis and comparable number of monomials, in all three cases of 1, 2, and 3 round algorithm. Considering the Groebner bases computed for an input, we derive the key of the CTC cipher for one, two and three rounds.

6 Conclusion

In this paper, we proposed a new computing Groebner basis algorithm for a system of multivariate polynomial equations, describing a cryptosystem. Be-
sides it is proved that the new division algorithm is finitely determined, resulting in a unique residue for a specified order relation. The proposed Groebner basis algorithm based on the new division algorithm has two advantages over the F4 and improved Buchberger’s, namely less number of monomials within the polynomials of the basis and smaller dimension of the computed Groebner basis. Taking the abovementioned facts into account, we conclude that systematizing the change of order relation during division algorithm plays an important role in computing Groebner basis.

References


Received: November, 2009