IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 5, May 2016

Comparative Study of Algebraic Attacks

Ahmed A. Abdel-Hafez[1], Reda-Elbarkouky[2], Wageda_ Hafez[3]

Department of Communication, Military Technical College, Egypt[1]

Department of Mathematics, Ain Shams Engineering Faculty, Ain Shams University, Egypt[2]

Department of Mathematics, Benha Engineering Faculty, Benha University, Egypt[3]

Abstract: Cryptographic schemes have an algebraic structure and can be described as multivariate polynomial equations. Even though algebra is the default tool in the cryptanalysis of asymmetric cryptosystems, there has been recently an increase in interest in the use of algebraic cryptanalysis techniques in the analysis of symmetric cryptosystems. The basic idea behind the algebraic attack is to express the whole cryptosystem as a large system of multivariate polynomial equations, then considers methods for solving the system to recover the key. Solving multivariate polynomial systems is a typical problem studied in Algebraic Geometry and Computational Algebra. Computing Grobner basis is the best well known method to solve this problem. Finding grobner bases is a difficult task which requires lots of computational resources. This paper discusses and explains in depth different algorithms to compute grobner bases using examples. This paper also, compares these algorithms from the point of views of accuracy and efficiency (the required resources: time and effort) to get the accurate results. Finally, the worthiness of these algorithms to be applied to cryptanalysis has been discussed.

Keywords: Grobner bases, Cryptanalysis ,Algebraic attacks, F4 Algorithm, F5 Algorithm.

I. INTRODUCTION

In his seminal paper, Claude Shannon asked the question how we can ever be sure that a cryptosystem, which is not ideal, will require a large amount of work to break with every method of analysis. [1]," Shannon suggested two approaches to that problem:
1) We can study the possible methods of solution available to the cryptanalyst and attempt to describe them in sufficiently general terms to cover any of the methods he might use. We then construct our system to resist this general method of solution. 2) We may construct our cipher in such a way that breaking it is equivalent to (or requires at some point in the process) the solution of some problem known to be laborious. Thus, if we could show that solving a certain system requires at least as much work as solving a system of simultaneous equations in a large number of unknowns, of a complex type, then we would have a lower bound of sorts for the work characteristic, the structure of this paper as follows: firstly; cryptanalysis types is explained in section (2), then detailed explanation of Algebraic attacks in section (3), Grobner bases will be discussed in section (4) while sections (5,6) presents and analysis F4 and F5 , finally conclusion and future work is presented in section (7).

II. CRYPTANALYSIS TYPES

Cryptology [2, 3] is an art and science of hidden or secret writing. It has two main areas: cryptography and cryptanalysis [4, 5]. Cryptography is basically related with converting data to make them secure and immune to attacks where cryptanalysis is related with breaking of codes [2]. There are two categories of cryptography.

A. Symmetric key cryptography
B. Asymmetric key cryptography

In symmetric key, there is only single key which is used by sender for encryption and receiver for decryption. In this type the key is shared between both the parties [4]. In asymmetric key, there are two keys: a private key and a public key. Private Key is kept by receiver for decryption and public key is announced to public and is used for encryption of the data [3]. The main goal of a cryptanalyst is to attack the cryptosystem to obtain maximum information about the plaintext (original data). Classification of attacks can be done on following basis [4]:

A. Amount of Information Available to Attacker
The main objective of attacking is to access the encryption key in place of simply decrypt the data. Attacks can be classified on the basis of information available to attacker. (a) Cipher text only. (b) Known Plain text. (c) Chosen Cipher text. (d) Chosen Plain text. (e) Adaptive Chosen Plain text. (f) Adaptive Chosen Cipher text. (g) Related Key Attack.

B. Computational Resources Required
Attacks can also be classified on the basis of resources they require. Those resources are

Time: the number of computation steps (like encryption) that must be performed.
Memory: the amount of memory required to perform the task.
Data: the amount of required plain text or cipher text. Actually it is very difficult to find out all these resources very precisely, especially when the attack isn't practical to actually implement for testing. But academic cryptanalyst tend to provide at least estimated order of magnitude of their attacks difficulty.

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 5, May 2016

1- Cryptanalysis of Asymmetric Cipher

Asymmetric cryptography is a type which relies on two keys, one private key for decryption and one public key for encryption. Such kind of cipher relies on the hard mathematical problem for their security. So the main point of attack is to develop methods to solve such problems. The security of Asymmetric cryptography depends on mathematical questions in a way that symmetric cryptography doesn't, conversely links to wider area of mathematical research in a new way. Asymmetric techniques are designed based on the difficulty of solving of various hard mathematical problems. In case any improved algorithm is found to solve the problem then system is weakened. For example the security of Diffie-Hellman key exchange depends on calculating the discrete logarithm [2]. While the security of RSA protocol depends on the difficulty of integer factorization of a large composite number. A breakthrough in factoring would impact security of RSA. Another main feature of asymmetric over symmetric cipher is that cryptanalyst has an opportunity to make use of knowledge obtained from public key [3].

2- **Cryptanalysis of Symmetric Cipher** [4]

There are various types of well-known attacks on symmetric cipher; as given below:

Boomerang Attack: This is a method of cryptanalysis of block cipher based on differential cryptanalysis .This attack provides various avenues of attack on various cipher which are deemed safe from differential cryptanalysis, **Brute Force Attack** or exhaustive key search is a type of strategy which can be applied on any type of encrypted data. In this type of attack all possible keys are tried systematically until correct key is found. This method is used when any other weakness is not useful; **Davies' Attack** This attack is dedicated statistical cryptanalysis method for attacking Data Encryption Standard (DES). This attack was originally created by Donald Davies in 1987[?], **Differential Cryptanalysis** This attack is a chosen plaintext attack where some relationship is found out between the cipher texts produced by two related plaintext. It focuses on the statistical analysis of two inputs and two outputs of cryptographic algorithm**, Integral cryptanalysis**; this attack is applicable on block cipher based on Substitution-Permutation Networks (SPN). Unlike differential cryptanalysis, it uses sets or even multiset's of chosen plaintext of which part is held constant and other part varies with all possibilities It is commonly known as Square attack, **Linear Cryptanalysis**: this is a known plaintext attack that requires access to large amount of plaintext and cipher text pairs which are encrypted with unknown keys. It focuses on statistical analysis against one round of decryption on large number of cipher text. The attacker decrypts each cipher text using all possible sub keys for one round of encryption and studies the resulting intermediate cipher text to seek the least random result**.**

All the previous mentioned attacks are statistical in nature. On the other hand, algebraic attacks depend on the structural nature of the cryptosystems. Notably, during the last two decades algebraic cryptanalysis grabbed a lot of Attention. This interest shows up because Rijndael (the AES candidate) has a rich algebraic structure. The following section mentioned cryptanalysis from point of view **Algebraic attacks**.

III. ALGEBRAIC ATTACKS

Algebraic cryptanalysis is a general tool which permits one to breach the security of a wide range of cryptographic schemes. Algebraic techniques have been successfully applied against a number of multivariate schemes and stream ciphers. Yet, their feasibility against block ciphers remains the source of much speculation. The goal of algebraic cryptanalysis is to break cryptosystems by using mathematical tools coming from symbolic computation and modern algebra. More precisely, an algebraic attack can be decomposed in two steps: first the cryptosystem and its specifics have to be converted into a set of multivariate polynomial equations, then the solutions of the obtained polynomial system have to be computed. The security of a cryptographic primitive thus strongly relies on the difficulty of solving the associated polynomial system. These attacks have been proven to be very efficient for both public key or symmetric cryptosystems and stream ciphers. In this paper, we focus on the polynomial system solving part. It is well known that this problem is very difficult (NP-hard in general). However, for many instances coming from algebraic attacks, the resolution is easier than in the worst-case scenario. Grobner bases, first introduced in [6], are a fundamental tool for tackling this problem the basic idea behind the algebraic attack is to set up a system of equations including key bits and output bits and then to solve this system to recover key or key stream information [7]. A system of linear equations may be solved by Gaussian elimination method or any other known method. However, a cipher may contain a non-linear part. In this case the equations will be non-linear. If the system of equations is clearly defined then the equation set can be solved using techniques such as linearization, or other methods such as Gröbner bases. However, since the complexity of solving such equations grows exponentially with the degree of the equations, the cryptanalysis may try to identity low degree equations we do some preliminaries then grobner base algorithm is presented which provides us a platform to analyze and solve common problems.

IV. GROBNER BASES

One way to solve a system of polynomial equations is to construct a new system of polynomial equations with the same solutions as the initial one, but with a simpler structure and then solve this "simpler" system .This method is based on polynomial ideal theory and multivariate polynomial division and generates special bases of these ideals, called Grobner bases. The algorithm is based on the construction of S-polynomials and on polynomial division of these S-polynomials [8, 9]. Multivariate polynomial division requires a monomial ordering and different orderings can give rise to radically

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 5, May 2016

different Grobner bases .For some problems and some orderings, especially lexicographic ordering, the construction of Grobner bases using this standard Buchberger's algorithm or its variations [10, 11, 12] is very time-consuming and sometimes even does not finish in reasonable time.

Definition (1). (Ideal)[10] The ideal defined by a set of polynomials $F = \{f_1, \dots, f_m / f_i \in \mathbb{F}[x_1 \dots, x_n]\}$ is the set of all polynomials that can be generated as polynomial combinations of the initial polynomials $f_1, \dots, f_m$

$$I = \{\sum_{i=1}^{m} f_i h_i : h_i \in \mathbb{C}[x_1 \dots, x_n]\}$$

Where h_i are arbitrary polynomials from $\mathbb{F}[x_1 \dots, x_n]$.

Definition (2). [10] (Lexicographic ordering) let x^α and x^β be some monomials. we say $x^\alpha >_{lex} x^\beta$ if, in the difference $\alpha - \beta \in \mathbb{Z}^n$, the left most nonzero entry is positive.

Definition (3). [12] (Graded Reverse lexicographic ordering)) let x^α and x^β be some monomials. we say $x^\alpha >_{grevlex} x^\beta$ if $\sum_{i=1}^{n} \alpha_i > \sum_{i=1}^{n} \beta_i$ or if $\sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} \beta_i$ and the difference $\alpha - \beta \in \mathbb{Z}^n$, the right most nonzero entry is negative.

Definition (4). [12](S-Polynomial) let $f, g \in \mathbb{F}[x_1 \dots, x_n]$ be non zero polynomials and $>$
Some fixed monomial ordering on $\mathbb{F}[x_1 \dots, x_n]$.
The S-Polynomial of f and g, denoted $S(g_p, g_q)$, is the polynomial

$$S(g_p, g_q) = \frac{LCM(LM(g_p), LM(g_q))}{LT(g_p)} g_p - \frac{LCM(LM(g_p), LM(g_q))}{LT(g_q)} g_q$$

Where $LCM(LM(g_p), LM(g_q))$ is the least common multiple of the monomial $LM(g_p)$ and $LM(g_q)$. The above mentioned definition indicates that S-polynomials are cross product of leading terms and are constructed to cancel leading terms. The leading terms of the two components of $S(g_p, g_q)$ are equal and therefore, cancel each other.

Example. Let G=$\{g_1, g_2\}$ where $g_1 = xy^2z - xyz$ and $g_2 = x^2yz - z^2$.These polynomial are ordered with respect to Lex order .$LM(g_1) = xy^2z$, $LM(g_2) = x^2yz$ so $LCM(LM(g_1), LM(g_2)) = x^2y^2z$. *then* $S(g_1, g_2) = \frac{x^2y^2z}{xy^2z} g_1 - \frac{x^2y^2z}{x^2yz} g_2$
$= -x^2yz + yz^2$.

Theorem 1. (Buchberger's criterion)
A finite set of polynomials $G = \{g_1, \dots, g_l\}, G \subset I$ is a Grobner basis of I if and only if $\overline{S(g_p, g_q)^G} = 0$ for all pairs $i, j \in 1, \dots, t, i \neq j$.
Proof. The proof of this theorem can be found in [8]
The simplest version of the Buchberger's algorithm for computing a Grobner basis of a given ideal is based on this criterion.

Algorithm 1. Buchburger's[8]
Input:$F = \{f_1, \dots, f_m\}$
Output: A Grobner basis $G = \{g_1, \dots, g_l\}$ *for* $I = \langle f_1, \dots, f_m \rangle$, *with* $F \subset G$.
1:G:=F
2:repeat
3: $G' := G$
4: for each pair (p, q) such that $g_p, g_q \in G'$ *and* $p \neq q$ *do*
5: S:= $\overline{S(g_p, g_q)^G}$
6: *if* $S \neq 0$ *then*
7: $G = G \cup \{S\}$
8: end if
9: end for
10: until $G := G'$

This algorithm can be analyzed in simple form within the following example
F ={2xyz+3,3xz+y, $z^3 + 1$}this is the original system the following steps shows how to calculate grobner basas.

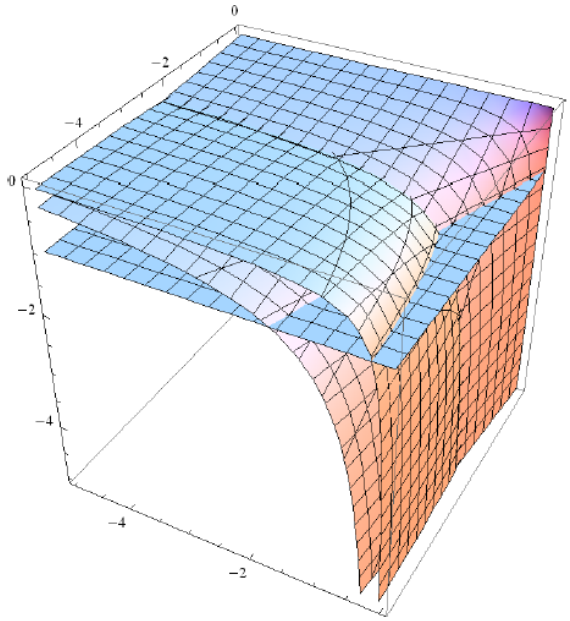
I	g_1, g_2	S	$\overline{S^G}$	Added critical pairs
0				{2xyz+3,3xz+y}, {2xyz+3,$z^3 + 1$}, {3xz+y, $z^3 + 1$}
1	2xyz+3,3 xz+y	$3y^2 + 4$	$3y^2 +$ 4	{2xyz+3, $3y^2$ +4}, {3xz+y, $3y^2$ +4}, {$z^3 + 1$, $3y^2$ +4}
2	2xyz+3, z^3+1	3xy+ $3z^2$	3xy+ $3z^2$	{2xyz+3, 3xy+$3z^2$} ,{3xz+y, 3xy+$3z^2$}, {$z^3 + 1$, 3xy+$3z^2$}, { $3y^2$ +4, 3xy+$3z^2$}
3	3xz+y, z^3+1	yz^2+ 2x	yz^2+ 2x	{2xyz+3, yz^2+2x}, {3xy+$3z^2$, yz^2+2x}, {z^3+1, yz^2+2x}, { $3y^2$ +4, yz^2+2x}, {3xy+$3z^2$, yz^2+2x}
4	2xyz+3, $3y^2$ +4	2xz-y	0	
. . .				
10	$3y^2$ +4, 3xy+$3z^2$	yz^2+ 2x	o	
11	2xyz+3, yz^2+2x	x^2+3 z	x^2+3 z	{2xyz+3, x^2+3z}, { 3xz+y, x^2+3z}, {$z^3 + 1$, x^2+3z}, { $3y^2$ +4,x^2+3z}, {3xy+$3z^2$, x^2+3z}, {{$y\,z^2$+2x, x^2+3z}
12	3xy+$3z^2$, yz^2+2x	yz^2- x^2	0	
. .	. . .	. . .	0	
21	$y\,z^2$+2x, x^2+3z	2 yz^3+ $2x^3$	0	

Now the ideal become {$3y^2$ +4, 3xy+$3z^2$, $y\,z^2$+2x, x^2+3z, 2xyz+3,z^3+1, 3xz+y }

DOI 10.17148/IARJSET.2016.3519

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 5, May 2016

This example shows how many steps required to find ideal ,critical pairs is considered as waiting list to evaluate its S polynomial then apply polynomial division; As soon as $\overline{S^G} = 0$ for all polynomials then ideal is obtained . For demonstration only a system of three equations have been used to show how the grobner base work in Fig (1) shows the original system while Fig(2) shows 3 polynomial of the ideal only



Fig(1) The original system
{2xyz+3,3xz+y ,$z^3 + 1$},

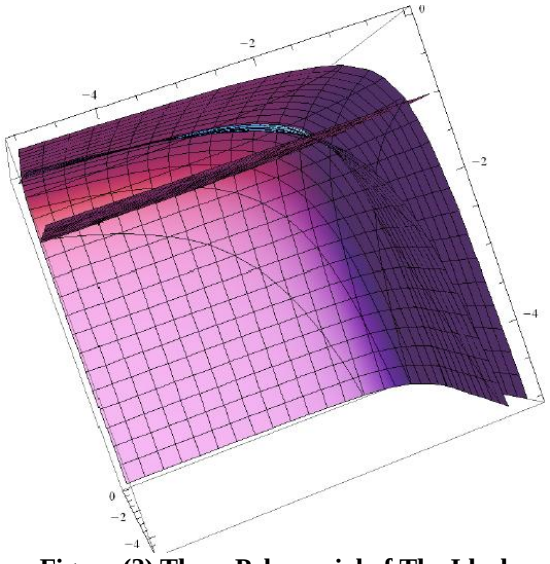


Figure (2) Three Polynomial of The Ideal

Analysis of Buchberger's Algorithm [8,9,10]
From this example the algorithm become clear enough to understand, although algorithm is very simple to describe and implement but computing of S every time is very waste of time and difficult to obtain for big system. or some problems and some orderings, especially lexicographic ordering, the construction of Grobner bases using this standard Buchberger's algorithm or its variations is very time-consuming and sometimes even does not finish in reasonable time. Therefore, many improvements of Buchberger's algorithm have been proposed in recent years. They are mostly divided into two groups.

V. F4 ALGORITHM

The first group of improvements is dealing with so-called strategies of selection. During the Grobner basis computations, several choices can be made. We can select an S-pair and also polynomials for reduction. The well-known algorithm, with an improved selection strategy, is the F4 algorithm developed by Faug`ere [13], though this algorithm is slightly different from standard Buchberger's algorithm. The F4 algorithm not only improves the selection strategy but it also replaces multiple polynomial divisions by row reduction (Gauss-Jordan elimination) of a single sparse matrix. In this way the F4 algorithm transforms computations with polynomials to linear algebra computations. The main idea of Faugère's F4 algorithm is to use linear algebra to simultaneously reduce a large number of pairs. F4 works with critical pairs instead of S-polynomials: the critical pair $c(f_1, f_2)$ of two polynomials and is defined as the tuple (lcm, u_1, f_1, u_2, f_2) where lcm=LCM(LM(f_1), LM(f_2)), , the least common multiple of LM(f_1) and LM(f_2), and$u_i = \frac{lcm}{LT(f_i)}$.

At each iteration step, a Macaulay-style matrix is constructed, whose columns correspond to monomials and rows to polynomials. This matrix contains the products $(u_i f_i)$ coming from the selected critical pairs (classically, all pairs with the lowest total degree lcm, but other selection strategies are possible) and also all polynomials involved in their reductions, which are determined during the Symbolic preprocessing phase. By computing the reduced row echelon form of this matrix, we obtain the reduced S-polynomials of all pairs considered. This algorithm, combined with an efficient implementation of linear algebra, yields very good results. A complete description of this F4 is presented below (algorithm2, 3, 4 ,5). For a more detailed discussion we refer the reader to [14].

Algorithm 2 F4[14]
Input :$F = (f_1, f_2, \dots, f_m) \in \mathcal{R}^m$
Output : The Grobner bases of F .
Initialization : $G := \emptyset \;\; and \; P := \emptyset \;\; and \; d := 0$

1. While $F \neq \emptyset \;\; do$
2. $f := first(F)$
3. $F := F \setminus \{f\}$
4. $(G, P) := Update(G, P, f)$
5. $While \;\; P \neq \emptyset \;\; do$
6. $d := d + 1$
7. $P_d := Select(P)$
8. $P := P \setminus P_d$
9. $\left(\tilde{F}_d^+, F_d\right) := Reduction(P_d, G, (F_i)_{d=1,\dots,(d-1)})$
10. $for \; h \in \tilde{F}_d^+ \;\; do$
11. $(G, P) := Update(G, P, h)$
12. Return G

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 5, May 2016

Algorithm 3: Reduction[14]
Input: P_d a finite subset of selected pairs, G a finite subset of $\mathcal{R}[x]$,$\mathbb{F} = (F_k)_{k=1,\dots,d}$, where F_k is a finite subset of $\mathcal{R}[x]$.

Output : two finite subsets of $\mathcal{R}[x]$.
1. F:= Symbolic Preprocessing($P_d G, \mathbb{F}$)
2. $\widetilde{F}:=$
Reduction to Row Echelon Form of F *w.r.t.* $<$
3. $\tilde{F}^+ := \{f \in F \setminus HT(f) \notin HT(F)\}$
4. Return($\tilde{F}^+$, F)

Algorithm 4: Symbolic Preprocessing[14]
Input: P_d a finite subset of selected pairs, G a finite subset of $\mathcal{R}[x]$,$\mathbb{F} = (F_k)_{k=1,\dots,d}$, where F_k is a finite subset of $\mathcal{R}[x]$.

Output : two finite subsets of $\mathcal{R}[x]$.
1. F=$\cup_{c(f_1,f_2)\in P_d} \begin{Bmatrix} mult\big(Simplify(u_1, f_1, \mathbb{F})\big), \\ mult\big(Simplify(u_2, f_2, \mathbb{F})\big) \end{Bmatrix}$
2. Done :=HT(F)
3. While T(F)≠ *Done do*
4. m an element of T(F) \ Done
5. Done := Done ∪ {m}
6. if m top reducible module G then
7. m =$m' * H\acute{T}(f)$for some$f \in G$ and some $m' \in T$
8. F:=F∪ $\{mult\Big(Simplify(m', f, \mathbb{F})\Big)\}$
9. Return F

Algorithm 5 simplify[14]
Input: $t \in T$ *a term*
$f \in R[X]$ *apolynomial*
$\mathbb{F} = (F_k)_{k=1,\dots,d}$, where F_k is a finite subset $\mathcal{R}[x]$.

Output: a non evaluated product, i.e. an element of T×R[x]
1. for $u \in$ list of divisors of t do
2. if $\exists j(1 \le j < d)$*such that* $(u * f) \in F_j$ *then*
3. $\widetilde{F_j}$ is the row echelon form of F_j w.r.t. $<$
4. There exists a (unique) $p \in \tilde{F}_j^+$ such that HT(p)=HT(u*p)
5. If $u \neq t$ *then*
6. Return Simplify$\left(\frac{t}{u}, p, \mathbb{F}\right)$
7. Else
8. Return(1,p)
9. Return (t, f)

Example
If $G = [(107xy + y^2 + 29, x^2 + 80xy + 114)]$
With respect to Lex order under $\mathbb{F}_{127}$ critical pairs in the main loap are :
$P_1 = [\big((x, 107xy + y^2 + 29), (y, x^2 + 80xy + 114)\big)]$ as the intermediate basis .
$L_1 = [(y, x^2 + 80xy + 114), (x, 107xy + y^2 + 29)]$
Sympolic processing returns
$[107xy^2 + y^3 + 29y, x^2y + 80xy^2 + 114y, 107x^2y + xy^2 + 29x]$

Or in matrix form

$$F = A_F.v_F = \begin{pmatrix} 0 & 107 & 0 & 1 & 29 \\ 1 & 80 & 0 & 0 & 114 \\ 107 & 1 & 29 & 0 & 0 \end{pmatrix} \begin{pmatrix} x^2y \\ xy^2 \\ x \\ y^3 \\ y \end{pmatrix},$$

The row echelon form of F is

$$\tilde{F} = \widetilde{A_F}.v_F = \begin{pmatrix} 1 & 0 & 0 & 4 & 103 \\ 0 & 1 & 0 & 19 & 43 \\ 0 & 0 & 1 & 24 & 17 \end{pmatrix} \begin{pmatrix} x^2y \\ xy^2 \\ x \\ y^3 \\ y \end{pmatrix}$$

Those polynomial whose leading monomial are not in F are $\tilde{F}$=$[x + 24y^3 + 17y$
P=P_2 = $[((y, x + 24y^3 + 17y),(1, 107xy + y^2 + 29)),((x, x + 24y^3 + 17y), (1, x^2 + 80xy + 114))]$
G = $[x + 24y^3 + 17y]$
L_2 = $[(1, , 107xy + y^2 + 29),(1, x^2 + 80xy + 114),(y, x + 24y^3 + 17y),(x, x + 24y^3 + 17y)]$

F = $17y^2 + 24y^4 + xy, 107xyy^2 29, 17y^4 + 24y^6 + xy^3, 114 + 80xy + x^2, 17xy + 24xy^3 + x^2]$
$\tilde{F}$ = $[67 + 74y^2 + y^4, 122 + 52y^2 + y^6, 43 + 19y^2 + xy, 124 + 34y^2 + xy^3, 103 + 4y^2 + x^2]$
$\tilde{F}^+$ = $[67 + 74y^2 + y^4, 122 + 52y^2 + y^6]$
The third iteration
P=P_3 = $[\big((y^2, 67 + 74y^2 + y^4), (1, 122 + 52y2 + y6\big)]$

G= $[67 + 74y^2 + y^4, x + 24y^3 + 17y]$
L_3 = $[(1, 122 + 52y^2 + y^6), (y^2, 67 + 74y^2 + y^4),$
F=$[67 + 74y^2 + y^4, 122 + 52y^2 + y^6, 67y^2 + 74y^4 + y^6]$
$\tilde{F}$ = $[67 + 74y^2 + y^4, 122 + 52y^2 + y^6]$
$\tilde{F}^+$ = ∅

After finishing 3 iterations this work can be summarize in two shown figures

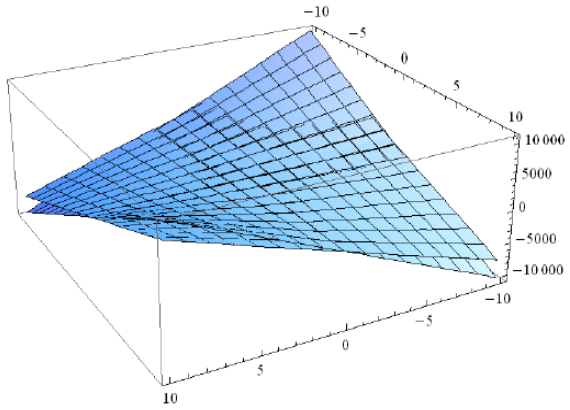
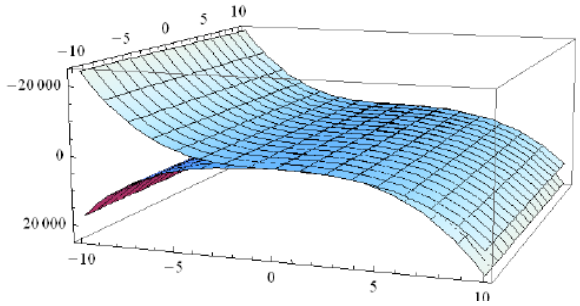


Fig (3) The original system $107xy + y^2 + 29, x^2 + 80xy + 114$

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 5, May 2016



Fig(4) The ideal obtained by F4
$67 + 74y^2 + y^4, 122 + 52y^2 + y^6$

The two figure shown have the same solution .Although there were two variable but there's solution can't be appear because solution were complex values.

Analysis of F4 Algorithm [10,14]
F4 algorithm uses Gaussian elimination to speed up the time-consuming step of "critical pair" reductions, also the reduction is not a singles S-Polynomial at a time; but There are no criteria to detect useless critical pairs.

VI. F5 ALGORITHM

The second group of improvements is trying to remove such useless computations by removing unnecessary S-polynomials. One way how this can be done is to apply a selection strategy [15] which will eliminate S-polynomials that would reduce to zero. The best known algorithm which solves this problem is another algorithm from Faug´ere called F5 [16]. This algorithm is based on ideas from the paper [17], and in many cases, results in computations without reductions to zero. The idea of F5 algorithm is to compute simultaneously a Grobner basis and a basis of the module of syzygies: a critical pair is not considered if the corresponding syzygy is a linear combination of some elements of the current basis of the module of syzygies. They have in all in common to use implicitly or explicitly the trivial sysygies $fifj = fjfi$. Another common point is that all the algorithms are nearly Buchberger's algorithm except that some reductions are avoided. The efficiency of those algorithms is not yet satisfactory in theory and practice because a lot of useless critical pairs are not removed. For instance we quote from [15] that "many useless pairs are discovered, but it involves a lot of extra computation, so the execution time is increased The strategy in this section is to take into account only the trivial syzygies $f_i f_j - f_j f_i = 0$ but not to compute the module of syzygies. This imply two major differences with the standard Buchberger's algorithm or the F4 algorithm: first we need to compute all the Grobner basis of the following ideals (f_m), (f_{m-1}, f_m), . . . , $(f_1, \ldots, f_m)$. The second difference is that some reductions are not allowed; as a result the reduction of one polynomial by a list of polynomials may be several polynomials. A consequence of the restriction to trivial syzygies is that, in worst cases, the algorithm does not avoid all the useless pairs: for instance if we have two times the same polynomial in the original equations there is a reduction to zero. That if the input system is a regular sequence, then there is no reduction to zero. Moreover, in practice, for most systems there is no reduction to zero.

Analysis of F5 [16, 17]
This algorithm *is* limited to solve a system of homogenous polynomials under finite field, if system is regular; there will be no reduction pairs. So this algorithm is not suitable for cryptanalysis of symmetric cryptosystems. The only application was HFE.

VII. CONCLUSION AND FUTURE WORK

In this paper we discussed various types of cryptanalysis techniques. If we know about various types of attacks then it is very useful to improve the cryptographic algorithm or encryption techniques. From the previous discussion; the overall conclusion F4 is more efficient than grobner bases (reaches to the same result in faster processes). The result of anatomy of grobner bases F5 is always efficient than F4. But F4 is more suitable than F5 for cryptanalysis. Future work will try to mention all variants of F4 To deduce which of them is the best for cryptanalysis.

REFERENCES

[1] Shannon, C. E. 1949. "Communication Theory of Secrecy Systems," Bell System Technical Journal 28, pages 656 - 715.
[2] Willi Geiselmann and Rainer Steinwandt, "Cryptanalysis of a Hash Function" , ICISC ,2007.
[3] Christopher Swenson," Modern Cryptanalysis Techniques For Advanced Code Breaking", wiley publication Inc.2008.
[4] Ashish Kumar Kendhe, Himani Agrawal International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-2, May 2013.
[5] William Stallings (2003), "Cryptography and Network Security", 3rd edition, Pearson Education.
[6] Chengqing Li ,"Cryptanalysis of Some Multimedia Encryption Schemes",IEEE transactions on multimedia ,vol.10,no.3,2008.
[7] B. Buchberger. Bruno buchberger's phd thesis 1965: "An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal". Journal of Symbolic Computation, pages 475–511, 2006.
[8] B. Buchberger." A criterion for detecting unnecessary reductions in the construction of groebner bases". In EUROSAM'79, pages 3–21, 1979
[9] B. Buchberger. Grobner-Bases: "An Algorithmic Method in Polynomial Ideal Theory". Reidel Publishing Company, Dodrecht - Boston - Lancaster, 1985.
[10] Zazuna Kukelova, "Algebraic Methods in Computer vision", Doctoral thesis Czech Technical Universtyin pragne, Feb 2013
[11] A. Joux and V. Vitse. "A variant of the f4 algorithm". In CT-RSA'11, pages 356–375, 2011.
[12] J. C. Faugère. "A new efficient algorithm for computing Gröbner bases (F4)", In ISSAC 2002, ACM Press: New York, 2002; 75-83
[13] Pure and Applied Algebra, 139(1-3):pages 61–88, 1999.
[14] D.A. Cox, J. Little, and D.O'Shea. "Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra". Undergraduate Texts in Mathematics. Springer, 2010
[15] H. M. Moller, T. Mora, and C. Traverso." Grobner bases computation using sysygies", ,1992
[16] J. C.Faug`ere. "A new efficient algorithm for computing grobner bases without reduction to zero (F5)". In ISSAC'02, pages 75–83, 2002.
[17] J. Faug`ere, P. Gianni, D. Lazard, and F. Mora. "Efficient computation of zero-dimensional grobner bases by change of ordering". Journal of Symbolic Computation, 16(4):pages 329–344, October 1993.