A Complete algorithm for local inversion of maps: Application to Cryptanalysis

Virendra Sule
Dept. of Electrical Engineering
Indian Institute of Technology Bombay, India
(vrs@ee.iitb.ac.in)
January 24, 2022

Subject Classification: cs.CC, cs.CR, cs.SC, cs.LO

Abstract

For a map (function) $F(x) : F_2^n \rightarrow F_2^n$ and a given $y$ in the image of $F$ the problem of local inversion of $F$ is to find all inverse images $x$ in $F_2^n$ such that $y = F(x)$. In Cryptology, such a problem arises in Cryptanalysis of One way Functions (OWFs). The well known TMTO attack in Cryptanalysis is a probabilistic algorithm for computing one solution of local inversion using $O(\sqrt{N})$ order computation in offline as well as online for $N = 2^n$. This paper proposes a complete algorithm for solving the local inversion problem which uses linear complexity for a unique solution in a periodic orbit. The algorithm is shown to require an offline computation to solve a hard problem (possibly requiring exponential computation) and an online computation dependent on $y$ that of repeated forward evaluation $F(x)$ on points $x$ in $F_2^n$ which is polynomial time at each evaluation. However the forward evaluation is repeated at most as many number of times as the Linear Complexity of the sequence $\{y, F(y), \ldots\}$ to get one possible solution when this sequence is periodic. All other solutions are obtained in chains $\{e, F(e), \ldots\}$ for all points $e$ in the Garden of Eden (GOE) of the map $F$. Hence a solution $x$ exists iff the former sequence is periodic or a solution occurs in a chain starting from a point in GOE. The online computation then turns out to be polynomial time $O(L^k)$ in the linear complexity $L$ of the sequence to compute one possible solution in a periodic orbit or $O(l)$ the chain length for a fixed $n$. Hence this is a complete algorithm for solving the problem of finding all rational solutions $x$ of the equation $F(x) = y$ for a given $y$ and a map $F$ in $F_2^n$. Due to the dependence on forward evaluation by $F$ only, in the online computation this algorithm is expected to be scalable for inversion in realistic cases of maps when the linear complexity $L$ of the sequence above or lengths of chains are not exponential in $n$. The chains emerging from the GOE are independent and can be computed in parallel. The hard (NP) offline computation consists of computing all points in the GOE.
and the set of all orbit lengths of closed orbits of the iterative action of

\( F \).

1 Introduction

Inversion of a function is a fundamental problem of both pure and applied
mathematics and has enormously wide implications and important applications.
In this paper we consider the problem local inversion of a function \( F : \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n} \)
which is to find all rational (in \( \mathbb{F}_{2^n} \)) solutions \( x \) to the equation \( F(x) = y \) for
given \( y \) in \( \mathbb{F}_{2^n} \). For instance, inversion of One Way Functions (OWFs) is a central
problem of Cryptanalysis. Although such a problem can obviously be solved
by an exponential time brute force search algorithm, the challenge is to develop
algorithms which will be practically feasible for solving realistic cases of inversion
arising in applications where brute force is infeasible. Another methodology to
solve such a problem better than brute force approach is by means of a solver for
Boolean equations which should return all solutions \( x \). Such a solver in which
all solutions are represented by a complete set of implicants of the system of
equations is announced in \( [9] \). In this paper we propose an algorithm which
utilizes an offline computation on \( F(x) \) while the online computation utilizes
only the forward computation of the map \( F(z) \) on points \( z \) and finds all solutions
using the sequence \( S(F, z) = \{ z, F(z), \ldots \} \). Since the forward computation \( F(z) \)
is assumed to be efficiently feasible, the estimates of online computation required
are expected to be better than solving the Boolean system \( F(x) = y \) directly.
The offline computation however involves hard computation and is assumed to
be feasible with sufficient time and memory space for some of the realistic cases
of \( F \). For such cases local inversion can be practically feasible if the online
computation is feasible.

An important application of local inversion arises in Cryptanalysis of cipher
algorithms where it is necessary to find all inputs (arguments) to a known map
\( F \) for a given output (map value). For instance, the output is associated with a
known ciphertext of a known or chosen plaintext while the unknown input is the
secret key. The cipher algorithm is designed in such a way that the map \( F \) is a
OWF which is efficient for forward operation on its argument but hard to invert
given an output. The well known probabilistic algorithm, Time Memory Trade-
off (TMTO) attack is often used to solve such an inversion problem. TMTO
attack finds one inverse of a map \( F : X \rightarrow X \) in the set \( X \) of \( n \)-bit strings given
an output \( y = F(x) \) probabilistically. TMTO attack has exponential complexity
(in \( n \)) and is equivalent to \( O(\sqrt{N}) \) order Birthday attack where \( N = |X| = 2^n \).
Hence TMTO has a success probability of 63% once an offline data is gathered
which is also a problem of \( O(\sqrt{N}) \) order of computation. Apart from being
probabilistic TMTO is not meant for computing all solutions of \( F(x) = y \) nor
does the method utilize the structure of the finite field possible for strings of
\( n \)-bits. A complete algorithm on the other hand by definition succeeds with
certainty in computing the solutions. Purpose of this paper is to propose a
complete algorithm for the local inversion problem by utilizing the structure
of the map $F$ as an $n$-tuple polynomial function of $n$ variables over $\mathbb{F}_2$. An important outcome of the result of this paper is that when the local inverse for a given $y$ is unique (which is most likely in a practical situation) and the linear complexity of the sequence $S(F, y)$ is not too large, then a key recovery attack on the cipher algorithm is practically feasible.

1.1 Offline and online computations for local inversion

We show in this paper that the computation of local inversion can be divided into offline and online phases.

1. The offline phase is used for computing the set of all possible periods of closed orbits of iterations of $F$ which may require an exponential order of computation in general.

2. Offline computation is also required for computing the Garden of Eden of the map $F$ which is the set of all $y$ which are outside the range of $F$ (or those $y$ for which there is no rational solution to the equation $F(x) = y$). This computation is equivalent to computing all satisfying assignments of a Boolean system of equations.

3. Then in online computation, one local inverse $x$ when it lies in a periodic orbit containing $y$ can be computed in polynomial order of complexity $O(L^k)$ where $L$ is the linear complexity of the sequence $S(F, y) = \{y, F(y), F^{(2)}(y), \ldots\}$ under iterations of $F$. An upper bound on $L$ is the maximum period of possible orbits of $F$.

4. With the knowledge of GOEs of $F$, other inverses of $y = F(x)$ belong to the sequences under iterations $F^k(z)$ on points $z$ in the GOE, known as chains of $F$ which is a polynomial time online parallel computation independent for each point $z$ in GOE.

The most important advantage of this splitting of computation is that, since the period of the sequence $S(F, y)$ is the order of the minimal polynomial in $\mathbb{F}_2[X]$ the polynomial ring over $\mathbb{F}_2$, the unique solution of the local inverse in the periodic orbit of $y$ in the above sequence has a polynomial time solution in the degree of the minimal polynomial of the sequence (or its linear complexity $L$). All other solutions too are computed by the forward application of $F$. Hence we show that by taking into account the structure of the finite field and the function $F$, one possible solution of the local inversion can be computed in $O(L^k)$ once the offline computation is completed.

1.2 Relations with previous literature

Linear complexity and the minimal polynomial of sequences have been well studied in the Cryptology literature since long \cite{15,16}. The Berlekamp Massey algorithm \cite{17} computes the minimal polynomial efficiently. However linear complexity was discovered for modeling and predicting the sequence as generated
by the output of a LFSR and hence predicting an exponentially long sequence from just $2L$ samples by a linear recurrence of a polynomial size subsequence.

The computation of linear complexity $L$ of a sequence is also not feasible if $L$ is exponential comparable to the period. This paper shows another application of linear complexity and the minimal polynomial of the sequence $S(F,y)$, that of inverting the function $F(x)$ at its value $y$.

Our results are further development of ideas on linear representation of maps $F$ using the Koopman operator and its applications recently announced in [6]. It is shown that one solution of the inverse is obtained in polynomial time in $O(L^K)$ where $L$ is the linear complexity of the sequence $S(F,y)$. Use of linear complexity of such sequences for inversion of $F$ is believed to be a new proposition of this paper and follows from the ideas of [7]. Other known literature relates to this problem as follows.

1.2.1 Inversion of permutation polynomials

The problem of inversion of permutation polynomials in finite fields has a vast literature. References [11, 12, 13] are good representatives. However inversion of permutation polynomials does not result in a method for local inversion. Recently in [7] a linear representation of functions in finite fields is proposed and it is shown that when $F$ (is a permutation function) with the global inverse $G$ then the inverse $G$ as well as its linear representation also follows from that of $F$. The linear representations of inverses are inverses of linear representations. However linear representation of $F$ does not solve the problem of local inversion when $F$ is not globally invertible (since $G$ does not exist). Hence local inversion problem needs to be solved afresh.

1.2.2 Relation with TMTO

As discussed above, the Time Memory Tradeoff (TMTO) attack on OWFs is a well known algorithm for solving the local inversion of a map $F : X \rightarrow X$ in $X = \{0,1\}^n$ given $y = F(x)$ in $X$. TMTO algorithm was proposed by Hellman in [2] and has been well researched over last several years as shown in [3, 4, 5]. TMTO algorithm is however probabilistic and does not utilize structure of the set $X$ as a vector space over a finite field and its success probability is based on the Birthday attack which has complexity of order $O(2^{n/2})$. In practice TMTO attack requires gathering of multiple large sized tables offline not just one to overcome merging of chains, false positive solutions and even failure to capture a solution. However the online search can be carried out in parallel which is advantageous.

1.2.3 Relation with Boolean equation solving

An algorithm for representing all solutions (satisfying assignments) of a Boolean systems of equations is proposed in [9]. This algorithm is completely thread parallel hence the time performance improves with increase in memory space.
The problem of representing all solutions of Boolean equations is however of \#P class whose special case, that of deciding solvability of 3-CNF system is an NP complete problem. If the Boolean equations are represented in the form \( y = F(x) \) for a known \( y \) in \( \mathbb{F}_2^n \) with \( n \)-unknowns taking values in \( \mathbb{F}_2 \), this method can be used for local inversion. Hence all solutions of inverses can be solved using the algorithm proposed in [9] better than brute force search. However if the offline computation on \( F \) of the method proposed in this paper is feasible, the local inversion does not require solution of non-linear Boolean equations but only depends on forward computations of the function. Hence the algorithm proposed in this paper is a definite way to solve these hard problems even without solving the system of Boolean equations and only resorting to forward computation of the map \( F \) if the offline computation is feasible.

1.2.4 Related work on local inversion

Problem of solving rational roots of polynomial of systems of equations over finite fields is one of the well known hard problems of computation. This problem has been addressed in past by algebraic geometric and probabilistic algorithmic methods [10,11]. The case of deciding the solvability of a multivariate quadratic system is an NP complete problem. Hence in general the local inversion problem is computationally hard. More specifically the problem of inversion of permutation polynomials requires symbolic arithmetic and results in an inverse function [12]. The problem of local inversion of a function \( F(x) \) at a point \( y = F(x) \) is thus a special counterpart of both of these problems. Any algorithm for finding roots of polynomial systems can solve all rational roots in \( \mathbb{F}_q \) of the equation \( y = F(x) \) given \( y \). Similarly if \( F \) is a permutation of \( \mathbb{F}_q \) and \( G \) is the global inverse permutation then \( x = G(y) \) is the unique inverse image of \( y \). However when \( F \) is not a permutation these methods cannot solve the local inversion problem. For \( \mathbb{F}_{2^n} \) existence and computation of all rational roots of \( y = F(x) \) can be accomplished by the Boolean system solver [9]. However no such algorithm is known for fields in other characteristics.

2 Structure of trajectories and solutions

Solutions to the equation \( F(x) = y \) are governed by the structure of trajectories of the iterations of the map \( F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n \) and on what part of a trajectory \( y \) belongs. A trajectory of the map \( F \) through a point \( z \) in \( \mathbb{F}_2^n \) is the sequence of iterations \( \{ z, F(z), F^2(z), \ldots, F^k(z), \ldots \} \). Due to finiteness of \( \mathbb{F}_{2^n} \) and invariance of \( F \) with respect to \( k \), it follows that there exist \( 0 \leq l \leq k \) such that \( F^k(z) = F^l(z) \) for any \( z \). Hence every trajectory can be uniquely split into a periodic part

\[
P(z) = \{ F^{(k)}(z), F^{(k+1)}(z), \ldots, F^{(k+m-1)}(z) \}
\]

such that

\[
F^{(k)}(z) = F^{(k+m)}(z)
\]
where $m$ depends on $z$, called a periodic orbit of period $m$ and a unique segment of a chain

$$C(z) = \{z, F(z), \ldots, F^{(k-1)}(z)\}$$

which merges with a periodic orbit or another chain. Points of a periodic orbit of length $m$ are called $m$-periodic points. The fixed points of $F$ which are points $z$ such that $F(z) = z$ are 1-periodic points. The set of all points $z$ which are not in the range of $F$ is called the Garden of Eden of $F$. Following lemma follows from this structure of trajectories.

**Lemma 1.** Given an arbitrary point $y$ in $\mathbb{F}_2^n$ solutions of $F(x) = y$ are described by following possibilities

1. $F(x) = y$ has no solution iff $y$ belongs to the GOE of $F$.
2. $F(x) = y$ has solution in a periodic orbit $P$ iff $y$ belongs to $P$. Such a periodic orbit is unique. Given $y$ in a periodic orbit $P$ and the unique solution to $F(x) = y$ in $P$, all other possible solutions belong to the chains $f^k(z), k \geq 1$ for $z$ in the GOE of $F$.
3. If $y$ is neither in a periodic orbit nor in the GOE, then all solutions to $F(x) = y$ arise in some of the chains $F^{(k)}(z), k \geq 1$ for $z$ in the GOE.

**Proof.** From the definition of the periodic orbits (1) and chains (2) it follows that the space $\mathbb{F}_2^n$ is partitioned by the action of $F$ in periodic orbits and segments of chains from points in GOE to a point which is mapped by $F$ to a periodic orbit. Hence given any point $y$ it is either on a unique chain segment or on a periodic orbit. If there is no solution to $F(x) = y$ then $y$ belongs to GOE conversely if $y$ is in GOE then there is no $x$ such that $F(x) = y$. If $y$ is in a periodic orbit $P$, there is unique predecessor $x$ in $P$ such that $F(x) = y$. Any other solution $x$ which is outside $P$ cannot be in any other periodic orbit since the two orbits cannot intersect at $y$. Hence all other solutions $x$ are on chains merging with $P$ at $y$ under iterations of $F$. Hence every solution outside $P$ is on one of the chains $F^{(k)}(z)$ for $z$ in GOE and $k \geq 1$.

2.1 Solution in a periodic orbit

We need to revisit some of the well known background on the theory of recurrent sequences over finite fields [14,13] for the special iterative sequences we encounter in this problem of local inversion of maps. The theory of linear recurring sequences over finite fields in [14,13] considers recurrence relations with for sequences and their co-efficients in the same field. On the other hand the sequences encountered in the problem of local inversion are in $\mathbb{F}_2^n$ and generated iteratively by the map $F$ while the co-efficients are over $\mathbb{F}_2$. Because of this minor difference we choose to revisit proofs of fundamental results for the special sequences and show that the well known results go through without change. Consider a periodic orbit of iterations of $F$ which contains $y$, denoted as

$$S(F, y) = \{y, F(y), F^{(2)}(y), \ldots, F^{(N-1)}(y)\}$$
which has period $N$ which is the smallest number such that $F^k(y) = F^{k+N}(y)$ for $k = 0, 1, 2, \ldots$. We say that the sequence $S(F, y)$ has a linear recurrence relation over $\mathbb{F}_2$ of degree $m$ if there exist constants $\{a_0, a_1, \ldots, a_{(m-1)}, a_m\}$ in $\mathbb{F}_2$ such that

$$a_m F^{(k+m)}(y) + \sum_{i=0}^{(m-1)} a_i F^{(k+i)}(y) = 0 \quad (3)$$

The periodicity condition on a sequence $S(F, y)$ of period $N$,

$$F^{(k+N)}(y) = F^{(k)}(y)$$

gives existence of one such linear recurrence relation with $m = N$. The polynomial

$$\phi(X) = a_m X^m + \sum_{i=0}^{(m-1)} a_i X^i \quad (4)$$

associated with a linear recurrence relation is called a characteristic polynomial of $S(F, y)$. The indeterminate $X$ of a characteristic polynomial represents the operation

$$X^k(y) = F^{(k)}(y) \quad (5)$$

($X^0(y)$ is the identity map), which is also compatible with the compositional operation by $F$ as seen by

$$X^{(k+l)}(y) = F^{(k+l)}(y) = F^{(k)}(F^{(l)}(y))$$

The linear operation of terms in powers of $X$ is by definition

$$(X^k + X^l)(y) = X^k(y) + x^l(y)$$

A monic linear recurrence relation i.e. with $a_m = 1$ and of least degree is unique and the monic polynomial $\phi(X)$ associated with it is called the minimal polynomial. In a finite field $\mathbb{F}_q$ a polynomial $\phi(X)$ in the polynomial ring $\mathbb{F}_q[X]$ satisfying $\phi(0) \neq 0$ has an order, which is the minimum number $N$ such that $\phi(X)$ is a divisor of $X^N - 1$.

Following result follows from this background.

**Proposition 1.** The sequence $S(F, y)$ is periodic iff it has a minimal polynomial which divides any of its characteristic polynomials. The minimal polynomial satisfies $a_0 \neq 0$ and the period $N$ of $S(F, y)$ is the order of the minimal polynomial.

**Proof.** Let $S(F, y)$ be periodic of period $N$, then it has the characteristic polynomial $\phi(X) = X^N - 1$. Hence it has a minimal polynomial. Let the minimal polynomial of $S(F, y)$ be,

$$m(X) = X^m + \sum_{i=0}^{(m-1)} \alpha_i X^i \quad (6)$$

Then from the definition of the operation $X$ in (5) and interpretation of $X^k$ as compositional operation $F^{(k)}(\cdot)$, $m(X)$ is the smallest degree polynomial such that

$$m(X)(y) \triangleq F^{k+m}(y) + \sum_{i=0}^{(m-1)} \alpha_i F^{(k+i)}(y) = 0$$

If $\phi(X)$ is any characteristic polynomial and $R(X)$ the residue $\phi(X) \mod m(X)$ then there exists a polynomial $Q(X)$ such that $\phi(X) = Q(X)m(X) + R(X)$. Hence using the above algebraic rules of operation of $X$ on $y$ through the action of $F$ it follows that

$$\phi(X)y = R(X)y = 0$$

hence due to minimality of degree of $m(X)$ and noting that $\deg R(X) < \deg m(X)$ it follows that $R(X)$ must be a zero polynomial. Hence in particular if $S(F, y)$ is periodic of period $N$ then $m(X)|(X^N - 1)$. It also follows from this divisibility by $m(X)$ that $m(0) = \alpha_0 \neq 0$. Moreover since $N$ is the smallest, $N$ must be the order of $m(X)$ as an element of $\mathbb{F}_2[X]$. Conversely if $S(F, y)$ has minimal recurrence relation and the minimal polynomial satisfies $m(0) \neq 0$ then for $N$ the order of $m(X)$ it is a divisor of $X^N - 1$. Hence $(X^N - 1)(y) = 0$ which proves that $S(F, y)$ is periodic of period $N$. \qed

2.1.1 Rank criterion and computation of minimal polynomial

Consider the sequence $S(F, y)$. It is a-priori not possible to know whether the sequence is periodic from a short length samples $F^{(k)}(y)$ (for small $k$) since the period can be exponentially large. The issue here is that while the sequence $S(F, y)$ is not fully available for computation, to compute a linear recurrence relation (3) of shortest degree. In fact this is what is the well known computation of linear complexity of a sequence by the Berlekamp-Massey algorithm. Important point is that such a computation requires the a-priori knowledge of the period of the sequence $S(F, y)$ to stop the search for the degree of the minimal polynomial. We introduce the Hankel matrix

$$H_{m+j} = \begin{bmatrix} y & F^{(1)}(y) & \cdots & F^{(m-1)}(y) \\ F(y) & F^{(2)}(y) & \cdots & F^{(m)}(y) \\ \vdots & \vdots & \ddots & \vdots \\ F^{(j-1)}(y) & F^{(j+m-1)}(y) & \cdots & F^{(j+2m-2)}(y) \end{bmatrix}$$

Following proposition is well known on computation of minimal polynomial of sequences in finite fields. In the present context the sequence is the vector sequence $S(F, y)$ in $\mathbb{F}_2^n$. Hence the Hankel matrix (7) is no more square as in the case of sequences over finite fields but is a block Hankel matrix.

**Proposition 2.** Let $S(F, y)$ be periodic then it has minimal polynomial of degree $m$ iff

$$\text{rank } H_{m+j} = \text{rank } H_m = m$$

(8)
for all \( j = 0, 1, 2, \ldots \). The co-efficient vector
\[
\hat{\alpha} = (\alpha_0, \ldots, \alpha_{(m-1)})^T
\]
of the minimal polynomial is the unique solution of
\[
H_m \hat{\alpha} = h(m)
\]  
(9)
where
\[
h(m) = [F^m(y), F^{(m+1)}(y), \ldots, F^{(2m-1)}(y)]^T
\]

Proof. In Proposition 1 it is shown that the sequence \( S(F, y) \) is periodic iff it has a minimal polynomial. Hence it is required to find its degree and coefficients using the sequence. Let \( m(X) \) denoted in (6) be the minimal polynomial of degree \( m \). Then there is recurrence relation satisfied by the sequence \( S(F, y) \)
\[
F^{m+j}(y) + \sum_{i=0}^{(m-1)} \alpha_i F^{(k+i+j)}(y) = 0
\]
writing these equations for \( j = 0, 1, 2, \ldots \) and noting that there is a unique solution to the co-efficients \( \alpha_i \) is equivalent to the rank condition (8) and the co-efficient vector is the solution of the linear system (9).

Conversely if the rank condition is satisfied there is a shortest linear recurrence relation of degree \( m \) satisfied by the sequence \( S(F, y) \). The co-efficients of the recurrence are obtained uniquely by the linear system (9).

2.1.2 Algorithm to compute a solution in a periodic orbit

One solution of \( F(x) = y \) when sequence \( S(F, y) \) is periodic, is obtained as described in the theorem below.

**Theorem 1.** Let \( S(F, y) \) be a periodic sequence and \( m(X) \) as described in (6) be its minimal polynomial. Then there is a unique solution to \( F(x) = y \) in \( S(F, y) \) given by
\[
x = (1/\alpha_0)[F^{(m-1)}(y) + \sum_{i=1}^{(m-1)} \alpha_i F^{(i-1)}(y)]
\]  
(10)

Proof. Let \( S(F, y) \) be periodic of period \( N \). The point \( x = F^{(N-1)}(y) \) then satisfies the equation \( F(x) = y \), hence this is one and the unique solution of the equation in the periodic orbit. But then it follows that for this solution \( x \), the periodic sequence \( S(F, x) = S(F, y) \). Hence if \( m(X) \) as described in (6) is the minimal polynomial of \( S(F, y) \),
\[
m(X)(x) = F^{(m)}(x) + \sum_{i=0}^{(m-1)} \alpha_i F^i(x) = 0
\]
From this expression of \( m(X)(x) \) the term \( x \) can be solved uniquely since \( \alpha_0 \neq 0 \).

\[
x = (1/\alpha_0)[F^{(m)}(x) + \sum_{i=1}^{(m-1)} \alpha_i F^{(i)}(x)]
\]

Then by using the condition \( y = F(x) \) one gets the relation (10). This is the expression of the unique solution \( x \) in the periodic orbit \( S(F,y) \).

We note following important points about the solution (10).

1. The unique solution (10) in the periodic orbit \( S(F,y) \) is expressed in terms of the linear combinations of the sequence \( S(F,y) \) which consists of only forward operations of the map \( F \) (which is assumed to be feasible in practical time).

2. The number of terms in (10) are \( m \) the degree of the minimal polynomial. Hence the solution \( x \) is computed in polynomial order of computations in \( O(m^k) \) given \( y \).

3. Since \( S(F,y) \) is periodic of period \( N \) if it has a minimal polynomial, \( x = F^{(N-1)}(y) \) is also the same solution as computed in (10). However \( N \) may be exponentially large hence this formula for computation of \( x \) is not feasible without knowing the period \( N \).

4. If a solution \( x \) is obtained by a possible minimal polynomial as in (10) then knowing the possible period \( N \) it can be easily verified whether \( x \) lies in the periodic orbit \( S(F,y) \) by \( F(x) = y \) instead of verifying by the equation \( F^{(N-1)}(y) = x \).

5. If the degree \( m \) itself is of polynomial order in \( n \) (number of variables) then the online computation of the solution \( x \) in \( S(F,y) \) is computable in polynomial time. (The situation is comparable to primitive polynomials of degree \( n \) whose order is \( N = 2^n - 1 \). Hence \( m \) is expected to be of the order \( \log N \) in same proportions (density) as primitive polynomials of degree \( n \) among all polynomials).

Based on this theorem the online algorithm to compute one solution \( x \) when \( y \) is in a periodic orbit \( S(F,Y) \) is Algorithm 1.

2.1.3 Online Incomplete algorithm to compute local inverse

It is interesting to observe how an online incomplete online algorithm can be constructed which may possibly find a solution given \( y \) but without any offline computation. Such an algorithm may be practically useful within certain bounds on the degree of the minimal polynomial. Let a bound \( M \) on the degree of the minimal polynomial \( m(X) \) is specified and the step of computation of \( m(X) \) is stopped when degree \( m \) of a possible minimal polynomial exceeds \( M \). The bound \( M \) should be theoretically at most equal to any of the possible periods
Algorithm 1: Online Algorithm: Unique solution in a periodic orbit

1: **procedure** `SolutionInPeriodicOrbit`(`S(F, y)`)
2: **Input**: Set of all possible periods $\Pi$ (computed offline), $N_m = \max \{\Pi\}$, $m_0$ a minimum value of degree below which minimal polynomial does not exist.
3: **Output**: One solution of $F(x) = y$ in the periodic orbit $S(F, y)$ if one exists otherwise returns that there is no solution in a periodic orbit.
4: Set $m = m_0$
5: **repeat**
6: Compute matrices $H_m, H_{m+1}$
7: if $\text{rank } H_m = \text{rank } H_{m+1} = m$
8: find solution $\hat{\alpha}$ (co-efficients of polynomial $m(X)$) in (9).
9: **for** $N \in \Pi$ **do**
10: while $(m(X)|X^N - 1) \land (F(x) = y)$ **do**
11: Return solution $x$ as computed in (10)
12: **end while**
13: **end for**
14: **else**
15: $m \leftarrow m + 1$
16: **end if**
17: **until** $m > N_m$
18: Return: No solution in periodic orbits.
19: **end procedure**
of closed orbits $N$ of $F$. However $M$ is chosen from a point of view of feasibility of the computation of the minimal polynomial using the rank criterion of the section 2.1.1. Hence $M$ may be chosen of polynomial order $O(n^k)$ in the number of variables for suitable $k$. Algorithm 2 shows this computation.

**Algorithm 2** Online Algorithm (Incomplete): Unique solution in a periodic orbit or report no conclusion

1: procedure POSSIBLESOLUTIONIN($S(F, y)$)
2: Input: $y$, $m_0$ a minimum value of degree below which minimal polynomial does not exist and $M$ maximum value.
3: Output: One solution of $F(x) = y$ in the periodic orbit $S(F, y)$ if one exists with minimal polynomial of degree $m < M$ otherwise returns that there is no conclusion.
4: Set $m = m_0$
5: repeat
6: Compute matrices $H_m, H_{m+1}$
7: if $\text{rank } H_m = \text{rank } H_{m+1} = m$
8: then
9: find solution $\alpha$ (co-efficients of polynomial $m(X)$) in (9)
10: compute $x$ as a possible solution as in (10).
11: compute $N = \text{ord } m(X)$
12: while $F(x) = y$ do
13: Return solution $x$ as computed in (10).
14: end while
15: else
16: $m \leftarrow m + 1$
17: end if
18: until $m > M$
19: Return: No conclusion about solution in a periodic orbit.
20: end procedure

Algorithm 2 is practically feasible and results in one correct solution when the linear complexity of $S(F, y)$ is less than $M$. The algorithm eliminates false positive solutions by checking the equation $F(x) = y$ in step 11.

Theorem 1 leaves open the question whether there are other solutions outside the periodic orbit $S(F, y)$ or when such solutions exist when $S(F, y)$ is not periodic. This problem is considered next.

### 2.2 Solutions outside periodic orbits

From the lemma it follows that solutions which are not in the periodic orbit $S(F, y)$ or if $S(F, y)$ is itself not periodic can only possibly arise on chains $F^k(z)$ starting from $z$ in GOE of $F$. Hence assuming the GOE is computed in offline
computation an online algorithm for discovering solutions on chains from GOE uses only forward computations of the map $F$. However an efficient termination condition for the forward computation on a chain is needed.

2.2.1 Termination of a chain on a periodic orbit and all solutions on the chains

If $z(k) = F^{(k)}(z)$ is a sequence of points on a chain, the chain terminates when $z(k)$ meets a periodic orbit. If all possible periods of periodic orbits of $F$ are known a-priori then for each $N$ of these periods, the relation $F^N(z(k)) = z(k)$ seems to be the only way to check whether $z(k)$ is on a periodic orbit of period $N$. Hence it is necessary to show that even when $N$ is exponential order in $n$, evaluation $F^N(z)$ can be carried out efficiently in polynomial time in $\log N$ on any point $z$ in $\mathbb{F}_2^n$. This is shown in Appendix. Algorithm 3 to compute solutions of $F(x) = y$ outside periodic orbits $S(F, y)$ can be as follows.

**Algorithm 3** Solution in a chain

1: **procedure** Solutioninachain($z, y$) 
2: **Input:**
3: 1. $z$ a point in GOE (computed offline),
4: 2. $y$ online value of $F(x)$ to be inverted,
5: 3. set of all possible periods $\tilde{\Pi}$ (computed offline)
6: **Output:** Solution of $F(x) = y$ on a chain starting from $z$.
7: for $N$ in $\tilde{\Pi}$ do 
8: $k = 1,$
9: Compute $z(k) = F^{(k)}(z)$
10: while $z(k) \neq y$ do
11: if $F^{(N)}(z(k)) = z(k)$ then
12: Print: No solution on the chain from $z$.
13: end if
14: end while
15: if $z(k) = y$ then
16: a solution is
17: $x = z(k - 1) = F^{(k-1)}(z)$
18: end if
19: $k \leftarrow k + 1$
20: \hfill $\triangleright$ The loop always ends because $z(k)$ joins a periodic orbit of period $N$ or is equal to $y$
21: end for
22: **end procedure**

Algorithm 3 is repeated in parallel for all points $z$ in GOE. Both the online algorithms Algorithm 1 for computing an inverse in a periodic orbit containing $y$ and Algorithm 3 to compute all other solutions, require an a-priori (offline) computation on $F$ which is independent of $y$. This computation creates a data,
the set $\hat{\Pi}$ of all possible periods of periodic orbits of $F$ and the set GOE. This offline computation is discussed in the next section.

### 3 Computation of possible periods of orbits and the GOE

For linear maps $L : F_2^n \rightarrow F_2^n$, [1] showed how the computation of the set of all periods of closed orbits and the set of all points in the GOE is accomplished by computing the rational canonical form of $L$ over $F_2$ in a fixed basis in $F_2^n$. The online algorithms presented in the previous section to compute local inverse of $F$ at $y$ make use of the offline data of all possible periods of closed orbits of the map $F$ and the GOE of $F$. However since $F$ is not linear the linear algebraic computation of [1] is no more applicable. In this section we take up the problem of computation of the GOE when $F$ is not linear. We show that the computations of all periods of closed orbits of the map $F$ can be achieved by computing an analogous linear algebraic map whose set of periods of closed orbits contains this information of periods of orbits of $F$. The GOE of $F$ however needs to be computed separately by solving a Boolean system. These two computations are achieved as follows.

1. Computation of all possible periods of orbits of map $F$, by showing that this set belongs to the set of periods of all orbits of a linear map dual to $F$ restricted to a dual vector space. This observation allows the linear algebraic computation of all possible periods by the rational canonical form as shown in [1].

2. Computation of GOE of the map $F$ by means of the implicant based Boolean solver algorithm of [9].

Both of these computations are expected to be hard. However since these are performed offline on $F$ they may be feasible practically for special instances of $F$ given sufficient memory for parallel computation.

#### 3.1 Linear representation of map $F$

To compute the information on periods of closed orbits of the map $F$ a suitable representation of the action of the map $F$ in a linear space is logically appropriate. We shall revisit the essential ideas on this representation which have recently been announced in [6]. The map $F$ acts in the space $F_2^n$ as a polynomial function of the co-ordinates of the space $F_2^n$. Let $V$ denote the $F_2$-linear space of $F_2$-valued functions on $F_2^n$. Denote by $\chi_i$ the co-ordinate functions in $V$ defined by their evaluation at points $x$ in $F_2^n$ as

$$\chi_i(x) = x_i$$
where \( x_i \) is the \( i \)-th co-ordinate of \( x \). The map \( F \) has the dual action on functions \( \phi \) in \( V \) defined by the composition

\[
F^* (\phi (x)) \triangleq \phi (F(x))
\]

\( F^* \) is thus a linear operator \( \phi \circ F \) in \( V \). Consider the \( F^* \)-invariant subspace \( W \) which is the smallest invariant subspace of \( F^* \) containing all the co-ordinate functions \( \chi_i \). The space \( W \) can be obtained as the sum of cyclic invariant subspaces \( W_i \) generated by \( \chi_i \),

\[
W_i = \text{span}_{\mathbb{F}_2} \{ \chi_i, F^*(\chi_i), \ldots, (F^*)^{m_i-1}(\chi_i) \} \quad (11)
\]

where \( m_i \) is the smallest number such that \( (F^*)^{m_i}(\chi_i) \) is linearly dependent on \( W_i \). Define

\[
W = W_1 + W_2 + \ldots + W_n \quad (12)
\]

This space \( W \) is thus the smallest invariant subspace of \( F^* \) which contains all co-ordinate functions. Consider the restriction \( F_1 = F^* | W \). We define a linear dynamic system using the action of \( F_1 \) in \( W \) as shown next.

### 3.2 Computation of all periods from the linear representation

In this section we gather some of the relevant results of [6]. Consider a basis \( B \) (of functions in \( V \)) of \( W \) as the ordered set

\[
B = \{ \psi_1, \psi_2, \ldots, \psi_N \}
\]

where \( N = \text{dim} \, W \). Define the matrix representation of \( F_1 \) by the relation

\[
F_1 [\psi_1, \ldots, \psi_N] = [\psi_1, \ldots, \psi_N]K \quad (13)
\]

where \( K \) is an \( N \times N \) matrix over \( \mathbb{F}_2 \). Columns of \( K \) are the co-efficients \( k_{ij} \) appearing in the expression

\[
F_1 (\psi_i) = \sum_{j=1}^{N} k_{ij} \psi_j
\]

We first show how \( F_1 \) hence \( K \) capture information about the map \( F \) and its dynamical trajectories.

**Proposition 3.** The map \( F \) is a permutation of \( \mathbb{F}_2^n \) iff \( F_1 \) is a permutation, equivalently the matrix \( K \) is non-singular

**Proof.** Let \( \psi \) be any function on \( \mathbb{F}_2^n \). Then \( F^*(\psi) = 0 \) implies \( \psi(F(x)) = 0 \) for all \( x \). Hence \( F \) is one to one implies \( \psi(y) = 0 \) for all \( y \) in \( \mathbb{F}_2^n \) i.e. \( \psi \) is the zero function. Hence \( F^* \) is also one to one. Hence its restriction \( F_1 \) in \( W \) is also one to one and equivalently the matrix representation \( K \) is non-singular.
Conversely, assume $F$ is not surjective, then there exists a point $\beta$ in the GOE of $F$ for which there is no solution $x$ such that $\beta = F(x)$. Any function $\psi$ in $F^*(W)$ is of the form

$$\psi = \sum_{i=1}^{N} a_i \psi_i \circ F$$

where $\psi_i$ are functions in the basis $\mathcal{B}$. Then for any function $\psi$ in $F^*(W)$ as above, the value

$$\sum_{i=1}^{N} a_i \psi_i(\beta)$$

does not belong to image the image of $\psi$. In particular for a co-ordinate function $\chi_i$ in $W$ which has a representation in the basis as

$$\chi_i = \sum_{i=1}^{N} \alpha_i \psi_i$$

the value of $i$-th component $\beta_i = \chi_i(\beta)$ does not belong to the image of $F^* \chi_i$. Hence it follows that $\chi_i$ does not belong to $F^*(W)$. But since $W$ is an invariant subspace of $F^*$ which contains all co-ordinate functions $\chi_i$ it follows that $F^*(W) = F_1(W)$ does not contain the function $\chi_i$. Which implies that $F_1$ is not surjective. Equivalently $K$ is singular.

Above proposition gives a necessary and sufficient condition for $F$ to be a permutation in terms of the matrix representation $K$ on the subspace $W$. Next we draw a correspondence between the trajectories of dynamics of $F$ with dynamics of $K$ in $\mathbb{F}_2^N$.

### 3.2.1 Embedding of trajectories in trajectories of a Linear dynamic system

Consider the linear dynamic systems, called hereafter as Koopman system corresponding to $F$, defined by iterations by the linear map

$$K^T : \mathbb{F}_2^N \rightarrow \mathbb{F}_2^N \quad y \mapsto K^T y \quad (14)$$

For a trajectory

$$x, F(x), F^{(2)}(x), \ldots$$

of iterations of the map $F$ in $\mathbb{F}_2^N$ define an embedding in the trajectories of Koopman system in $\mathbb{F}_2^N$ corresponding to the basis $\mathcal{B}$ by

$$\mathbb{F}_2^N \rightarrow \mathbb{F}_2^N \quad x \mapsto \hat{\psi}(x) \quad (15)$$

where

$$\hat{\psi}(x) = [\psi_1(x), \psi_2(x), \ldots, \psi_N(x)]^T$$
Then a trajectory of the map $F$ in $\mathbb{F}_2^n$ is embedded as a trajectory of the linear map (14) in $\mathbb{F}_2^N$ by

\[
(x, F(x), F^{(2)}(x), \ldots) \mapsto (\hat{\psi}(x), F^*\hat{\psi}(x), (F^*)^2\hat{\psi}(x), \ldots) = (\hat{\psi}(x), K^T\hat{\psi}(x), (K^T)^2\hat{\psi}(x), \ldots)
\]

(16)

**Lemma 2.** Following correspondence holds between trajectories of iterations of map $F$ and that of $K^T$.

1. To every closed orbit of the iteration of the map $F$ there corresponds a closed orbit of the Koopman system (iterations of the linear map (14)) of the same period.
2. To every chain in iterations of $F$ there corresponds a chain in iterations of $K^T$ of same length.

**Proof.** The proof essentially follows from the embeddings (16) of trajectories of $F$ into that of trajectories of $K^T$. If there is a periodic trajectory $F^{(k)}(x)$ of $F$ of period $N$ then it satisfies $F^{(k+N)}(x) = F^{(k)}(x)$. Hence under the embeddings (15) and (16) the trajectory satisfies

\[
(K^T)^{(k+N)}(\hat{\psi}(x)) = (K^T)^k(\hat{\psi}(x))
\]

Since $N$ is the period of the trajectory it is also the period of the embedded trajectory.

The argument about the embedding of chain and their lengths also follows on similar reasoning.

From this lemma we get a set of all numbers which contains all periods of closed orbits of $F$ as follows.

**Theorem 2.** The set $\Pi$ of all periods of closed orbits of iterations of $F$ in $\mathbb{F}_2^n$ is a subset of the set of all periods $\hat{\Pi}$ of closed orbits of iterations of $K^T$ in $\mathbb{F}_2^N$. $\hat{\Pi}$ can be computed in polynomial time in $N$.

**Proof.** That $\Pi \subset \hat{\Pi}$ follows from the above lemma. The rational canonical form of $K^T$ gives all elementary divisors of $K^T$ and as shown in [1] the periods of closed orbits of iterations of $K^T$ are obtained as the orders of powers of elementary divisors of $K^T$. Hence computation of $\hat{\Pi}$ can be accomplished in polynomial time in $N$ which is the size of $K^T$.

For the sake of clarity of the computation of $\hat{\Pi}$, the set of periods of all closed orbits of iterations of $K^T$, we present the well known result from [1] here without proof.

**Theorem 3.** Consider the linear map $A : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$. Let

\[
\{d_1(X), d_2(X), \ldots, d_w(X)\}
\]


denote the elementary divisors of $A$ in $\mathbb{F}_2[X]$ such that $d_i(0) \neq 0$ and let each $d_i(X)$ has the form

$$d_i(X) = (p_i(X))^{r_i}$$

where $p_i(X)$ are irreducible polynomials in $\mathbb{F}_2[X]$. Then the set of all periods of closed orbits of iteration of the map $A$ is given by

$$\Pi(A) = \{1, T_{ij}, i = 1, \ldots, w, j = 1, \ldots, e_i\}$$

where

$$\{T_{ij} = \text{Order} (p_i(X))^j, j = 1, 2, \ldots, e_i\}$$

(Here Order $p(X)$ of a polynomial satisfying $p(0) \neq 0$ is the smallest number $T$ such that $p(X)$ divides $X^T - 1$). For a proof of this well known theorem reader may see section 12.10 in [1]. The theorem above shows that the computation of the set $\Pi(A)$ of periods of the map $A$ is accomplished by computing the rational canonical form of $A$ to compute all elementary divisors of $A$.

### 3.2.2 Offline algorithm for computing the set of all possible periods

An offline algorithm can thus find the input $\hat{\Pi}$ required in Algorithms 1 and 2 in following by computing $\Pi(K)$ and treating it as $\hat{\Pi}$ since by theorem 2 $\hat{\Pi}$ contains all possible periods of closed orbits of iterations of $F$.

1. Compute the matrix $K$, a representation of $F^*$ on the invariant space $W$ in a fixed basis.

2. Compute the set $\hat{\Pi}$ of all periods of closed orbits of $K$ by computing the rational canonical form of $K$ and using theorem 3. (Note that this set is same as that of set of all periods of closed orbits of iterations of $K^T$).

The computation of the matrix representation $K$ of $F$ may turn out to be a hard computation of $NP$ class. This is because the dimensions of the subspaces $W_i$ in (11) are likely to grow very fast with $n$ depending on the degrees of polynomial components of the map $F$. Also the computation of composition $F^*(f) = f \circ F$ is complicated although theoretically of polynomial time in $n$. However as this computation is offline it has much relaxed conditions on practical time and memory issues as compared to online computation. However the computation of $\Pi$ is polynomial time in $N$ the dimension of the space $W$. Doing this hard computation offline is the price one has to pay for getting a complete algorithm for inversion.

### 3.3 Computation of the GOE

The input required for the Algorithm 2 is the set GOE of $F$. This computation is formalised as the following Boolean system problem.
Problem 1. Given the map \( F \) in polynomial form, determine the set \( G \) of all simultaneous assignments \( \{y_i, i = 1, \ldots, n\} \) in \( \mathbb{F}_2 \) such that for the vector \( y = (y_1, \ldots, y_n)^T \) the Boolean system of equations

\[
F(x_1, \ldots, x_n) = y
\]

is not satisfiable for any assignments \( \{x_i, i = 1, \ldots, n\} \) in \( \mathbb{F}_2^n \). Then GOE of \( F \) is the set of all such vectors \( y \) in \( \mathbb{F}_2^n \).

Computation of GOE can be accomplished in offline by means of an algorithm which can represent all solutions of a Boolean system of equation. Such an algorithm is announced in [9]. The algorithm is in general of exponential order in \( n \). However the algorithm is parallel and given sufficient memory has complexity \( O(n) \) for parallel computation of an \( n \) variable Boolean system. This computation is described in the next section.

4 Computation of the GOE in terms of implicants

The final offline computation required to solve the local inversion problem is the computation of the GOE of the map \( F \). As stated in Problem 11 a strategy to solve this problem makes use of the Boolean system \( F(X) = Y \) defined by the map \( F \). Let the map \( F \) be described in terms of its component functions of \( n \) variables as

\[
F(x_1, \ldots, x_n) = [f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n)]^T
\]

where \( f_i \) are Boolean functions of \( n \) Boolean variables variables \( x_i \) taking values in the Boolean ring \( \mathbb{F}_2 \). We shall briefly revisit background of the implicant based representation of all solutions of Boolean systems from [9]. Let \( g(x_1, \ldots, x_n) \) be a Boolean function of \( X = \{x_1, \ldots, x_n\} \). A term \( t(X) \) in Boolean variables \( X \) is an elementary conjunction

\[
t(X) = \prod_{i \in \{1:n\}} x_i^{a_i}
\]

where \( a_i = 0, 1 \) and \( x^a = x \) for \( a = 1 \) while \( x^a = x' \) (the compliment) if \( a = 0 \). An implicant of a Boolean function \( g(X) \) is a term \( t(X) \) such that if \( t(a) = 1 \) for any assignment \( X = a \) for \( a \in \mathbb{F}_2^n \) then \( g(a) = 1 \). A set \( I_g \) of implicants of a function \( g \) is said to be complete if \( g(a) = 1 \) for any assignment \( X = a \) then there exists a term \( t \) in \( I_g \) such that \( t(a) = 1 \). A basic fact in Boolean algebra is that if \( I_g \) is a complete set of implicants of a function \( g \) then

\[
g(X) = \sum_{t \in I_g} t(X)
\]

One of the fundamental problems of computation in Boolean algebra is the computation of the representation of all \textit{satisfying} assignments of a Boolean
function $g$ of $n$ variables. An assignment $X = a$ is said to satisfy a Boolean function $g$ if $g(a) = 1$. Such a representation is accomplished by an algorithm using implicants in [9]. This shall be described next.

### 4.1 Representation of all satisfying assignments

A set of nonzero orthogonal (OG) terms $\{t_1, \ldots, t_m\}$ is one in which $t_it_j = 0$ for $i \neq j$. An OG set of terms is said to be orthonormal (ON) if

$$\sum_i t_i(X) = 1$$

Hence following fact follows.

**Lemma 3.** If $I_g$ is a complete set of OG implicants of a Boolean function $g$ then the set of all satisfying assignments of $g$ is the set

$$S(g) = \bigcup_{i \in I_g} S(t_i)$$  \hspace{1cm} (17)

where $S(t_i)$ is the satisfying assignment of the implicant $t_i$.

#### 4.1.1 Implicants of a system

For any implicant $t(X)$ the satisfying assignment $t(a) = 1$ is a partial assignment of all literals in $t(X)$ to equal 1. Hence the above expression (17) is a compact representation of all satisfying assignments of $g$. The union in (17) is disjoint because the implicants are OG hence for any $t_i = 1$, $t_j = 0$ at the same assignment for $j \neq i$.

The implicant based solver for a Boolean system of equations of [9] computes the set set of all satisfying assignments of a Boolean system in terms of a complete OG set of implicants of the system. Here an implicant of a system of Boolean functions $G = \{g_i(X)\}$, $i = 1, \ldots, m$ is meant a term $t(X)$ such that if $t(a) = 1$ for an assignment $X = a$ then $g_i(a) = 1$ for all $i = 1, \ldots, m$. Hence such a system of Boolean functions is represented by a single function $g$ in terms of a complete set $I_G$ of implicants of the system $G$

$$g(X) = \sum_{t \in I_G} t(X)$$

Then we have

$$g(X) = \prod_{i=1}^{m} g_i(X)$$

Hence the set of all satisfying assignments of the system $G$ is given by

$$S(G) = S(g) = \sum_{t \in I_G} S(t)$$  \hspace{1cm} (18)
4.2 Formulation of the problem of GOE

Consider the map $F$ in Problem 1, then GOE of $F$ consists of points $(y_1, y_2, \ldots, y_n)$ in $\mathbb{F}_2^n$ such that

$$F(x_1, \ldots, x_n) = (y_1, \ldots, y_n)^T$$

has no satisfying assignments $X$. Consider the above system of Boolean equations in variables $X$ and $Y = (y_1, \ldots, y_n)$, denoted as $F(X) \oplus Y = 0$. Let a complete set of OG implicants for satisfying assignments of this system of equations be denoted as $I(F, Y)$. Then each implicant in this set is of the form

$$t(X, Y) = t_1(X)t_2(Y)$$

where $t_1$ and $t_2$ are terms in $X$ and $Y$ respectively such that for assignments $t_1(a) = 1$, $t_2(b) = 1$, $(a, b)$ satisfies $t(a, b) = 1$ for some $t$ in the set $I(F, Y)$. Define the function

$$\phi(Y) = \sum_{t \in I(F, Y)} t_2(Y) \tag{19}$$

**Theorem 4.** GOE of $F$ is the set $S(\phi(Y)^\prime)$ or alternatively,

$$GOE = \{Y | \phi(Y) = 0\}$$

**Proof.** Since $t$ is itself an implicant of the system $F(X) \oplus Y = 0$, for an assignment $X = a, Y = b$ which satisfies this equation, $t_1(a) = t_2(b) = 1$. From this it follows that the set of all assignments of $Y$ for which there exist assignments of $X$ of the system is obtained as satisfying assignments of

$$\sum_{t \in I(F, Y)} t_2(Y) = 1$$

Hence all assignments of $Y$ for which there is no solution $X$ of the equation are equal to satisfying assignments of all the equations

$$t_2(Y) = 0, \forall t \in I(F, Y)$$

This is expressed as the set of all solutions of the the equation

$$\sum_{t \in I(F, Y)} t_2(Y) = 0$$

Defining the function $\phi(Y)$ as given the result follows. \qed

Such a set of assignments $Y$ can be thus computed by using the implicant algorithm again on the function $\phi(Y)^\prime$. 21
4.2.1 An example of computation of GOE using the Boolean solver algorithm

Consider the map

\[ F(x_1, x_2, x_3) = [x_1 + x_2 x_3 + x_1 x_2 x_3, x_1 + x_2, x_2 + x_1 x_3]^T \]

The system \( F(X) \oplus Y = 0 \) above in this case is

\[
\begin{align*}
g_1 &= x_1 + x_2 x_3 + x_1 x_2 x_3 + y_1 = 0 \\
g_2 &= x_1 + x_2 + y_2 = 0 \\
g_3 &= x_2 + x_1 x_3 + y_3 = 0
\end{align*}
\]

which we shall denote as \( S(X,Y) \). Define

\[ g(X,Y) = (g_1(X,Y) \oplus 1)(g_2(X,Y) \oplus 1)(g_3(X,Y) \oplus 1) \]

Then a complete set of implicants of \( g(X,Y) \) is a complete set of implicants of the system \( S(X,Y) \). We find this set using the Boolean system solver algorithm of [9]. Using the ON set \( I = \{ x_1, x'_1 x_2, x'_1 x'_2 \} = \{ t_i, i = 1, 2, 3 \} \) we find implicants of the function \( g(X,Y) \).

Pivot function \( g \)

\[ g_2 \oplus 1 = 1 \oplus x_1 \oplus x_2 + y_2 \]

Hence \( I(g_2 \oplus 1) = \{ x_1 x'_2 y_2, x_1 x_2 y'_2, x'_1 x_2 y_2, x'_1 x'_2 y'_2 \} \) denoted as \( \{ t_i, i = 1, \ldots, 4 \} \)

next we use this implicant set

Pivot function \( g \)

\[ g_1 \oplus 1 = 1 \oplus x_1 \oplus x_2 x_3 + x_1 x_2 x_3 + y_1 \]

Hence

\[ I((g_1 \oplus 1)(g_2 \oplus 1)) = \{ x_1 x'_2 y_1 y_2, x_1 x_2 y_1 y'_2, x'_1 x_2 y_2 x_3 y_1, x'_1 x_2 y_2 x'_3 y_1, x'_2 x'_2 y_1 y'_2 \} \]

This set has five implicants \( \{ t_i, i = 1, \ldots, 5 \} \). We next take the final pivot to find complete implicant set of the function \( g \) or the system \( S(X,Y) \).

Pivot function \( g \)

\[ g_3 \oplus 1 = 1 \oplus x_2 \oplus x_1 x_3 + y_3 \]

Hence the complete set of implicants of the system \( F(X) = Y \) is

\[
\{ x_1 x'_2 y_1 y_2 x'_3 y_1, x_1 x_2 y_1 y'_2 x_3 y_1, x'_1 x_2 x_3 y_1 y_2 x_3 y_3, x_1 x'_2 y_1 y_2 x_3 y_3, x_1 x_2 y_1 y'_2 x'_3 y_3, x'_1 x_2 x'_3 y'_1 y_2 y'_3 \}
\]

Expressing these as products \( \{ t_{i1}(X), t_{i2}(Y) \} \) the factors affecting assignments of \( Y \) for which there exist solution \( X \) are

\[ T(Y) = \{ y_1 y_2 y_3, y_1 y'_2 y'_3, y_1 y_2 y_3, y_1 y'_2 y_3, y'_1 y_2 y_3, y'_1 y'_2 y'_3 \} \]
These are six minterms in $y_1, y_2, y_3$. Hence all points in GOE are given by solutions of

$$\phi(Y) = \sum_i t_{12}(Y) = 0$$

Since all minterms in $Y$ add up to 1 the above equation is equivalent to the following equation in terms of minterms not present in the above set $T(Y)$,

$$y_1'y_2y_3' + y_1'y_2'y_3 = 1$$

On simplifying, this equation is equivalent to

$$y_1'(y_2 \oplus y_3) = 1$$

Hence $y_1 = 0$ along with pairs $(y_2 = 0, y_3 = 1), (y_2 = 1, y_3 = 0)$. Hence GOE of $F$ consists of two points

$$\text{GOE} = \{(0,0,1), (0,1,0)\}$$

To verify correctness of the GOE computed above we can see actual trajectories of the map $F$ as follows

- orbit of length 1: $(0,0,0) \rightarrow (0,0,0)$
- orbit of length 4: $(1,0,0) \rightarrow (1,1,0) \rightarrow (1,0,1) \rightarrow (1,1,1) \rightarrow (1,0,0)$
- chain of length 2: $(0,1,0) \rightarrow (0,1,1) \rightarrow (1,1,1)$
- chain of length 1: $(0,0,1) \rightarrow (0,0,0)$

Hence the actual trajectories show that GOE consists of the predicted two points $(0,1,0), (0,0,1)$.

### 5 Applications to Cryptanalysis

In this section we briefly discuss application of the local inversion algorithm we have developed in previous sections to illustrate Cryptanalysis of block and stream ciphers.

#### 5.1 Cryptanalysis of block ciphers

A block cipher is a map $E : \mathbb{F}_2^{n+m} \rightarrow \mathbb{F}_2^p$ where $n$ is the length in bits of the symmetric key $K$, $m$ is the length of the input plaintext block $P$ and $p$ is the length of the output ciphertext $C$. $E$ is called an encryption function and $C$ the encryption of $P$. The map $E$ satisfies the properties:

1. Computation of $C$ for given $P$ and $K$ is efficiently feasible.

2. Computation of $P$ given $C$ and $K$ is efficiently feasible.

3. For any fixed $K$, $E(K, X) : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^p$ is a practical OWF. Alternatively this condition is same as, given a pair $(P, C)$ such that $C = E(K, P)$ for some $K$ it is practically infeasible to find any bit of $K$. 

23
The Cryptanalysis problem in its general sense is usually understood as recovery of $K$ given $(P,C)$ pair of blocks. Such a problem is practically justified because same key $K$ is used in encryption of many plaintext blocks $P$. Some of these blocks are known a-priori. Hence if a likely block $P$ is fixed the problem is to compute $K$ given $C = E(K,P)$. Hence this is a local inversion problem of $F(X) = E(X,P)$ when $n = m = p$ which we shall address here.

5.1.1 Algorithm for cryptanalysis

The theory of local inversion developed in previous sections leads to Algorithm 3 for cryptanalysis of block ciphers of above form with $n = m = p$.

| Algorithm 4 Cryptanalysis of block ciphers $C = E(K,P)$ |
|---------------------------------------------------------|
| 1: **procedure** BLOCKCIPHERCRYPTANALYSIS($C = E(K,P)$) |
| 2: **Input**: The algorithm $E$ and one pair $(P,C)$. |
| 3: **Output**: All solutions $K$ such that $C = E(K,P)$. |
| 4: **Offline Computation**: |
| 5: Compute the set $\Pi$ of all possible periods of closed orbits of the map $F(X) = E(X,P)$. |
| 6: **Offline Computation**: |
| 7: Compute the set $\text{GOE}$ of the map $F$. |
| 8: **Online Computation**: |
| 9: Use Algorithm 1 to find the minimal polynomial of the sequence $S(F,C)$ and one solution if the sequence turns out periodic. |
| 10: **Online Computation**: |
| 11: Find all other solutions using the Algorithm 2. |
| 12: **end procedure** |

The algorithm 3 shows that once the offline computation of periods of closed orbits and $\text{GOE}$ of $F(X) = E(X,P)$ is carried out a-priori, the online computation of all $K$ for a given $C$ can be achieved only by forward iterative action by $F$ which is computationally very efficient.

5.2 Cryptanalysis of stream ciphers

Stream ciphers come with an iteration map (a dynamical system) with an output function. A general model of such an algorithm is

$$
\begin{align*}
    x(k+1) &= F(x(k)) \\
    w(k) &= f(x(k))
\end{align*}
$$

$k = 0, 1, 2, \ldots$. The initial condition $x(0) = (K, IV)$ where $K$ is symmetric key. The output stream is $w(k)$ is used for encryption of the plaintext stream of bits $p(k)$ as

$$
c(k) = p(k) \oplus w(k)
$$

Hence $w(k)$ is called keystream. When some of the bits $p(k)$ of plaintext at instances $k$ are known, then the keystream bits $w(k)$ are also known.
5.2.1 Cryptanalysis problems

Cryptanalysis of the stream cipher (20), (21) consists of two problems

1. Problem 1. (Internal state recovery). Given a partial key stream \( w(k) \) over some interval \( k_0, k_0 + 1, \ldots, k_0 + t \), find all possible internal states \( x(k_0) \).

2. Problem 2. (Key recovery from internal state). Given an internal state \( x(k_0) \) at some \( k_0 \) find all the initial states \( x(0) \). This leads to key recovery by matching the known IV in the \( x(0) \).

We show how each of these can be formulated as local inversion problems of maps.

5.2.2 Internal state recovery

Let \( F^* \) denote the dual linear map on the space of functions \( V \) on \( \mathbb{F}_2^n \) for the map \( F \). For a keystream \( w(k_0), \ldots, w(k_0 + n - 1) \), of length \( n \), the internal state \( x(k_0) \) is related by the equations

\[
w(k_0 + j) = (F^*)^j f(x(k_0))
\]  

(22)

Hence when all the \( n \) samples of \( w(k_0 + j) \) for \( j = 0, \ldots, (n - 1) \) denoted as \( \hat{w}(k_0) \) are given the equation (22) defines the map

\[
\hat{F}(x(k_0)) = [f, (F^*)^1 f, \ldots, (F^*)^{(n-1)} f]^T(x(k_0))
\]  

(23)

Whose local inversion at \( \hat{w}(k_0) \) gives solutions of internal states. Due to the special structure of the map \( \hat{F} \) the evaluation of the sequence \( S(\hat{F}, \hat{w}(k_0)) \) can be carried out by repeated forward action of the stream cipher map \( \hat{F} \) itself. Hence online computation by algorithms 1 and 2 is as easy as the stream cipher operation itself.

5.2.3 Computation of set of possible periods of iteration of \( \hat{F} \)

For local inversion of the map \( \hat{F} \) it is thus necessary to examine the computation of the set of all possible periods of iterations of \( \hat{F} \). This map \( \hat{F} \) is same as given in (23) acting on any state \( x \) in \( \mathbb{F}_2^n \). The definition of the dual map \( \hat{F}^* \) is then

\[
\hat{F}^*(\chi_1) = f
\]

\[
\hat{F}^*(\chi_2) = (F^*)^1 f
\]

\[
\vdots
\]

\[
\hat{F}^*(\chi_n) = (F^*)^{(n-1)} f
\]  

(24)

Alternatively the map \( \hat{F} : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n \) is defined as

\[
\hat{F} = [f(x), (F^* f)(x), \ldots, (F^*)^{(n-1)}(x)]^T
\]  

(25)

Hence it follows that for any function \( \phi(x_1, \ldots, x_n) \)

\[
\hat{F}^* \phi = \phi(f, F^* f, \ldots, (F^*)^{(n-1)} f)
\]  

(26)
Definition 1. A stream cipher denoted as system \((F, f)\) is said to be observable if \(\hat{F}\) defined in (25) is a permutation of \(\mathbb{F}_2^n\).

Proposition 4. The stream cipher defined by the system \((F, f)\) is observable iff all co-ordinate functions belong to the space of functions

\[
S(F, f) = \text{span} \{ f, F^* f, (F^*)^2 f, \ldots, (F^*)^{(n-1)} f \}
\]

(27)

Proof. Let the stream cipher \((F, f)\) be observable which by definition implies that \(\hat{F}\) is a permutation of \(\mathbb{F}_2^n\). Let \(\hat{G}\) be the inverse of \(\hat{F}\) and let \(\hat{G}^*\) denote its dual linear map on functions as defined in (26). By definition of inverse it follows that

\[
\hat{G} \circ \hat{F} = \text{Id}_{\mathbb{F}_2^n} \iff \hat{F}^* \hat{G}^* = \text{Id}_\mathcal{V}
\]

By the expression of \(\hat{F}^*\) in (24) it follows that

\[
\chi_i = \hat{G}^* (F^*)^{(i-1)} f
\]

for all \(i = 1, \ldots, n\). Hence all co-ordinate functions belong to the space of functions \(S(F, f)\).

Conversely, if all co-ordinate functions \(\chi_i\) belong to \(S(F, f)\), then the map \(\hat{F}\) is surjective on \(\mathbb{F}_2^n\) hence for an output stream \(w(k_0 + j)\) of length \(n\)-there is a unique \(x = x(k_0)\) such that equations (22) hold. This implies observability of \((F, f)\). \(\square\)

We thus get the following observation about the set of all possible periods of closed orbits of iterations of the map \(F\).

Theorem 5. If the stream cipher system \((F, f)\) is observable, then

1. \(S(F, f)\) is the smallest \(\hat{F}^*\)-invariant subspace which contains all co-ordinate functions as well as the function \(f\).

2. All periods of closed orbits of iterations of \(F\) are contained in the periods of closed orbits of iterations of the linear map

\[
\hat{F}_1 = \hat{F}^*|S(F, f)
\]

whose matrix representation in a basis of \(S(F, f)\) is denoted \(K\). Hence all possible periods of closed orbits are contained in \(\Pi\) the set of all periods of closed orbits of \(K\)

3. Every output stream \(w(k)\) of length \(n\) has a unique solution to the internal state.

Proof. Let a function \(g\) be in \(S(F, f)\). Then by definition of \(S(F, f)\) in (27), \(g\) has an expression

\[
g = \sum_{j=0}^{n-1} \alpha_j (F^*)^j f
\]
for some constants \(\alpha_j\) in \(F_2\). Now \(\hat{F}^*g\) is

\[ g(\hat{F}\chi_1, \ldots, \hat{F}\chi_n) \]

Hence using expressions \(^{24}\) and expression of \(g\) itself as linear combination of generators of \(S(F, f)\) it follows that \(\hat{F}^*g\) belongs \(S(F, f)\). Which proves that \(S(F, f)\) is \(\hat{F}^*-\)invariant. It is smallest such space being cyclically generated and contains both \(f\) and all co-ordinate functions.

Being a permutation, all trajectories of iteration of \(\hat{F}\) are closed orbits. Let \(\dim S(F, f) = N\) then \(K\) is an \(N \times N\) matrix and we can consider trajectories of iteration of \(K\) in \(F_2^N\). From Proposition 2 it follows that \(K\) is nonsingular hence trajectories of iteration of \(K\) are also closed orbits. From Theorem 2 it follows that the set of all possible periods of orbits of \(\hat{F}\) are contained in the set of periods \(\hat{\Pi}\) of closed orbits of \(K\).

Unique internal state corresponds to an output stream of length \(n\) since \(\hat{F}\) is a permutation.

The offline algorithm for computation of all possible periods of \(\hat{F}\) follows from above theorem. We can thus state the solution of Problem 1 of cryptanalysis of stream ciphers as in Algorithm 4 below. Assume that \((F, f)\) is observable.

**Algorithm 5 Cryptanalysis of stream ciphers \((F, f)\)**

1: **procedure** STREAMCIPHERCRYPTANALYSIS\((w(k_0 + k), k = 0, \ldots, n)\)
2: **Input:** The algorithm \((F, f)\) and sequence \(\hat{w}\) of length \(n\).
3: **Output:** Unique internal state \(x(k_0)\).
4: **Offline Computation:**
5: Compute the set \(\hat{\Pi}\) of all possible periods of closed orbits of the map \(\hat{F}\) as the set \(\hat{\Pi}\) of periods of closed orbits of the linear map \(K\).
6: **Online Computation:**
7: Search for the minimal polynomial of the sequence \(S(\hat{F}, \hat{w})\). Verify that the order of the polynomial belongs to \(\hat{\Pi}\).
8: Find the unique solution \(x(k_0)\) using formula in equation \(^{10}\)

\[ x(k_0) = (1/\alpha_0)[\hat{F}^{(m-1)}(\hat{w}) + \sum_{i=1}^{(m-1)} \alpha_i \hat{F}^{(i-1)}(\hat{w})] \]

9: The computation of \(\hat{F}^{(k)}(w)\) is defined by single step \(\hat{F}(w)\) as

\[ \hat{F}(w) = [f(w), f(F(w)), \ldots, f(F^{(n-1)}(w))]^T \]

10: **end procedure**

Note that as \((F, f)\) is assumed observable the map \(\hat{F}\) is a permutation. Hence the GOE is empty. The online computation only involves forward computation by the map \(\hat{F}\) which is in turn obtained via forward computation of the stream cipher \((F, f)\) which is very efficiently possible. The expression of \(x(k_0)\) can
also be alternatively obtained by linear representation of the inverse map \( \hat{G} \) computed from \( \hat{F}_1 \) as shown in [7]. Yet another approach to recovering the internal state using observer theory is announced in [8].

5.2.4 Case of unobservable \((F, f)\)

In the general case of the stream cipher system \((F, f)\), the map \( \hat{F} \) is not a permutation. Hence local inversion of the equation

\[
\hat{F}(x) = \hat{w}
\]  

(28)

to recover all internal states \( x \) requires computation of the GOE of \( \hat{F} \) in addition to the period of \( S(\hat{F}, \hat{w}) \). Although this additional requirement (in offline) makes the computation of all internal states more complex in the unobservable case, this approach of computing one solution in a periodic orbit of \( S(\hat{F}, \hat{w}) \) online by Algorithm 1 and all other solutions using the chains starting from GOE by Algorithm 2 is expected to be far faster than direct solution of the Boolean system \([28]\). Hence at the cost of doing an offline computation using the Boolean solver for GOE the online computation is made much efficient.

5.2.5 Key recovery from internal state: solution to Problem 2

This problem has the fastest solution in the case when the map \( F \) of the system is a permutation. Suppose \( F \) is a permutation. Then by locating the inverse of the internal state \( x(k_0) \) in the periodic orbit \( S(F, x(k_0)) \) the unique inverse can be found by repeated forward computation of \( F \). Repeating this process till one gets \( x(0) \) recovers the initial condition. Hence the offline computation involves only the computation of the set of all periods of closed orbits of iteration of \( F \) by linear representation of \( F \).

When \( F \) is not a permutation, an expression of \( x(k_0) \) when \( k_0 \) is a-priori known, can be obtained by offline computation as

\[
F^{(k_0)}(x_0) = x(k_0)
\]

This is again a local inversion problem of the map \( F^{(k_0)} \). This problem can be solved by repeated forward computations on \( x(k_0) \) and the GOE of the map as shown in Algorithms 1 and 2. Hence the offline computation consists of finding the linear representation of \( F^{k_0} \) and set of all periods of closed orbits of its iteration. By linear representation of \( F \), as the restriction of the dual linear map \( F^* \) on the \( F^* \)-invariant subspace \( W \), let \( K \) denotes the matrix representation. Then the linear representation of \( F^{(k_0)} \) is obtained as the matrix \( K^{k_0} \). Unfortunately computation of GOE does not allow such a simplification in terms of \( K \).

Appendix

We show here how the checking of the condition \( F^N(y) = y \) is efficiently feasible in Algorithm 3. Consider the map \( F \) in \( \mathbb{F}_2^n \) and assume that an evaluation \( F(x) \)
at any point $x$ is efficiently possible. We have following rules for composition of
$F$ with itself ($F \circ F)(x) = F(F(x))$ for non-negative numbers $a, b$.

\[
\begin{align*}
F^{(a)}(x) &\triangleq (F \circ \ldots \circ F)(x), a \text{ times. By definition} \\
F^{(a+b)}(x) &= (F^{(a)} \circ F^{(b)})(x) \\
F^{(ab)}(x) &= (F^{(a)} \circ \ldots \circ F^{(a)})(x), b \text{ times.}
\end{align*}
\]

From these it follows that for the repeated squaring under composition we have

\[
F^{(2^i)}(x) = (F^{(2^{i-1})} \circ F^{(2^{i-1})})(x)
\]

Hence using this recursive rule $F^{(2^i)}(x)$ can be computed in $i$ compositions.

Now to compute $F^{(N)}(x)$ for an exponential $N$, we can write binary expansion

\[
N = n_0 + n_1 \cdot 2 + n_2 \cdot 2^2 + \ldots + n_i \cdot 2^{(i-1)}
\]

for $N$ of log $N = i$. Then by above rules of composition,

\[
F^N(x) = F^{(n_0)} \circ F^{(n_1 \cdot 2)} \circ \ldots \circ F^{(n_i \cdot 2^{(i-1)})}(x)
\]

These are $i$ compositions and evaluations of functions. But each composition
above can be achieved in at most $i$ compositions and evaluations. Since basic
operation of evaluation $F(x)$ is efficiently feasible all of the above individual
evaluations, the computation of $F^N(x)$ is feasible in polynomial number of eval-
uations in log $N$.

6 Conclusions

A complete algorithm is constructed for local inversion of a map $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ at
$y$ in $\mathbb{F}_2^n$ which is equivalent to an algorithm for computation of all solutions
$x$ of the equation $F(X) = y$ over $\mathbb{F}_2^n$. The algorithm is constructed in two parts
one which depends only on $F$ called Offline computation and another which
depends on $y$ and a repeated forward action of $F$ called online computation.
Offline computation consists of computation of two sets associated with $F$, the
set of all possible periods of closed orbits of iterations of $F$ and the GOE of
$F$. The set of closed orbits of the a linear representation of $F$ is shown to
contain all possible periods of closed orbits of $F$. It is shown that if there
exists a solution in a periodic orbit $S(F, y)$ then this solution is unique in any
periodic orbit and is efficiently obtained by forward computation by $F$ once the
minimal polynomial of the orbit is computed. Finally other solutions of the
inversion lying on chains are discovered by computing the chains by the forward
map $F$ starting from the points in GOE. These computations are inherently
parallel for each point in GOE. The most important practical impact of this
paper for Cryptanalysis is that, one solution of the inverse is feasible efficiently
for an online data when the linear complexity of $S(F, y)$ is small enough to
facilitate practically feasible computation even if data of offline computation is
Offline computation is required to compute all other solutions. Hence any encryption algorithm in order to be secure must have high linear complexity of all of its periodic orbits. Since the offline computation can be efficiently done by parallel computation, an estimation of complete security of cipher algorithms by inversion is practically feasible and can be carried out offline by large scale parallel computation.

Acknowledgements

Author is thankful to Shashank Sule for his suggestions which led him to the investigation of the local inversion problem. He is also thankful to Ramachandran for pointing out important omissions in the first draft of the paper.

References

[1] Arthur Gill. Linear Sequential Circuits, Analysis, Synthesis and Applications. McGraw-Hill Book Company, New York, 1966.

[2] Martin Hellman. A cryptanalytic time-memory trade-off. IEEE Trans. Information Theory, 26(4), pp.401-406, 1980.

[3] J. Hong and P. Sarkar. New Applications of Time Memory Data Tradeoff. B. Roy (Ed) ASIACRYPT, 2005. LNCS 3788, pp.353-372.

[4] Howard M. Hays. Distributed Time Memory Tradeoff Attacks on Ciphers. http://eprint.iacr.org/2018/123.

[5] S. Gangopadhyay. The Time-memory Trade-off Attack. ACM Winter School 2019 on Cybersecurity, December 2019. National Institute of Science Education and Research, Bhubaneswar.

[6] Ramamchanran Anantharaman and Virendra Sule. Koopman operator approach for computing structure of solutions and Observability of non-linear dynamical systems over finite fields. Mathematics of Control, Signals and Systems, 2021. DOI 10.1007/s00498-021-00286-y

[7] Ramachandra A. and Virendra Sule. On computation of the inverse of a polynomial map over finite fields using the reduced Koopman dual linear map. arXiv.org/cs.SY/2010.14601.

[8] Ramachandra A. and Virendra Sule. Observability attack on Stream Generators. eprint.iacr.org/2021/126.

[9] Virendra Sule. An implicant based, parallel, all solution solver for Boolean satisfiability. arXiv.org/1611.09590v3, Feb 6, 2017.

[10] A. Cafure, G. Matera and A. Waissbein. Inverting bijective polynomial maps over finite fields. IEEE Information Theory Workshop - ITW ’06, Punta del Este, 2006.
[11] Nicolas T. Courtois, Gregory Bard, David Wagner. Algebraic and Slide Attacks on Keeloq. Proc. Fast Software Encryption 2008. LNCS 5086, pp.97-115.

[12] Yanbin Zheng, Qiang Wang and Wenhong Wei. On Inverses of Permutation Polynomials of Small Degree Over Finite Fields. IEEE Transactions on Information Theory. vol. 66, pp.914-922, 2020.

[13] Gary Mullen, Daniel Panario (Ed.). Handbook of Finite Fields. Taylor and Francis, 2013.

[14] Lidl R. and Neiderreiter H. Finite Fields. Cambridge University Press, 1997.

[15] Rainer Ruppel. Linear Complexity and Random Sequences. F. Pichler (Ed), Advances in Cryptology-EUROCRYPT’85, LNCS 219, pp.167-188, Springer-Verlag, Berlin Heidelberg 1986.

[16] Alfred Menezes, Paul van Oorschot, and Scott Vanstone (Ed.). Handbook of Applied Cryptography. CRC Press, 1996.

[17] Joachim von zur Gathen. Modern Computer Algebra. Cambridge University Press, 2013.