The ISC Int'l Journal of Information Security

July 2017, Volume 9, Number 2 (pp. 101–110)

http://www.isecure-journal.org

On the Computational Complexity of Finding a Minimal Basis for the Guess and Determine Attack

Shahram Khazaei [1], and Farokhlagha Moazami [2,*]

[1] *Sharif University of Technology, Department of Mathematical Sciences, Iran, Tehran*

[2] *Shahid Beheshti University, Cyberspace Research Institute, Iran, Tehran*

ARTICLE INFO.

Article history:
Received: 6 March 2017
Revised: 13 June 2017
Accepted: 10 July 2017
Published Online: 12 July 2017

Keywords:
Guess-and-determine Attack, Computational Complexity, NP-complete, Fixed Parameter Tractable, Uniquely Restricted Matching, Alternating Cycle Free Matching, Perfect Matching, Jump Number, Forcing Number.

ABSTRACT

Guess-and-determine attack is one of the general attacks on stream ciphers. It is a common cryptanalysis tool for evaluating security of stream ciphers. The effectiveness of this attack is based on the number of unknown bits which will be guessed by the attacker to break the cryptosystem. In this work, we present a relation between the minimum number of the guessed bits and uniquely restricted matching of a graph. This leads us to see that finding the minimum number of the guessed bits is NP-complete. Although fixed parameter tractability of the problem in term of minimum number of the guessed bits remains an open question, we provide some related results. Moreover, we introduce some closely related graph concepts and problems including alternating cycle free matching, jump number and forcing number of a perfect matching.

1 Introduction

Guess-and-determine (GD) attack is a general cryptanalysis technique for evaluating security of symmetric cryptosystems. In particular, GD attacks have been a favourite tool for evaluating strength of many prominent stream ciphers during the past decade. Examples include, the attacks on RC4 in [1], on A5/1 in [2], on Snow family in [3–7] on Sober family in [8–13], on Sosemanuk in [14–16], on Polar Bear in [17, 18] and on Rabbit in [19].

Any cryptosystem can essentially be described using a system of nonlinear equations. In many cases, the system is an over-defined set of equations over some finite field. In particular, many ciphers proposed during the last decades have been designed to be efficient on modern w-bit processors where typically $w = 8, 16, 32, 64$. These ciphers can be described using some equations over $\mathbb{F}_{2^w}$. GD attack is a basic divide-and-conquer approach to solve the corresponding equation system. The attacker guesses the values of some unknowns, and then recovers the remaining variables using the available relations. If the guessed values are incorrect, the attacker reaches a contradiction. However, if he is lucky and the guessed values are correct, the remaining unknowns will be determined. To mount the attack, all possible values for the set of guessed variables, known as *basis*, must be tried. Thus, the effectiveness of the attack is determined by the size of the basis.

In this paper, we consider a class of GD attacks on a system of m equations over $\mathbb{F}_q$ involving n unknown variables where $m = \text{poly}(n)$ and $q = O(1)$. We assume that each equation in this system has the following property: if the equation depends on, let us say, t variables and $t-1$ of them are known, then the last

* Corresponding author.
Email addresses: shahram.khazaei@sharif.ir (S. Khazaei), f_moazemi@sbu.ac.ir (F. Moazami)

ISSN: 2008-2045

variable is uniquely and efficiently determined. This is a typical property of equations derived for many ciphers in practice. Assume that a basis of size k exists and the attacker knows one. To mount the actual attack, the attacker then needs to try all q^k possible ways that the unknowns can be assigned. Since the correctness of each assignment can be efficiently verified, one can find all satisfying solutions essentially in time $O(q^k)$, ignoring polynomial factors.

The naive approach to find a basis of size k, if one exists, is to examine all $\binom{n}{k}$ subsets of size k of the set of all unknowns. Checking if a given subset is a basis can be done in polynomial time. Therefore, this leads to the time complexity $O(n^k \text{poly}(n))$ for finding a basis of size k.

Finding a basis using the naive approach is already unfeasible even for moderate values such as $n = 64$ and $k = 10$ since $n^k = 2^{60}$ is considered hardly affordable for cryptanalysis purposes. Therefore, GD attacks have often been designed ad-hoc based on the experience of cryptanalysts to find a basis of a small size. There are also some research [5–7] trying to find a (non-optimal) basis in a more systematic way using some heuristics based on greedy algorithms.

1.1 Motivation and Contribution

Although finding an optimal basis, i.e., a basis with minimum size, has been understood to be a difficult problem by cryptanalysts, to the best of our knowledge, no attempt has been devoted to understand the computational complexity aspects of this problem. In this paper, we consider a rigorous treatment of the problem of finding a minimum-size basis and show that it is NP-complete. We also study the fixed parameter tractability of the minimal basis problem in terms of the size of optimal basis size k; that is, if it has an $O(f(k)\text{poly}(n))$-algorithm. This is very important both from a theoretical complexity point of view and practical cryptanalysis since the naive approach has a lower bound complexity $\Omega(n^k)$. Although we are not able to prove the fixed parameter tractability of the problem in terms of k, we show that it is fixed parameter tractable in terms of the treewidth of a graph which we associate to the problem.

The remainder of this paper is organized as follows. In Section 2, we introduce some graph concepts that are used in the next sections. In Section 3, we give a precise definition of finding a minimal basis for GD attack and present an equivalent description of the problem in graph theory terminology. Then in Section 4, we study computational complexity of the finding a minimal basis. Section 5 relates our problem to two other mathematical problems from graph theory. In Section 6, we discuss some aspects of the GD attack which leads to improved cryptanalytic attacks in practice. Finally, the paper is concluded in Section 7.

2 Graph Preliminaries

We assume that the reader is familiar with basic graph terminologies. Here we introduce some more advance notions. Appendix A includes some illustrating examples. An *isolated* vertex is a vertex with degree zero. An *empty graph* has no edges, i.e., it consists of only isolated vertices. A *pendant* vertex is a vertex with degree one. A *matching* in a graph is a set of edges where no two edges share a common vertex. A vertex is *matched* (or *saturated*) if it is an endpoint of one of the edges in the matching. A matching of a graph is called *perfect* if all vertices of the graph are matched.

Definition 2.1 (uniquely restricted matching). A matching M in a graph G is called a *uniquely restricted matching* if its matched vertices induce a subgraph which has a unique (perfect) matching, namely M itself (see Figure A.1a and A.1b).

Definition 2.2 (alternating cycle with respect to a matching). Let M be a matching in graph G. A cycle in G is called alternating with respect to M if its edges appear alternately in M and $E(G)setminusM$ (see Figure A.1c).

Definition 2.3 (alternating cycle free matching). A matching M in a graph G is called alternating cycle free if G has no alternating cycle with respect to M.

Definition 2.4 (tree decomposition of a graph). A tree decomposition of a graph G is a pair $(T, \mathcal{B})$, where T is a tree and $\mathcal{B} = \{B_i\}_{i \in V(T)}$ is a collection of subsets of V called bags, such that (see Figure A.2):

- Every vertex of G is contained in at least one bag B_i,
- For each edge of G, some bag B_i contains both its vertices,
- For all vertices $i, j, k \in V(T)$, if j belongs to the unique path from i to k in T, then $B_i \cap B_k \subset B_j$.

Definition 2.5 (treewidth of a graph). The width of a tree decomposition $(T, \mathcal{B})$ is one less than its maximum bag size, i.e., $\max_{i \in V(T)} |B_i| - 1$. The treewidth of a graph G is defined as the minimum width over all possible tree decompositions of G.

See [20, 21] for more details about treewidth and tree decomposition.

3 Minimal Basis Problem

In this section, we provide a formal definition of the problem of finding a minimum-size basis of an

equation set with some special properties. We also present an equivalent description of the problem in graph theory terminology which is easier to work with. We need to introduce some definitions.

Definition 3.1 (invertible equation). An equation $f(x_1, \ldots, x_t) = 0$ over $\mathbb{F}_q$ is called invertible if any variable of the equation is uniquely determined in time $O(t \log q)$ when all other variables are known. A preprocessing phase that computes a table of size $O(q)$ in time $O(q)$ is allowed.

See Appendix B for a typical invertible equation that appears during cryptanalysis of symmetric cryptosystems.

Definition 3.2 (invertible equations system). Let $X = \{x_1, \ldots, x_n\}$ be a set of n variables and $F = \{f_1, \ldots, f_m\}$ be a set of m invertible equations, each depending on a subset of X. We call (F, X) an invertible equations system.

Definition 3.3 (basis of an invertible equations system). Let (F, X) be an invertible equations system. A subset $B \subsetneq X$ is called a basis for the system if at the end of Procedure 1 we have $U = X$.

Procedure 1 *(used in Definition 3.3)*

> **Input:** An invertible equations system (F, X) and a subset $B \subsetneq X$
> $U \leftarrow B$ **while** *there is an equation* $f \in F$ *depending on* t *variables with* $t-1$ *of them in* U **do**
> Let $x \notin U$ be the (remaining) variable that f depends on Add x to U
> **end**

Note that any subset $B \subset X$ of size $|B| = |X|-1$ is a basis; also a basis might be empty. Therefore, for a basis B we have $0 \leq |B| \leq |X|-1$. If B is a basis for (F, X), any assignment to the variables in B can efficiently be checked in terms of consistency. If the whole set of equations are consistent given the assigned values, the remaining variables are uniquely determined. Otherwise, a contradiction is reached.

Below, we present a toy example as well as a real example of a set of invertible equations.

Example 3.1 (real). Appendix C describes the set of equations that fully specifies the Snow 2.0. [4, 22] stream cipher.

Example 3.2 (toy). The following set of invertible equations

$$\begin{cases} f_1(x_1, x_2, x_3) = 0 \\ f_2(x_2, x_3, x_4) = 0 \\ f_3(x_1, x_2, x_4) = 0 \\ f_4(x_1, x_3, x_4) = 0 \\ f_5(x_1, x_2, x_3, x_4) = 0 \end{cases}$$

has no basis of size one, but has some bases of size two, e.g., $B = \{x_2, x_3\}$.

Our goal is to study the problem of finding a minimal basis of an invertible equations system. Instead, we will work with an equivalent description of the problem in terms of bipartite graphs. We first define the basis of such graphs.

Definition 3.4 (basis of a bipartite graph). Let $G = (F, X, E)$ be a bipartite graph. A subset $B \subsetneq X$ is called a basis of G if at the end of Procedure 2, G becomes empty:

Procedure 2 *(used in Definition 3.4)*

> **Input:** A bipartite graph $G = (F, X, E)$ and a subset $B \subsetneq X$
> Remove every edge in G which is incident to some vertex in B **while** *there is an edge* $fx \in E$ *such that* $f \in F$, $x \in X$ *and* $\deg(f) = 1$ **do**
> Remove every edge in G which is incident to x
> **end**

We now formally introduce the graph version of minimal basis problem, that we will work with in the rest of this paper.

Definition 3.5 (minimal basis problem). The minimal basis problem is defined with the following input instance and question:

- **Instance:** A connected bipartite graph $G = (F, X, E)$ and an integer k
- **Question:** Does G have a basis of size at most k?

We ignore to provide a variant of the above definition for an invertible equations system. The relation between the two versions of the problem of finding a minimal basis should be clear. Let $F = \{f_1, f_2, \ldots, f_m\}$ be a set of invertible equations depending on variables $X = \{x_1, x_2, \cdots, x_n\}$. We associate a bipartite graph $G = (F, X, E)$ with vertex set $F \cup X$ to the equation set as follows: the equation $f_i \in F$ is adjacent to the variable $x_j \in X$, i.e., $f_i x_j \in E$, if and only if f_i effectively depends on x_j. Clearly, a set $B \subsetneq X$ is a basis for the invertible equations system (F, X) if and only if it is a basis for the associated graph G. Notice that the associated graph has no isolated vertex.

4 Computational Complexity of Minimal Basis Problem

4.1 NP-completeness

The following theorem presents a close connection between bases and uniquely restricted matchings of a bipartite graph.

Theorem 1. *Let $G = (F, X, E)$ be a non-empty bipartite graph with no isolated vertex and let $0 \leq k \leq |X|-1$ be an integer. The graph G has a basis of size k if and only if it has a uniquely restricted matching of size $|X|-k$.*

In the following we propose and prove two lemmas from which Theorem 1 is concluded.

Lemma 1. *Let $G = (F, X, E)$ be a non-empty bipartite graph with no isolated vertex and let $B \subsetneq X$ be a basis of size k for G, where $0 \leq k \leq |X|-1$. Then G has a uniquely restricted matching of size $|X|-k$.*

Proof. Similar to the first step of Procedure 2, remove every edge in G which is incident to some vertex in B. Denote the remaining subgraph by H_0. The graph H_0 is non-empty and it has at least one pendant vertex $f_1 \in F$. Assume that $x_1 \in X$ is the only vertex incident to f_1 and let $e_1 = f_1x_1$. Let H_1 be a graph obtained from H_0 by deleting every edge in H_0 which is incident to x_1. Since B is a basis for the graph G, then (assuming $k \leq |X|-2$) there exists at least one pendant vertex $f_2 \in F$ in H_1. Let $x_2 \in X$ be the only vertex incident with f_2 and let $e_2 = f_2x_2$. Repeat the previous procedure until an empty graph is obtained. Note that since B is a basis of size k for the graph G, this procedure stops after $|X|-k$ rounds. In the sequel, we show that $M = \{e_1, e_2, \ldots, e_{|X|-k}\}$ is a uniquely restricted matching for G. Assume that K is the induced subgraph of the graph G by the vertices $\{f_1, x_1, f_2, x_2, \cdots, f_{|X|-k}, x_{|X|-k}\}$. The degree of the vertex f_1 in the graph K is one. Therefore, if K has a matching M' of size $|X|-k$, then all vertices of the graph K must be matched. The only edge which is incident to the vertex f_1 is e_1. Hence, $e_1 \in M'$. Clearly, the degree of f_2 in the subgraph $K \setminus \{f_1, x_1\}$ is one. Therefore, if K has a matching M' of size $|X|-k$, then $e_2 \in M'$. By repeating this procedure, we conclude that $M' = M$ and, therefore, M is a uniquely restricted matching. □

Lemma 2. *Let $G = (F, X, E)$ be a bipartite graph and let $M \subsetneq E$ be a uniquely restricted matching of size $|X|-k$ for G, where $0 \leq k \leq |X|-1$. Then, $B = X \setminus V(M)$ is a basis of size k for G.*

Proof. Clearly, $B \subsetneq X$ since $|M| \geq 1$. We show that at the end of Procedure 2, on input G and B, the graph G becomes empty. The algorithm first removes every edge in G which is incident to some vertex in B. Denote the remaining subgraph by H_0. Since $|M| \geq 1$, the graph H_0 is non-empty. We show that there exist some edge $fx \in E(H_0)$ such that $f \in F$, $x \in X$ and $\deg(f) = 1$.

We use a contrapositive argument. Let $M = \{f_1x_1, \ldots, f_\ell x_\ell\}$, where $\ell = |X|-k$, and all vertices $\{f_1, \ldots, f_\ell\}$ have degree at least 2. Notice that M is the unique matching of H_0 of size ℓ. Let K be the subgraph of H_0 induced by the vertices of the set $V(M) = \{x_1, \ldots, x_\ell, f_1, \ldots, f_\ell\}$. Since all vertices in the set $\{f_1, \ldots, f_\ell\}$ have degree at least 2, then K has a cycle C. Since K is a bipartite graph, the number of edges of the cycle C is an even number. Also, C has some edges of the matching M alternatively. Thus, C is an alternating cycle with respect to a submatching of M. This is a contradiction since M is a uniquely restricted matching. Hence, K has a pendant vertex $f_1 \in F$. Degree of f_1 in the graph H_0 is also one since every vertex of the graph H_0 in the section X is a vertex of the graph K.

Let $x_1 \in X \setminus B$ be the only adjacent vertex to f_1 in H_0. Remove all edges incident to x_1 in H_0 and let H_1 denote the resulting graph.

If $|M| = 1$, then $E(H_1) = \emptyset$ and hence B is a basis as required. Otherwise we have $E(H_1) \neq \emptyset$. Let $M_1 = M \setminus \{f_1x_1\}$. Clearly, M_1 is the only matching of size $\ell - 1$ for the graph H_1. Again, we argue that H_1 has a pendent vertex $f_2 \in F$. By repeating the previous procedure $|M|$ times, we finally obtain an empty graph. Therefore, B is a basis for the graph G. □

In [23], it has been shown that maximum uniquely restricted matching problem is NP-complete for bipartite graphs. Therefore, we have the following corollary.

Corollary 1. *The minimal basis problem is NP-complete.*

4.2 Fixed Parameter Tractability

A naive algorithm to find a basis of size k for a bipartite graph $G = (F, X, E)$, if one exists, is to examine all $\binom{n}{k}$ subsets $B \subset X$ of size k. Checking if a given subset is a basis can be done in time $O(|E|+|X|)$ using Procedure 2. Therefore, this leads to the time complexity $O(n^k \text{poly}(n))$ for finding a basis of size k, where $|X| = n$ and $|F| = \text{poly}(n)$. This section discusses the fixed parameter tractability of the minimal basis problem. Unfortunately, the fixed parameter tractability of the problem in terms of the size of optimal basis size k—that is, if an $O(f(k)\text{poly}(n))$-time algorithm solves the problem [1] —remains unanswered.

[1] As an example for a fixed parameter tractable problem on graphs, we mention the *vertex cover* problem. A vertex cover of a graph $G = (V, E)$ is a subset of vertices $S \subseteq V$ that includes at least one endpoint of every edge in E. There is a simple algorithm that finds a vertex cover of size k, if any exists, in time $O(2^k|E|)$: pick an edge $uv \in E$ and recursively check if

Therefore, we draw our attention to studying the fixed parameter tractability of the problem in terms of the graph treewidth.

In [23], it has been shown that a matching of a bipartite graph is uniquely restricted if and only if it is alternating cycle free. Therefore, the problem of finding a maximum alternating cycle free matching in bipartite graphs is polynomially equivalent to the problem of finding a minimum-size basis. The Courcelle's celebrated theorem [24] (also see [25]) can be used to show that finding a maximum alternating cycle free matching is fixed parameter tractable in terms of the graph treewidth. Also notice that the problem of computing treewidth is fixed parameter tractable [26]. Therefore, we have the following corollary.

Corollary 4.1. Finding a minimal basis of a bipartite graph is fixed parameter tractable with respect to the graph treewidth.

We remark that as it has been noticed in [25], the stated complexity by Courcelle's theorem contains towers of exponents in the treewidth parameter, making it impractical and a purely theoretical result. However, in [25], a significantly faster algorithm with running time $O(4^{w^2+w} \cdot w^3 \cdot \log(w) \cdot n)$ has been proposed for finding a maximum alternating cycle free matching of a bipartite graph G, where n is the number of vertices of G and w denotes the width of the tree decomposition.

5 Problems Related to Minimal Basis Problem

In this section, we draw reader's attention to two mathematical problems related to minimal basis problem.

5.1 Jump Number

We introduce a parameter for a bipartite graph which is related to its minimum-size basis. In the sequel, we give the necessary notations and definitions.

Recall that a *partially ordered set* or a *poset* is a pair $P = (X, \leq)$ where X is a set and $\leq$ is a reflexive and transitive binary relation. A pair of elements $a, b \in X$ are called comparable if $a \leq b$ or $b \leq a$; otherwise they are called incomparable. We write $a < b$ if $a \leq b$ and $a \neq b$. A sequence $a_1, \ldots, a_s$ is called a *chain* of P if $a_1 \leq \cdots \leq a_s$. A poset without incomparable elements is called a *linear* or *total order*. Let $P = (X, \leq)$ be a poset on a finite set X. A *linear extension* of P is a total ordering $L = (X, \preceq)$ where the relation between comparable elements of P is preserved in L; that is, if $a \leq b$, for $a, b \in X$, then $a \preceq b$. It then follows that $x_1, \ldots, x_n$ is a chain of L where $X = \{x_1, \ldots, x_n\}$. A pair of consecutive elements (x_i, x_{i+1}) is a *jump* of L if x_i and x_{i+1} are incomparable in P. In fact, the jumps partition the chain $x_1, \ldots, x_n$ of L into disjoint chains $C_1, C_2, \ldots, C_m$ of P such that the maximum element of C_i is incomparable with the minimum element of C_{i+1}. The *jump number of a partial order* P is the minimum number of jumps taken over all linear extensions of P.

A (loopless undirected) graph $G = (X, E)$ is called a *comparability graph*, if it is possible to find a poset $P = (X, \leq)$, called a poset of G, such that for each distinct $a, b \in X$ if $ab \in E$ then a and b are comparable in P. It is well-known that a graph is a comparability graph if it is *transitively oriented*; that is, if it can be turned into a directed graph by orienting each edge such that for every vertices x, y, z, it holds that if xy and yz are oriented edges then so is xz. The *jump number of a comparability graph* is defined as the jump number of any of its posets. From the results of Habib [27] and Möhring [28] the jump number of a comparability graph is well-defined since it is invariant with respect to the underlying poset.

Orienting the edges of a bipartite graph from one side of the bipartition to the other clearly results in a transitive orientation. Thus, every bipartite graph is a comparability graph. A result from Chaty and Chein [29] shows that for a bipartite graph $G = (F, X, E)$ with a maximum alternating cycle free matching M we have

$$|M| + j(G) = |F| + |X| - 1 \ ,$$

where $j(G)$ is the graph's jump number. Therefore, we have the following theorem:

Theorem 2. *Finding the jump number of a bipartite graph is polynomially equivalent to the problem of finding the maximum alternating cycle free matching of G and it is itself equivalent to finding the minimum-size basis of G.*

5.2 Smallest Forcing Number

This subsection establishes a relationship between a basis of a bipartite graph and another parameter defined for graphs. To continue, we need some terminology that follows.

Let G be a graph that admits a perfect matching. A *forcing set of a perfect matching* M in G is a subset S of M contained in no other perfect matchings of G. The *forcing number of a perfect matching* M of G is defined as

$$f(G, M) = \min\{|S| \colon S \text{ is a forcing set of } M\}.$$

The *forcing number* of G is defined as

$$f(G) = \min\{f(G, M) : M \text{ is a perfect matching of } G\}.$$

$G \setminus u$ or $G \setminus v$ has a vertex cover of size $k - 1$.

Remark 1. Motivated by applications in chemistry, the concept of forcing number was first introduced in [30]. Later, the forcing number of a perfect matching was studied in [31] as a mathematical concept. See also ([32]) for a survey. The computational complexity of finding forcing number of perfect matching and graph are studied in the literature, see ([33, 34]).

Remark 2. Forcing set can be considered for other mathematical parameters other than perfect matchings. In fact, this concept has a general definition as follows. Let $(\mathcal{F}, S)$ be a pair such that $\mathcal{F}$ is a family of sets and $S \in \mathcal{F}$. A set $D \subseteq S$ is a *forcing* set, also known as *defining* set, of $(\mathcal{F}, S)$ if S is the only element of $\mathcal{F}$ that contains D as a subset. This concept has been studied in numerous cases, such as vertex colorings, perfect matchings, dominating sets, block designs, geodetics, orientations, and Latin squares [35].

We can prove the following relation between the forcing number and the minimum basis size of a graph G.

Theorem 3. *Let G be a bipartite graph that admits a perfect matching. Let $b(G)$ be the size of the minimal basis of the graph G. Then, $b(G) = f(G)$.*

Proof. Let $G = (F, X, E)$ be a bipartite graph and M be a perfect matching of G that $f(G) = f(G, M)$. Let S be a forcing set of the matching M that $|S| = f(G, M)$. Then by definition of the forcing set, $M \setminus S$ is the only perfect matching of the graph $G \setminus S$. In other words, $M \setminus S$ is a uniquely restricted matching of the graph G.

By Lemma 2, the set $B = X \setminus V(M \setminus S) = V(S) \cap X$ is a basis of the graph G. Also notice that $|B| = |S|$ and $b(G) \leq |B|$, since $b(G)$ is the size of a minimum basis. Consequently, $b(G) \leq |S| = f(G, M) = f(G)$. Conversely, let B be a basis of the graph G of size $b(G)$. Then by Lemma 1, the graph G has a uniquely restricted matching M of size $|X| - b(G)$. Consider the subgraph H of the graph G induced by the vertices $F \cup (X \setminus (V(M) \cap X))$. Since G admits a perfect matching and H is an induced subgraph of the graph G, H has a matching M' such that every vertex of $(X \setminus (V(M) \cap X))$ is a saturated vertex in M'. So, $M' \cup M$ is a perfect matching of the graph G. Since M is a uniquely restricted matching then B is a forcing set of the matching $M' \cup M$. Therefore, $f(G) \leq f(G, M) \leq |B| = b(G)$. □

6 Improved Guess and Determine Attacks

In this section, we discuss some points with respect to the effectiveness of guess and determine attack. This might lead to improved algorithms for finding a satisfying solution.

6.1 Better Guessing Strategy

Suppose that an invertible equations system (F, X) over $\mathbb{F}_q$ has a minimal basis of size k. We have implicitly assumed that an attacker first guesses the basis and then computes the rest of the variables. This of course requires a computation of order $O(q^k)$, ignoring polynomial factors. However, this might not necessarily be the best strategy. Bellow, we consider two such cases.

1. Separable equations. One trivial example is when the system is separable. More precisely, assume that X and F can respectively be partitioned into (X_1, X_2) and (F_1, F_2) such that (F_1, X_1) and (F_2, X_2) both have unique solutions. If these systems have respectively minimal bases of sizes k_1 and k_2, then the original system has a basis of size $k = k_1 + k_2$. The unique solution can then be found in time $q^{k_1} + q^{k_2}$ instead of $q^{k_1+k_2}$.

2. Early contradictions. A less trivial case is when after guessing some part of a basis an early contradiction is reached. To be concrete, assume that (F, X), with $|X| = n$, has a minimal basis of size k. Furthermore, suppose that by guessing only $k' < k$ elements of the basis, m other variables are subsequently uniquely determined where $m < n$. Assume that there is an equation in F that only depends on the determined m variables and it is algebraically independent of all the equations that have so far been used to determine these m unknowns. This "check" equation can be used to reduce the initial $q^{k'}$ possibilities by a factor of q. Therefore, the overall attack complexity is reduced by a factor of q, i.e., q^{k-1} instead of q^k. In [14, 16] this idea has been used to mount a faster attack on the Sosemanuk [36] stream cipher.

6.2 Working With an Equivalent Equations System

In some cases working with modified equations system might lead to an improved attack. Bellow we present three examples.

1. Linear equations system. When (F, X) is a linear system over $\mathbb{F}_q$, say with unique solution, then the satisfying solution can be found using Gaussian elimination in polynomial time.

2. Using redundant equations. In some cases adding redundant equations to the equation system might lead to a faster attack. One case is when attacking a stream cipher based on LFSRs (Linear Feed-

back Shift Registers). The output of an LFSR is determined by its feedback polynomial. The feedback polynomial imposes a linear constraint on the output sequence of the LFSR. Redundant constraints can be obtained by considering any multiple of the feedback polynomial. See [37] for an example or refer to the last paragraph of Appendix C for further discussion.

3. Changing the underlying finite field. In some cases the equations can be seen over a smaller finite field. As an example, Sosemanuk [36] stream cipher works with 32-bit words. In contrary to the attack in [14, 16] which solves a system of equations over $\mathbb{F}_{2^{32}}$, the authors of [38] take a byte-based approach and solve a system of equations over $\mathbb{F}_{2^8}$. Consequently, the overall attack time is dramatically reduced form 2^{226} to 2^{176}.

7 Conclusion

In this paper, we showed that finding the minimum number of the guessed bits in the guess-and-determine attack is equivalent to finding the maximum uniquely restricted matching in a bipartite graph. Using this observation we studied the computational complexity aspect of this attack. Also, we introduced the relation between this problem and some other mathematical problems such as jump number and forcing number of a perfect matching. Exploring the fixed parameter tractability of the problem in terms of the minimum-basis size remains an open question. An interesting research problem is to study the approximability aspects of the minimal basis problem.

Acknowledgment

The first author has been supported by Iranian National Science Foundation (INSF) under contract no. 92027548 and Sharif Industrial Relation Office (SIRO) under grant no. G931223. The authors would like to thank the anonymous reviewers for their constructive and valuable comments that greatly contributed to improve the quality and presentation of the paper.

References

[1] Lars R. Knudsen, Willi Meier, Bart Preneel, Vincent Rijmen, and Sven Verdoolaege. Analysis Methods for (Alleged) RC4. In *ASIACRYPT*, pages 327–341, 1998.

[2] Jovan Dj. Golic. Cryptanalysis of Alleged A5 Stream Cipher. In *EUROCRYPT*, pages 239–255, 1997.

[3] Christophe De Cannière. Guess and determine attack on SNOW. *NESSIE Public Document, NES/DOC/KUL/WP5/011/a*, 2001.

[4] Philip Hawkes and Gregory G. Rose. Guess-and-Determine Attacks on SNOW. In *Selected Areas in Cryptography*, pages 37–46, 2002.

[5] Azad Mohammadi-Chambolbol. Cryptanalysis of word-oriented stream ciphers. Master thesis (in Persian), Sharif University of Technology., 2004.

[6] Hadi Ahmadi and Taraneh Eghlidos. Advanced Guess and Determine Attacks on Stream Ciphers. *International Symposium on Telecommunications (IST), Tehran, Iran*, 2005.

[7] Hadi Ahmadi and Taraneh Eghlidos. Heuristic guess-and-determine attacks on stream ciphers. *IET Information Security*, 3(2):66–73, 2009.

[8] Philip Hawkes. An attack on SOBER-II. *Technical report, QUALCOMM Australia, Suite 410, Birkenhead Point, Drummoyne NSW 2137, Australia*, 1999.

[9] S Blackburn, S Murphy, F Piper, and P Wild. A SOBERing remark. *Unpublished report. Information Security Group, Royal Holloway University of London, Egham, Surrey TW20 0EX, UK*, 1998.

[10] Daniel Bleichenbacher, Sarvar Patel, and Willi Meier. Analysis of the SOBER stream cipher. *TIA contribution TR45. AHAG/99.08*, 30, 1999.

[11] Daniel Bleichenbacher and Sarvar Patel. SOBER Crytanalysis. In *FSE*, pages 305–316, 1999.

[12] Christophe De Cannière. Guess and determine attack on SOBER. *NESSIE Public Document, NES/DOC/KUL/WP5/010/a*, 2001.

[13] Steve Babbage, Christophe De Cannière, Joseph Lano, Bart Preneel, and Joos Vandewalle. Cryptanalysis of SOBER-t32. In *FSE*, pages 111–128, 2003.

[14] Hadi Ahmadi, Taraneh Eghlidos, and Shahram Khazaei. Improved guess and determine attack on SOSEMANUK. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/085, 2005. http://www.ecrypt.eu.org/stream/papersdir/085.pdf.

[15] Yukiyasu Tsunoo, Teruo Saito, Maki Shigeri, Tomoyasu Suzaki, Hadi Ahmadi, Taraneh Eghlidos, and Shahram Khazaei. Evaluation of SOSEMANUK with regard to guess-and-determine attacks. *SASC (the State of the Art of Stream Ciphers)*, pages 25–34, 2006.

[16] Xiutao Feng, Jun Liu, Zhaocun Zhou, Chuankun Wu, and Dengguo Feng. A Byte-Based Guess and Determine Attack on SOSEMANUK. In *ASIACRYPT*, pages 146–157, 2010.

[17] John Mattsson. A Guess-and-Determine Attack on the Stream Cipher Polar Bear. eSTREAM, ECRYPT Stream Cipher Project, Report 2006/017, 2006. http://www.ecrypt.eu.org/stream/papersdir/2006/017.pdf.

[18] Mahdi Hasanzadeh, Elham Shakour, and Shahram Khazaei. Improved Cryptanalysis of Po-

lar Bear. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/084, 2005. http://www.ecrypt.eu.org/stream/papersdir/084.pdf.

[19] Xiutao Feng, Zhenqing Shi, Chuankun Wu, and Dengguo Feng. On Guess and Determine Analysis of Rabbit. *Int. J. Found. Comput. Sci.*, 22(6):1283–1296, 2011.

[20] Hans L. Bodlaender and Arie M. C. A. Koster. Treewidth computations i. upper bounds. *Inf. Comput.*, 208(3):259–275, 2010.

[21] Ton Kloks. *Treewidth, Computations and Approximations*, volume 842 of *Lecture Notes in Computer Science*. Springer, 1994.

[22] Patrik Ekdahl and Thomas Johansson. A New Version of the Stream Cipher SNOW. In *Selected Areas in Cryptography*, pages 47–61, 2002.

[23] Martin Charles Golumbic, Tirza Hirst, and Moshe Lewenstein. Uniquely restricted matchings. *Algorithmica*, 31(2):139–154, 2001.

[24] Bruno Courcelle. The monadic second-order logic of graphs: Definable sets of finite graphs. In *Graph-Theoretic Concepts in Computer Science, 14th International Workshop, WG '88, Amsterdam, The Netherlands, June 15-17, 1988, Proceedings*, pages 30–53, 1988.

[25] Benjamin A. Burton, Thomas Lewiner, João Paixão, and Jonathan Spreer. Parameterized complexity of discrete morse theory. *ACM Trans. Math. Softw.*, 42(1):6:1–6:24, 2016.

[26] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[27] Michel Habib. Comparability invariants. In *Orders: description and roles (L'Arbresle, 1982)*, volume 99 of *North-Holland Math. Stud.*, pages 371–385. North-Holland, Amsterdam, 1984.

[28] Rolf H. Möhring. Algorithmic aspects of comparability graphs and interval graphs. In *Graphs and order (Banff, Alta., 1984)*, volume 147 of *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, pages 41–101. Reidel, Dordrecht, 1985.

[29] G. Chaty and M. Chein. Ordered matching and matchings without alternating cycles in bipartite graphs. *Utilitas Math.*, 16:183–187, 1979.

[30] D. J. Klein and M. Randic. Innate degree of freedom of a graph. *J. Comput. Chem. 8*, pages 516–521, 1987.

[31] F. Harary, D. J. Klein, and T. P. Zivkovic. Graphical properties of polyhexes: perfect matching vector and forcing. *J. Math. Chem. 6*, pages 295–306, 1991.

[32] Z. Che and Z. Chen. Forcing on perfect matchings a survey. *MATCH Commun. Math. Comput. Chem. 66*, pages 93–136, 2011.

[33] Peter Adams, Mohammad Mahdian, and Ebadollah S. Mahmoodian. On the forced matching numbers of bipartite graphs. *Discrete Mathematics*, 281(1-3):1–12, 2004.

[34] P. Afshani, H. Hatami, and E. S. Mahmoodian. On the spectrum of the forced matching number of graphs. *Australasian J. Combin., 30*, pages 147–160, 2004.

[35] Diane Donovan, ES Mahmoodian, Colin Ramsay, and Anne Penfold Street. Defining sets in combinatorics: a survey. *Surveys in combinatorics*, pages 115–174, 2003.

[36] Côme Berbain, Olivier Billet, Anne Canteaut, Nicolas Courtois, Henri Gilbert, Louis Goubin, Aline Gouget, Louis Granboulan, Cédric Lauradoux, Marine Minier, Thomas Pornin, and Hervé Sibert. Sosemanuk, a Fast Software-Oriented Stream Cipher. In *The eSTREAM Finalists*, pages 98–118. 2008.

[37] Mohammad Sadegh Nemati Nia and Ali Payandeh. THE NEW HEURISTIC GUESS AND DETERMINE ATTACK ON SNOW 2.0 STREAM CIPHER. *IACR Cryptology ePrint Archive*, 2014:619, 2014.

[38] Xiutao Feng, Jun Liu, Zhaocun Zhou, Chuankun Wu, and Dengguo Feng. A byte-based guess and determine attack on SOSEMANUK. In *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, pages 146–157, 2010.

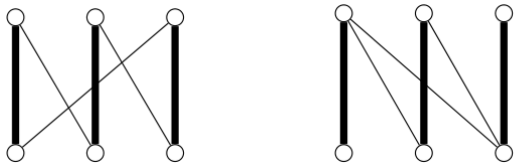
[39] Patrik Ekdahl and Thomas Johansson. SNOW-a new stream cipher. In *Proceedings of First Open NESSIE Workshop, KU-Leuven*, 2000.

[40] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

Shahram Khazaei is an assistant professor at the Department of Mathematical Sciences at Sharif University of Technology, Iran, since 2012. He received his Ph.D. in computer science from EPFL, Switzerland, in 2010 and was a postdoctoral researcher at KTH Royal Institute of Technology, Sweden, from 2011 to 2012. His main research interests is theoretical and practical aspects of cryptography.

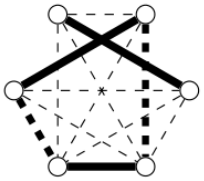
Farokhlagha Moazami is an assistant professor at the Cyber Space Research Institute at Shahid Beheshti University, Iran, Tehran, since 2013. She received B.S. and Ph.D. degrees in mathematics from Alzahra University, Tehran, Iran, in 2004 and 2012, respectively and M.S. degree in mathematics from Sharif University of Technology, Iran, Tehran, in 2006. She was a postdoctoral at Sharif University of Technology, Iran, Tehran, from 2012 to 2013. Her main research interests is theoretical and practical aspects of cryptography.

A Some Illusterating Graphs Examples



(a) a non-uniquely restricted matching ([23]).

(b) a uniquely restricted matching ([23]).



(c) an alternating cycle (of length 4) with respect to a (perfect) matching.

Figure A.1. Matching examples

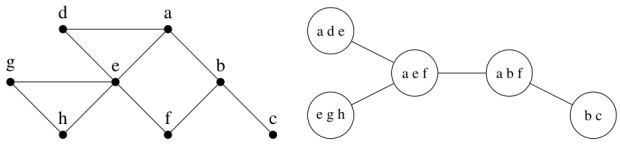


Figure A.2. A graph with a tree decomposition ([20]).

B Example of an Invertible Equation

In this section we give an example of a typical invertible equation that appears in analysis of cryposystems. First we introduce some notations.

Let p be the characteristic of $\mathbb{F}_q$ and denote the field addition and multiplication operations respectively by "$\oplus$" and "$\cdot$" where multiplication is normally dropped. Let $(\alpha_0, \cdots, \alpha_{w-1})$ be a fixed basis for the field extension $\mathbb{F}_q$ over $\mathbb{F}_p$.

The binary operation "+" from $\mathbb{F}_q \times \mathbb{F}_q$ to $\mathbb{F}_q$ is defined as follows:

$$x + y \mapsto z \ ,$$

where $x = \sum_{i=0}^{w-1} x_i\alpha_i$, $y = \sum_{i=0}^{w-1} y_i\alpha_i$, $z = \sum_{i=0}^{w-1} z_i\alpha_i$ with $x_i, y_i, z_i \in \mathbb{F}_p$ and

$$\sum_{i=0}^{w-1} x_i p^i + \sum_{i=0}^{w-1} y_i p^i = \sum_{i=0}^{w-1} z_i p^i \qquad \bmod p^w \ ,$$

where x_i, y_i, z_i are interpreted as elements of $\mathbb{Z}_p$.

The first three $\sum$'s are additions over $\mathbb{F}_q$ and the last three are over integers. The "$-$" operation can be defined similarly.

The binary operation "$\lll$" from $\mathbb{F}_q \times \{0, 1, \ldots, w-1\}$ to $\mathbb{F}_q$ is defined as follows:

$$x \lll r \mapsto y \ ,$$

where x and y are represented as above and $y_i = x_{i-r \bmod w}$. The binary operation $\ggg$ can be defined similarly as $y_i = x_{i+r \bmod w}$ where $x \ggg r \mapsto y$.

Example B.1. Let $r_1, r_2 \in \{0, 1, \ldots, w-1\}$ and $\beta_1, \beta_2, \beta_3 \in \mathbb{F}_q$ be some constants where $\beta_1, \beta_2 \neq 0$. Let $\sigma : \mathbb{F}_q \to \mathbb{F}_q$ be a permutation. Then,

$$f(x, y, z) = (\beta_1(x \ggg r_1) \oplus \beta_2 y) + \sigma(z \lll r_2) + \beta_3 = 0$$

is an invertible equation since we have

$$x = \left(\beta_1^{-1}(\beta_2 y \oplus (0 - \sigma(z \lll r_2) - \beta_3))\right) \lll r_1 \ ,$$

$$y = \beta_2^{-1}(\beta_1(x \ggg r_1) \oplus (0 - \sigma(z \lll r_2) - \beta_3)) \ ,$$

$$z = \sigma^{-1}(0 - (\beta_1(x \ggg r_1) \oplus \beta_2 y) - \beta_3) \ggg r_2 \ .$$

After computing the inverse permutation σ^{-1} in time $O(q)$ and saving it in a memory of size $O(q)$, the three inversions can be computed in $O(\log q)$.

C Equations Describing Snow 2.0. Stream Cipher

The Snow 2.0. [22] stream cipher is an updated version of Snow [39]. Figure C.1 shows a schematic picture of Snow 2.0. It is composed of a Linear Feedback Shift Register (LFSR) and a Finite State Machine (FSM). The LFSR is defined with the following primitive feedback polynomial

$$\pi(x) = \alpha x^{16} \oplus x^{14} \oplus \alpha^{-1} x^5 \oplus 1 \in \mathbb{F}_{2^{32}}[x],$$

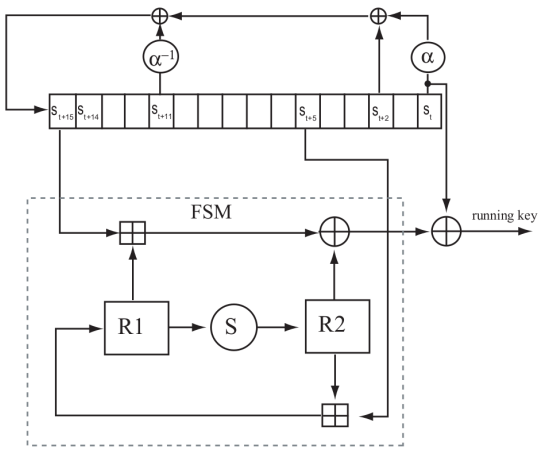


Figure C.1. A schematic of Snow 2.0. stream cipher [22]

where α is some known constant and $\oplus$ is the finite field addition (or bit-wise XOR operation on 32-bit registers). The initial state of the LFSR is denoted by $(s_{15}, s_{14}, \ldots, s_0) \in \mathbb{F}_{2^{32}}^{16}$. Therefore, the output sequence of the LFSR, $\{s_t\}_{t\geq 0}$, is determined according to the following recursion:

$$s_{t+16} = \alpha^{-1} s_{t+11} \oplus s_{t+2} \oplus \alpha s_t.$$

The FSM has two 32-bit registers, denoted by $R1$ and $R2$, and the value of the registers at time $t \geq 0$ are respectively denoted by $R1_t, R2_t \in \mathbb{F}_{2^{32}}$.

The output of the FSM at time $t \geq 0$, denoted by F_t, is then computed as follows from the initial values of the registers, $R1_0$ and $R2_0$, and the output sequence of the LFSR:

$$F_t = (s_{t+15} + R1_t) \oplus R2_t.$$

Here, $+$ denotes the modulo 2^{32} addition on 32-bit registers (see also Appendix B). The values of the registers are updated according to

$$R1_{t+1} = s_{t+5} + R2_t,$$
$$R2_{t+1} = S(R1_t).$$

where $t \geq 0$ and $S : \mathbb{F}_{2^{32}} \to \mathbb{F}_{2^{32}}$ is a known permutation based on round function of AES [40].

Finally, the output sequence of the stream cipher, also called the running key or keystream, is denoted by $\{z_t\}_{t\geq 0}$ and computed as follows:

$$z_t = F_t \oplus s_t, \ t \geq 0$$

In an initial state recovery attack, an attacker is given a piece of keystream, say $\{z_t\}_{t=0}^{N-1}$, and his goal is to find the unknown initial state $(s_{15}, s_{14}, \ldots, s_0, R1_0, R2_0) \in \mathbb{F}_{2^{32}}^{18}$. When $N \geq 18$, the initial state is almost uniquely determined. Clearly, all the involved equations are invertible. In [7], it is claimed that for a certain amount of N (unspecified in [7], but probably $50 - 70$), there exists a basis of size 8 for the corresponding equations system. This reduces the search from $2^{18\times 32} = 2^{576}$ to $2^{8\times 32} = 2^{256}$. However, an observation of [37] shows that one can do better by utilizing redundant equations such as those imposed on the output sequence of the LFSR by multiples of the feedback polynomial. In particular, the authors of [37] claim that, by incorporating equations imposed on $\{s_t\}_{t\geq 0}$ by $(x^2 + 1)\pi(x)$ and $(x^5 + 1)\pi(x)$, one can find a basis of size 6, which shows an attack with complexity $2^{6\times 32} = 2^{192}$. See Section 6 for further discussion.