On solving sparse algebraic equations over finite fields II

Igor Semaev

Department of Informatics, University of Bergen, Norway

Abstract. A system of algebraic equations over a finite field is called sparse if each equation depends on a small number of variables. Finding efficiently solutions to the system is an underlying hard problem in the cryptanalysis of modern ciphers. In this paper deterministic Agreeing-Gluing algorithm introduced earlier in [9] for solving such equations is studied. Its expected running time on uniformly random instances of the problem is rigorously estimated. This estimate is at present the best theoretical bound on the complexity of solving average instances of the above problem. In particular, it significantly overcomes our previous results, see [11]. In characteristic 2 we observe an exciting difference with the worst case complexity provided by SAT solving algorithms.

Keywords: sparse algebraic equations over finite fields, constraint satisfaction problem, agreeing, gluing.

1 Introduction

Let (q, l, n, m) be a quadruple of natural numbers, where q is a prime power. Then F_q denotes a finite field with q elements and $X = \{x_1, x_2, \ldots, x_n\}$ is a set of variables from F_q. By X_i, $1 \le i \le m$ we denote subsets of X of size $l_i \le l$. The system of equations

$$f_1(X_1) = 0, \ldots, f_m(X_m) = 0 \tag{1}$$

is considered, where f_i are polynomials over F_q and they only depend on variables X_i. Such equations are called l-sparse. We look for the set of all solutions in F_q to the system of equations (1). Therefore, we can only consider for f_i polynomials of degree at most $q-1$ in each variable or, in other words, the exponent in each variable is at most $q-1$. The polynomial f_i defines a mapping from the set of all l_i-tuples over F_q to F_q and vice versa any such mapping may be represented by a polynomial over F_q of degree at most $q-1$ in each variable. Obviously, the equation $f_i(X_i) = 0$ is determined by the pair (X_i, V_i), where V_i is the set of F_q-vectors in variables X_i, also called X_i-vectors, where f_i is zero. Following terminology in [9], we also call the pair (X_i, V_i) a symbol. For $q = 2$ the polynomial f_i is uniquely defined by V_i. Given f_i, the set V_i is computed with q^{l_i} trials. Solving (1) may be considered as a Constraint Satisfaction Problem, see [12], with constraints given by V_i.

In this paper deterministic Agreeing-Gluing Algorithm, introduced in [9] and aimed to find all solutions to (1), is studied and its average behavior is estimated. To this end equiprobable distribution on instances (1), each instance has the same probability, is assumed. That is, given the sequence of natural numbers m and $l_1, \ldots, l_m \leq l$, equations in (1) are generated independently. The particular equation $f_i(X_i) = 0$ is determined by the subset X_i of size l_i taken uniformly at random from the set of all possible l_i-subsets of X, that is with the probability $\binom{n}{l_i}^{-1}$, and the mapping (polynomial) f_i taken with the equal probability $q^{-q^{l_i}}$ from the set of all possible mappings to F_q defined on l_i-tuples over F_q (the set of polynomials of degree $\leq q-1$ in each of l_i variables).

In this setting the running time of the Agreeing-Gluing Algorithm is a random variable. We estimate its expected complexity. For fixed q, l and $c \geq 1$ let $\beta = \beta(\alpha)$, where $0 \leq \alpha \leq l$, be the only root to the equation

$$q^{\beta - \frac{\alpha}{l}} = qe^{g(\alpha)}(1 - \sum_{t=0}^{l} \binom{l}{t} \beta^{l-t}(1-\beta)^t(1-\frac{1}{q})^{q^t})^{c-\frac{\alpha}{l}},$$

or $\beta(\alpha) = 0$ if there is not any root for some α. Here $g(\alpha) = f(z_\alpha) - \alpha + \alpha \ln \alpha - \frac{\alpha \ln q}{l}$ and $f(z) = \ln(e^z + q^{-1} - 1) - \alpha \ln(z)$, where by z_α we denote the only positive root of the equation $\frac{\partial f}{\partial z}(z) = 0$. We realize that the above equation doesn't depend on n. We prove

Theorem 1. *Let $\frac{l_1 + l_2 + \ldots + l_m}{ln}$ tend to a constant $c \geq 1$ as n tends to ∞ while $q \geq 2$ and $l \geq 3$ are fixed. Let $r(q,l,c)$ be the maximal of the numbers*

$$\max_{0 \leq \alpha \leq l} q^{\beta(\alpha) - \frac{\alpha}{l}} \quad \textit{and} \quad 1.$$

Then the expected complexity of the Agreeing-Gluing Algorithm is

$$O((r(q,l,c) + \varepsilon)^n)$$

bit operations for any positive real number ε.

For any triple $q, l, c \geq 1$ the Theorem enables estimating the expected running time of the Agreeing-Gluing Algorithm with some mathematical software like Maple. To this end we realize that the equation $\frac{\partial f}{\partial z}(z) = 0$ is equivalent to $\frac{ze^z}{e^z + q^{-1} - 1} = \alpha$. So $\alpha = \alpha(z)$ and $\beta = \beta(z)$ are found to be functions in z and $z_\alpha = z$.

We did the computation with Maple for some of $2, l, 1$(e.g. n Boolean equations in n variables each equation depends on l variables) and show the data obtained in Table 1 along with the expected complexities of the Gluing1 and Gluing2 Algorithms from our previous work [11]. Agreeing-Gluing1 Algorithm is a variant of the Agreeing-Gluing Algorithm with the same asymptotical running time and polynomial in n memory requirement. We have shown in [11] that in case $q = 2$ each instance of (1) may be encoded with a CNF formula in the same set of variables and of clause length at most l. Therefore, l-SAT solving algorithms provide with the worst case complexity estimates found e.g. in [5].

So in the first line we show the worst case estimates for (1) too. We remark an exciting difference in the worst case complexity and expected complexity of the Agreeing-Gluing Algorithm. It is quite obvious that average instances of the l-SAT problem and that of (1) are different. That gives insight into why the expected complexity is so low in comparison with the worst case.

Table 1. Algorithms' running time.

l	3	4	5	6
the worst case	1.324^n	1.474^n	1.569^n	1.637^n
Gluing1, expectation	1.262^n	1.355^n	1.425^n	1.479^n
Gluing2, expectation	1.238^n	1.326^n	1.393^n	1.446^n
Agreeing-Gluing1, expectation	1.113^n	1.205^n	1.276^n	1.334^n

.

In case of n Boolean equations ($q = 2$) of algebraic degree d in n variables defining a so called semi-regular system(it was conjectured, see [1], but not proved that an average instance of the problem behaves semi-regularly) the running time of the popular Gröbner Basis Algorithm and its variants, e.g. [2] and [4], is known. For $d = 2$, one gets the bound $O(1.7^n)$ by guessing a number of variables before the Gröbner Basis Algorithm (or XL) application, see [13]. For $d \geq 3$ the bounds exceed the cost of the brute force algorithm, that is 2^n. Thus the Agreeing-Gluing family algorithms seem better on sparse equation systems (1) than the Gröbner Basis related algorithms.

This article was motivated by applications in cryptanalysis. Modern ciphers are product, that is the mappings they implement are compositions of functions in small number of variables. Then intermediate variables are introduced in order to simplify equations, describing the cipher, and to get a system of sparse equations. E.g., given one pair of plain-text, cipher-text blocks, DES is described by 512 Boolean equations in 504 variables, each equation involves at most 14 variables, see [9]. For a more general type of sparse equations, Multiple Right Hand Side linear equations describing in particular AES, see [10]. Solving such systems of equations breaks the cipher. Here we are studying an approach which exploits the sparsity of equations and doesn't depend on their algebraic degree. This approach was independently discovered in [14] and [8], where the Agreeing procedure was described for the first time. The term Agreeing itself comes from [9]. The Agreeing procedure was in these works then combined with guessing some of the variables values to solve (1). That makes somewhat similar to the Agreeing-Gluing1 Algorithm of the present paper. However no asymptotical estimates for that type of algorithms were given in [14, 8, 9].

It is often important in cryptology to understand the hardness of average instances of the computational problem under question rather than its worst case complexity. So the present paper, like [11], focuses on the expected performance of the algorithms.

The rest of the paper is organized as follows. In Sections 2 Gluing procedure and the Gluing Algorithm are presented. The Agreeing procedure and the

Agreeing-Gluing Algorithm are described in Section 3. Section 4 presents the Agreeing-Gluing1 Algorithm and in Section 5 asymptotic bound on its running time, that is Theorem 1, is proved.

Before reading this paper, we recommend to look through our previous work [11], where some necessary basic facts were proved. For the reader's convenience we also formulate them in Section 2.

The author is grateful to H.Raddum for careful reading this work and numerous remarks.

2 Gluing procedure and Gluing Algorithm

We describe the Gluing procedure. Given symbols (X_i, V_i) for $i = 1, 2$, one defines two sets of variables $Z = X_1 \cup X_2$ and $Y = X_1 \cap X_2$ and a set of Z-vectors U by the rule $U = \{(a_1, b, a_2) : (a_1, b) \in V_1, (b, a_2) \in V_2\}$, where a_i is an $(X_i \setminus Y)$-vector and b is a Y-vector. We denote $(a_1, b, a_2) = (a_1, b) \circ (b, a_2)$ and say that (a_1, b, a_2) is the gluing of (a_1, b) and (b, a_2). In order to glue (X_1, V_1) and (X_2, V_2) one can sort V_1 or V_2 by Y-subvectors and only glues vectors with the same Y-subvector. So the complexity of the gluing is

$$O(|U| + (|V_1| + |V_2|) \log(|V_i|)) \tag{2}$$

operations, as rewriting and comparison, with F_q-vectors of length at most $|Z|$, where $|V_i|$ are big enough. We actually use a simpler bound $O(|V_1||V_2| + |V_1| + |V_2|)$ in what follows. We denote $(Z, U) = (X_1, V_1) \circ (X_2, V_2)$.

```
Gluing Algorithm
input: the system (1) represented by symbols $(X_i, V_i)$, where $1 \le i \le m$.
output: the set $U$ of all solutions to (1) in variables $X(m) = X_1 \cup \ldots \cup X_m$.
put $(Z, U) \leftarrow (X_1, V_1)$ and $k \leftarrow 2$,
while $k \le m$ do $(Z, U) \leftarrow (Z, U) \circ (X_k, V_k)$ and $k \leftarrow k + 1$,
return $(Z, U)$.
```

It is obvious that U is the set of all solutions to (1) in variables $X(m)$.

Example. For the system

$$\begin{array}{c|c|c} & x_1 & x_2 \\ \hline a_1 & 0 & 0 \\ a_2 & 1 & 0 \\ a_3 & 1 & 1 \end{array}, \qquad \begin{array}{c|c|c} & x_2 & x_3 \\ \hline b_1 & 0 & 0 \\ b_2 & 1 & 0 \\ b_3 & 1 & 1 \end{array}, \qquad \begin{array}{c|c|c} & x_1 & x_3 \\ \hline c_1 & 0 & 0 \\ c_2 & 0 & 1 \\ & & \end{array}$$

the Algorithm computes two gluings:

$$\begin{array}{c|c} x_1 & x_2 \\ \hline 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{array} \circ \begin{array}{c|c} x_2 & x_3 \\ \hline 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{array} = \begin{array}{c|c|c} x_1 & x_2 & x_3 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{array}, \qquad \begin{array}{c|c|c} x_1 & x_2 & x_3 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{array} \circ \begin{array}{c|c} x_1 & x_3 \\ \hline 0 & 0 \\ 0 & 1 \end{array} = \begin{array}{c|c|c} x_1 & x_2 & x_3 \\ \hline 0 & 0 & 0 \end{array}.$$

So there is only one solution $(x_1, x_2, x_3) = (0, 0, 0)$. The Gluing Algorithm takes

$$O(\sum_{k=1}^{m-1} |U_k||V_{k+1}| + |U_k| + |V_{k+1}|) = O(\sum_{k=1}^{m-1} |U_k| + m) \tag{3}$$

operations with F_q-vectors of length at most n, where q and l are fixed, and n or m may grow. The memory requirement is of the same magnitude as the running time. Here $(X(k), U_k) = (X_1, V_1) \circ \ldots \circ (X_k, V_k)$ and (3) is the cost of $m-1$ gluings. The set U_k consists of all solutions to the first k equations in variables $X(k) = X_1 \cup \ldots \cup X_k$. The sequence of $|U_k|$ fully characterizes the running time (3) of the algorithm. The asymptotical analysis of $|U_k|$ and the Gluing Algorithm is done in [11] using Random Allocations Theory results found in [7, 6, 3]. In [11] the following Theorem was proved:

Theorem 2. *Let ϵ be any positive real number, $l \geq 3$ and $q \geq 2$ be fixed natural numbers as n or m tend to infinity. Then the expected complexity of the Gluing Algorithm is $O((q^{1-\gamma_{q,l}} + \epsilon)^n + m)$ bit operations, where $\gamma_{q,l} = \frac{1}{l} + (q^{\frac{1}{l}} - 1)\log_q(\frac{1-q^{-1}}{1-q^{-\frac{1}{l}}})$.*

Two technical statements from [11] are formulated. We use them in Section 5.

Lemma 1. *(Lemma 4 in [11]) Let the subsets of variables $X_1, \ldots, X_k$ be fixed while $f_1, \ldots, f_k$ are randomly chosen according to our model. Then the expected number of solutions to the first k equations in (1) is*

$$E_{f_1,\ldots,f_k}|U_k| = q^{|X(k)|-k}.$$

Lemma 2. *(Lemma 5 in [11]) Let $L_k = l_1 + \ldots + l_k$ and $\alpha = L_k/n$, and $k \leq n$. Let $0 < \delta < 1$ be fixed as n tends to ∞. Then the expected number of solutions to the first k equations is*

$$E|U_k| = E_{X_1,\ldots,X_k}(q^{|X(k)|-k}) = \begin{cases} < q^{n^\delta}, & \text{if } L_k < n^\delta; \\ O((qe^{g(\alpha)} + \epsilon)^n), & \text{if } L_k \geq n^\delta, \end{cases}$$

for any positive real number ϵ. Here $g(\alpha) = f(z_\alpha) - \alpha + \alpha \ln \alpha - \frac{\alpha \ln q}{l}$ and $f(z) = \ln(e^z + q^{-1} - 1) - \alpha \ln(z)$, where by z_α we denote the only positive root of the equation $\frac{\partial f}{\partial z}(z) = 0$.

3 Agreeing procedure and Agreeing-Gluing Algorithm

We describe the Agreeing procedure. Given symbols (X_i, V_i) for $i = 1, 2$, one defines the set of variables $Y = X_1 \cap X_2$. Let $V_{1,2}$ be the set of Y-subvectors of V_1. In other words, that is the set of projections of V_1 to variables Y. Similarly, the set $V_{2,1}$ of Y-subvectors of V_2 is defined. We say the symbols (X_1, V_1) and (X_2, V_2) agree if $V_{1,2} = V_{2,1}$. Otherwise, we apply the procedure called agreeing. We delete from V_i all vectors whose Y-subvectors are not in $V_{2,1} \cap V_{1,2}$. Obviously, we delete

V_i-vectors which can't make part of any common solution to the equations. So new symbols (X_i, V_i') are determined, where $V_i' \subseteq V_i$ consist of the vectors in V_i survived after agreeing. We finally put $(X_i, V_i) \leftarrow (X_i, V_i')$. In order to agree (X_1, V_1) and (X_2, V_2) one sorts V_1 or V_2 by Y-subvectors and do agreeing by table look ups. So the complexity of the agreeing is at most

$$O((|V_1| + |V_2|)\log(|V_i|)) \tag{4}$$

operations, as rewriting and comparison, with F_q-vectors of length at most n, where $|V_i|$ are big enough.

For instance, we agree the first and the third equations in the example.

$$(X_1, V_1) = \begin{array}{c|c|c} & x_1 & x_2 \\ \hline a_1 & 0 & 0 \\ a_2 & 1 & 0 \\ a_3 & 1 & 1 \end{array}, \qquad (X_3, V_3) = \begin{array}{c|c|c} & x_1 & x_3 \\ \hline c_1 & 0 & 0 \\ c_2 & 0 & 1 \end{array}.$$

Then one defines $Y = \{x_1\}$ and $V_{1,3} = \{0, 1\}$, and $V_{3,1} = \{0\}$. Therefore, a_2 and a_3 should be deleted from V_1. The new agreed symbols are

$$(X_1, V_1) = \begin{array}{c|c|c} & x_1 & x_2 \\ \hline a_1 & 0 & 0 \end{array}, \qquad (X_3, V_3) = \begin{array}{c|c|c} & x_1 & x_3 \\ \hline c_1 & 0 & 0 \\ c_2 & 0 & 1 \end{array}.$$

The following Agreeing-Gluing Algorithm combines the Agreeing and Gluing procedures to solve (1).

Agreeing-Gluing Algorithm

```
  **input**: the system (1) represented by symbols $(X_i, V_i)$, where $1 \le i \le m$.
  **output**: the set $U$ of all solutions to (1) in variables $X(m) = X_1 \cup \ldots \cup X_m$.
  **put** $(Z, U) \leftarrow (X_1, V_1)$ **and** $k \leftarrow 2$,
  **while** $k \le m$ **do** $s \leftarrow k$,
      **while** $s \le m$ **agree** $(Z, U)$ **and** $(X_s, V_s)$**, put** $s \leftarrow s + 1$,
  **put** $(Z, U) \leftarrow (Z, U) \circ (X_k, V_k)$ **and** $k \leftarrow k + 1$**,**
  **return** $(Z, U)$.
```

We introduce some notation. Let $(X(1), U_1')$ be the symbol (X_1, V_1) after $m-1$ agreeings with the symbols (X_i, V_i), where $1 < i \le m$. Generally, for any $1 \le k < m$ let $(X(k+1), U_{k+1}')$ denote the symbol $(X(k), U_k') \circ (X_{k+1}, V_{k+1})$ after agreeing with $(m-k-1)$ symbols (X_i, V_i), where $k+1 < i \le m$. It is easy to see that the Agreeing-Gluing Algorithm produces the sequence of $(X(k), U_k')$. For the above example the sequence of such symbols is

$$(X(1), U_1') = \begin{array}{|c|c} x_1 & x_2 \\ \hline 0 & 0 \end{array}, \qquad (X(2), U_2') = \begin{array}{|c|c|c} x_1 & x_2 & x_3 \\ \hline 0 & 0 & 0 \end{array}, \qquad (X(3), U_3') = \begin{array}{|c|c|c} x_1 & x_2 & x_3 \\ \hline 0 & 0 & 0 \end{array},$$

where $X(1) = \{x_1, x_2\}$ and $X(2) = X(3) = \{x_1, x_2, x_3\}$. One sees that the Agreeing-Gluing Algorithm takes

$$O(\sum_{k=1}^{m-1} |U_k'||V_{k+1}| + |U_k'| + |V_{k+1}| + (m-k-1)|U_k'||V_{k+1}|) =$$

$$O(m(\textstyle\sum_{k=1}^{m-1} |U_k'| + 1)) \tag{5}$$

operations with F_q-vectors of length at most n, where q and l are fixed, and n or m may grow. The formula (5) incorporates the cost of the gluing $(X(k), U'_k) \circ (X_{k+1}, V_{k+1})$, which is $O(|U'_k||V_{k+1}| + |U'_k| + |V_{k+1}|) = O(|U'_k|)$ operations, and the agreeing the resulting set of $X(k+1)$-vectors, of size at most $|U'_k||V_{k+1}| = O(|U'_k|)$, with the rest $m-k-1$ symbols. In our setting $|U'_k|$ is a random variable. We estimate the expectation of $|U'_k|$ in Section 5, see Theorem 3. That will imply Theorem 1.

The memory requirement of the Agreeing-Gluing Algorithm is of the same magnitude as the running time, but the Agreeing-Gluing1 Algorithm in the next Section only requires polynomial memory while its asymptotical running time is the same. From the definition of Gluing and Agreeing procedures we also get the following statement.

Lemma 3. *$(X(k), U'_k)$ is the symbol $(X(k), U_k) = (X_1, V_1) \circ \ldots \circ (X_k, V_k)$ after agreeing with $(m-k)$ symbols (X_i, V_i), where $k < i \leq m$.*

4 Agreeing-Gluing1 Algorithm

The Gluing1 Algorithm in [11] has the same time complexity as the Gluing Algorithm and only requires `poly`(n) bits of memory. The Algorithm walks through a search tree with backtracking. The complexity is roughly the number of the tree branches. The search tree for the above example is depicted in Fig.1, where $a_1 \circ b_1 \circ c_1 = (0, 0, 0)$ is the only solution.

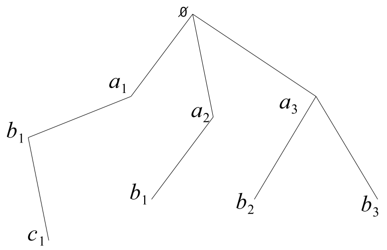

Fig. 1. The search tree for the Gluing1 Algorithm.

Similarly, the Agreeing-Gluing1 Algorithm is a variant of the Agreeing-Gluing Algorithm with minor memory requirements. In order to define the related search tree we say that an $X(k)$-vector a contradicts(does not agree) with the symbol (X_i, V_i), where $i > k$, if the projection of a to the common variables $X(k) \cap X_i$ is not in the projections of V_i to the same variables. In this case a can't be the part of any solution to the system.

A rooted search tree is now being defined. The root, labeled by $\emptyset$, is connected to nodes at level 1, labeled by elements of V_1 which do not contradict with the symbols (X_i, V_i) for all $i > 1$. Nodes at level $k \geq 2$ are labeled by some of $b \in V_k$. A node at level 1, labeled by a, is connected to a node at level 2, labeled by b, whenever the gluing $a \circ b$ is possible, that is a and b have the same sub-vector in common variables $X(1) \cap X_2$, and $a \circ b$ does not contradict with the symbols (X_i, V_i) for all $i > 2$. Generally, a sequence $a \in V_1, \ldots, b \in V_{k-1}, c \in V_k$ label a path from the root to a k-th level node if the gluing $a \circ \ldots \circ b \circ c$ is possible that is $a \circ \ldots \circ b$ and c have the same sub-vector in common variables $X(k-1) \cap X_k$ and $a \circ \ldots \circ b \circ c$ does not contradict with the symbols (X_i, V_i) for $i > k$. In this case $a \circ \ldots \circ b \circ c$ is a solution to the first k equations in (1) which does not contradict to each of the last $m - k$ equations. Orderings on V_k make the tree branches ordered. The Algorithm walks through the whole tree with backtracking. At each step the gluing d of labels from the current node to the root is computed along with the length k of the path. Then d is checked whether it contradicts to the rest of the system equations. The next step only depends on the current and previous pairs(states) d, k, so only they should be kept. The solution to the whole system (1) is the gluing of labels from any path of length m.

We estimate the complexity. Let d, k be the current state and the Algorithm extends $d \in U'_k$ with some $e \in V_{k+1}$, that is $d \circ e$ is computed and checked whether it is in contradiction with (X_i, V_i) for all $i = k+2, \ldots, m$. When d, k is the current state next time again, d is to be extended with some $e_1 < e$. This implies that the Agreeing-Gluing1 Algorithm passes through every $d \in U'_k$ at most q^l times for every k. The figure q^l may be reduced via a proper ordering of V_i but this doesn't change the asymptotic running time (5). In case of the above example the search tree is depicted in Fig.2 and favorably compared with that in Fig.1.

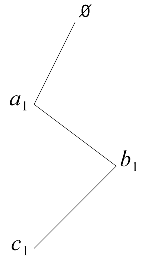

Fig. 2. The search tree for the Agreeing-Gluing1 Algorithm.

5 Complexity analysis of the Agreeing-Gluing Algorithm

In this section we prove Theorem 1. Let $Z, X_1, \ldots, X_k$ be fixed subsets of variables and U be a fixed set of Z-vectors, so that (Z, U) is defined by an equation

$f(Z) = 0$. Let V_i be the set of X_i-vectors, solutions to independent equations $f_i(X_i) = 0$ generated uniformly at random on the set of all equations in variables X_i.

Lemma 4. *Let (Z, U') be the symbol produced from (Z, U) by agreeing with (X_i, V_i) for all $1 \le i \le k$. Then the expectation of $|U'|$ is given by*

$$E_{f_1,\ldots,f_k}|U'| = |U| \prod_{i=1}^{k} (1 - (1 - \frac{1}{q})^{q^{|X_i \setminus Z|}}),$$

where $|X_i \setminus Z|$ stands for the number of variables X_i not occurring in Z.

Proof. As f_i are generated independently, it is enough to prove the Lemma for $k = 1$, then the full statement is proved by induction. Let $Y_1 = Z \cap X_1$ and $|U| = \sum_a |U_a|$, where U_a is the subset of U-vectors whose projection to variables Y_1 is a, in other words, whose Y_1-subvector is a. Then $|U'| = \sum_a |U_a| I_a$, where

$$I_a = \begin{cases} 1, \text{ if } V_{1,a} \neq \emptyset; \\ 0, \text{ if } V_{1,a} = \emptyset, \end{cases}$$

and $V_{1,a}$ is the subset of V_1-vectors whose projection to variables Y_1 is a. Let W_a be the subset of all vectors in variables X_1 whose projection to variables Y_1 is a. We see that $|W_a| = q^{|X_1 \setminus Y_1|}$. One computes

$$Pr(V_{1,a} = \emptyset) = Pr(f_1 \neq 0 \,\text{on}\, W_a) =$$

$$\frac{(q-1)^{q^{|X_1 \setminus Y_1|}} q^{q^{|X_1|} - q^{|X_1 \setminus Y_1|}}}{q^{q^{|X_1|}}} = (1 - \frac{1}{q})^{q^{|X_1 \setminus Y_1|}}.$$

So

$$E_{f_1}(I_a) = Pr(V_{1,a} \neq \emptyset) = 1 - (1 - \frac{1}{q})^{q^{|X_1 \setminus Y_1|}} = 1 - (1 - \frac{1}{q})^{q^{|X_1 \setminus Z|}}.$$

Then $E_{f_1}|U'| =$

$$\sum_a |U_a| E_{f_1}(I_a) = \sum_a |U_a| (1 - (1 - \frac{1}{q})^{q^{|X_1 \setminus Z|}}) = |U| (1 - (1 - \frac{1}{q})^{q^{|X_1 \setminus Z|}}).$$

This proves the statement for $k = 1$. Generally, as f_i are independent, the identity

$$E_{f_1,\ldots,f_k}|U'| = E_{f_1,\ldots,f_{k-1}}(E_{f_k}(|U'| \mid \{f_1, \ldots, f_{k-1}\}))$$

is easily proved, where $E_{f_k}(|U'| \mid \{f_1, \ldots, f_{k-1}\})$ denotes the expectation of $|U'|$ under $f_1, \ldots, f_{k-1}$ are fixed. This by induction implies the Lemma on the whole.

The following Corollary is obvious.

Corollary 1. *Let $Z, X_1, \ldots, X_k$ be fixed sets of variables and f be generated independently to f_i where $1 \le i \le k$. Then*

$$E_{f,f_1,\ldots,f_k}|U'| = E_f|U| \prod_{i=1}^{k}(1-(1-\frac{1}{q})^{q^{|X_i \setminus Z|}}).$$

We will use the Corollary in order to estimate the expectation of $|U'_k|$.

Lemma 5. *Let $0 \le \beta \le 1$ be any number. Then $E|U'_k| \le$*

$$q^{\beta n-k} + \sum_{|Z|>\beta n} Pr(X(k)=Z)\, q^{|Z|-k} \prod_{i=k+1}^{m} E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i \setminus Z|}}), \qquad (6)$$

where Z runs over all subsets of X of size $> \beta n$.

Proof. Let first the sets X_i be fixed, and therefore $X(k) = X_1 \cup \ldots \cup X_k$ is fixed, and all f_i are independently generated. Then by Lemma 3 and Corollary 1, we have

$$\begin{aligned} E_{f_1,\ldots,f_m}|U'_k| = E_{f_1,\ldots,f_k}|U_k| \prod_{i=k+1}^{m}(1-(1-\frac{1}{q})^{q^{|X_i \setminus X(k)|}}) = \\ q^{|X(k)|-k} \prod_{i=k+1}^{m}(1-(1-\frac{1}{q})^{q^{|X_i \setminus X(k)|}}), \end{aligned} \qquad (7)$$

as $E_{f_1,\ldots,f_k}|U_k| = q^{|X(k)|-k}$ by Lemma 2.

We now compute the expectation of $|U'_k|$ when the sets of variables are also chosen independently at random. So $E_{f_1,\ldots,f_k}|U'_k|$ is a random variable and we can compute the expectation of $E_{f_1,\ldots,f_m}(|U'_k|)$ under the fixed $X(k)$ with (7). So

$$\begin{aligned} E|U'_k| = E_{X_1,\ldots,X_m}(E_{f_1,\ldots,f_m}|U'_k|) = \\ \sum_{Z \subseteq X} Pr(X(k)=Z)\, E_{X_1,\ldots,X_m}(E_{f_1,\ldots,f_m}(|U'_k|) \mid X(k)=Z) = \\ \sum_{Z \subseteq X} Pr(X(k)=Z)\, E_{X_1,\ldots,X_m}(q^{|Z|-k} \prod_{i=k+1}^{m}(1-(1-\frac{1}{q})^{q^{|X_i \setminus Z|}})) = \\ \sum_{Z \subseteq X} Pr(X(k)=Z)\, q^{|Z|-k} E_{X_{k+1},\ldots,X_m}(\prod_{i=k+1}^{m}(1-(1-\frac{1}{q})^{q^{|X_i \setminus Z|}})) = \\ \sum_{Z \subseteq X} Pr(X(k)=Z)\, q^{|Z|-k} \prod_{i=k+1}^{m} E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i \setminus Z|}}) \end{aligned}$$

because X_i are independent. Then we partition the last sum:

$$E|U'_k| = \sum_{|Z| \le \beta n} \ldots + \sum_{|Z|>\beta n} \ldots$$

So $E|U'_k| \leq \sum_{|Z|\leq\beta n} Pr(X(k)=Z)q^{\beta n-k}+$

$$\sum_{|Z|>\beta n} Pr(X(k)=Z)\, q^{|Z|-k} \prod_{i=k+1}^{m} E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}}).$$

Therefore, $E|U'_k| \leq$

$$q^{\beta n-k}+\sum_{|Z|>\beta n} Pr(X(k)=Z)\, q^{|Z|-k} \prod_{i=k+1}^{m} E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}}).$$

This proves the Lemma.

In next three Lemmas we estimate the expectation

$$E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}}). \tag{8}$$

Lemma 6. *Let $Z \subseteq X$ be a fixed subset of variables. Then the expectation* (8) *only depends on the size of Z and doesn't depend on the set itself. The expectation is not decreasing as $|Z|$ is decreasing or $|X_i|$ is increasing.*

Proof. It is obvious that the expectation only depends on the size of Z. Then, it is not decreasing as $|Z|$ is decreasing. Whenever $|A| \leq |B|$ for two subsets A and B of X, we have

$$E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus A|}}) \geq E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus B|}}). \tag{9}$$

In order to see this, we realize that as these two values do not depend on the sets A and B, but on their cardinalities, we can assume that $A \subseteq B$. Therefore, $|X_i \setminus B| \leq |X_i \setminus A|$ for any subset X_i and so (9) follows.

Intuitively, the bigger $l_i = |X_i|$ the bigger the expectation. For a formal proof we see that the inequality for the conditional expectation

$$E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}} \mid X_i \subseteq A) \leq 1-(1-\frac{1}{q})^{q^{|A\setminus Z|}}$$

obviously holds for any subset A of some fixed size $l \geq l_i$. Then we see that

$$E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}}) = \sum_{A\subseteq X} Pr(X_0=A)\, E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}} \mid X_i \subseteq A) \leq$$
$$\sum_{A\subseteq X} Pr(X_0=A)\,(1-(1-\frac{1}{q})^{q^{|A\setminus Z|}}) = E_{X_0}(1-(1-\frac{1}{q})^{q^{|X_0\setminus Z|}}),$$

where X_0 is an uniformly random l-subset of X. The first equality follows from

$$\sum_{A\subseteq X} Pr(X_0=A)\, E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}} \mid X_i \subseteq A) =$$

$$\sum_{A\subseteq X}\frac{1}{\binom{n}{l}}\sum_{B\subseteq A}\frac{1}{\binom{l}{l_i}}(1-(1-\frac{1}{q})^{q^{|B\setminus Z|}})=\sum_{B\subseteq X}\frac{\binom{n-l_i}{l-l_i}}{\binom{n}{l}\binom{l}{l_i}}(1-(1-\frac{1}{q})^{q^{|B\setminus Z|}})=$$
$$\sum_{B\subseteq X}\frac{1}{\binom{n}{l_i}}(1-(1-\frac{1}{q})^{q^{|B\setminus Z|}})=E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}}),$$

where B runs over all subsets of X of size l_i. This proves the Lemma.

The following statement is obvious.

Lemma 7. *Let Z be a fixed u-subset of X and X_i be an l_i-subset of X taken uniformly at random. Then*

$$Pr(|X_i\setminus Z|=t)=\frac{\binom{u}{l_i-t}\binom{n-u}{t}}{\binom{n}{l_i}}.$$

From this Lemma and Lemma 6 we derive that

$$\begin{aligned}E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}})&=1-\sum_{t=0}^{l_i}\frac{\binom{u}{l_i-t}\binom{n-u}{t}}{\binom{n}{l_i}}(1-\frac{1}{q})^{q^t}\\&\leq 1-\sum_{t=0}^{l}\frac{\binom{\lfloor\beta n\rfloor}{l-t}\binom{n-\lfloor\beta n\rfloor}{t}}{\binom{n}{l}}(1-\frac{1}{q})^{q^t}\qquad(10)\end{aligned}$$

for $u=|Z|>\beta n$ and because $l\geq l_i$.

Lemma 8. 1. *Let $|Z|>\beta n$, where $0\leq\beta\leq 1$ is fixed as n tends to ∞, then*

$$E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}})\leq 1-\sum_{t=0}^{l}\binom{l}{t}\beta^{l-t}(1-\beta)^t(1-\frac{1}{q})^{q^t}+O(\frac{1}{n}),$$

where $O(\frac{1}{n})$ doesn't depend on i.

2. *The function $F(\beta)=1-\sum_{t=0}^{l}\binom{l}{t}\beta^{l-t}(1-\beta)^t(1-\frac{1}{q})^{q^t}$ is not increasing in $0\leq\beta\leq 1$ and $\frac{1}{q}\leq F(\beta)\leq 1-(1-\frac{1}{q})^{q^l}<1$.*

Proof. By taking $\lim_{n\to\infty}$, the first statement of the Lemma follows from (10).

To prove the second statement we fix any $0\leq\beta_1\leq\beta_2\leq 1$ and two subsets A and B of X such that $|A|=\lfloor\beta_1 n\rfloor$ and $|B|=\lfloor\beta_2 n\rfloor$. We see that $|A|\leq|B|$. Then (9) holds, where we put $|X_i|=l$, which is equivalent to

$$1-\sum_{t=0}^{l}\frac{\binom{\lfloor\beta_1 n\rfloor}{l-t}\binom{n-\lfloor\beta_1 n\rfloor}{t}}{\binom{n}{l}}(1-\frac{1}{q})^{q^t}\geq 1-\sum_{t=0}^{l}\frac{\binom{\lfloor\beta_2 n\rfloor}{l-t}\binom{n-\lfloor\beta_2 n\rfloor}{t}}{\binom{n}{l}}(1-\frac{1}{q})^{q^t}.$$

By applying $\lim_{n\to\infty}$ to both the sides of the inequality, we get $F(\beta_1)\geq F(\beta_2)$. The Lemma is proved.

The inequality (6) then implies

$$E|U'_k|\leq q^{\beta n-k}+E_{X_1,\ldots,X_k}(q^{|X(k)|-k})\,(F(\beta)+\varepsilon)^{m-k}.\qquad(11)$$

for any positive real ε as n tends to ∞.

For $0 \leq \alpha \leq l$ we define the function $0 \leq \beta(\alpha) \leq 1$ by the rule: $\beta = \beta(\alpha)$ is the solution of the equation

$$q^{\beta - \frac{\alpha}{l}} = qe^{g(\alpha)} F(\beta)^{c - \frac{\alpha}{l}} \tag{12}$$

if such a solution exists and $\beta(\alpha) = 0$ otherwise. By Theorem 3 there should be at most one solution to the equation (12). We remaind that $\frac{l_1 + l_2 + \ldots + l_m}{ln}$ tends to a constant $c \geq 1$ as n tends to ∞ while q and l are fixed.

Theorem 3. *1. The equation* (12) *has at most one solution for any* $0 \leq \alpha \leq l$.

2. Let $L_k = l_1 + \ldots + l_k$ *and* $\alpha = L_k/n$, *and* $k \leq n$. *Let* $0 < \delta < 1$ *be fixed as* n *tends to* ∞. *Then*

$$E|U'_k| = \begin{cases} < q^{n^\delta}, & \text{if} \quad L_k < n^\delta; \\ O((q^{\beta(\alpha) - \frac{\alpha}{l}} + \varepsilon)^n), & \text{if} \quad ln > L_k \geq n^\delta; \\ < 1, & \text{if} \quad L_k \geq ln, \end{cases}$$

for any positive real ε.

Proof. Since the function $F(\beta)$ is not increasing as $0 \leq \beta \leq 1$, the function on the right hand side of (12) is not increasing in β while the function on the left hand side is strictly increasing. Therefore, there should be at most one solution and the first statement is proved.

The second statement of the Theorem is obviously true if $L_k < n^\delta$. Let $L_k \geq ln$ now. Then $\frac{lk}{n} \geq \frac{L_k}{n} \geq l$ and $k \geq n$. So

$$E|U'_k| \leq E|U_k| = Eq^{|X(k)| - k} < 1$$

and the statement is true in this case as well.

Let $ln > L_k \geq n^\delta$. Then by Lemma 2 we get from (11) that

$$E|U'_k| \leq (q^{\beta - \frac{\alpha}{l}})^n + O((qe^{g(\alpha)} + \varepsilon)^n (F(\beta) + \varepsilon)^{\frac{m-k}{n} n}),$$

as $\frac{\alpha}{l} \leq \frac{k}{n}$ and for any positive real ε. We realize that $\frac{m-k}{n} \geq c_n - \frac{\alpha}{l}$, where $c_n = \frac{l_1 + l_2 + \ldots + l_m}{ln}$. Therefore, for a small enough ε to provide $F(\beta) + \varepsilon < 1$ and as n tends to ∞ we get

$$E|U'_k| \leq (q^{\beta - \frac{\alpha}{l}})^n + O((qe^{g(\alpha)} + \varepsilon)^n (F(\beta) + \varepsilon)^{(c_n - \frac{\alpha}{l})n}).$$

Since $\lim c_n = c \geq 1$ and $\alpha < l$, this implies

$$E|U'_k| \leq (q^{\beta - \frac{\alpha}{l}})^n + O((qe^{g(\alpha)} F(\beta)^{c - \frac{\alpha}{l}} + \varepsilon)^n) \tag{13}$$

for any real positive ε as n tends to ∞. If (12) has one solution, then the inequality $E|U'_k| = O((q^{\beta(\alpha) - \frac{\alpha}{l}} + \varepsilon)^n)$ follows from (13) and (12).

Let (12) have no solutions. So $\beta(\alpha) = 0$ by the definition of the function $\beta(\alpha)$. We claim that the value of the left hand side function in (12) at $\beta = 1$ is bigger than that of the right hand side function. That is,

$$q^{1 - \frac{\alpha}{l}} > qe^{g(\alpha)} q^{-(c - \frac{\alpha}{l})}.$$

This inequality is equivalent to $q^{c-\frac{\alpha}{l}} > e^{f(z_\alpha)-\alpha+\alpha\ln\alpha}$, where $f(z_\alpha)-\alpha+\alpha\ln\alpha < 0$ for any $\alpha > 0$, and is therefore true. So if the equation (12) doesn't admit any solution, then the right had side function is always bounded by the value of the left hand side function at $\beta = 0$, that is $q^{-\frac{\alpha}{l}}$. So the inequality $E|U'_k| = O((q^{\beta(\alpha)-\frac{\alpha}{l}} + \varepsilon)^n)$ follows from (13) and (12) again. The Theorem is proved.

The main Theorem 1 now follows from Theorem 3 and formula (5).

References

1. M. Bardet, J.-C.Faugére, and B. Salvy, *Complexity of Gröbner basis computation for semi-regular overdetermined sequences over F_2 with solutions in F_2,* Research report RR–5049, INRIA, 2003.
2. N. Courtois, A. Klimov, J. Patarin, and A. Shamir, *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations,* in Eurocrypt 2000, LNCS 1807, pp. 392–407, Springer-Verlag, 2000.
3. V.P. Chistyakov, *Discrete limit distributions in the problem of shots with arbitrary probabilities of occupancy of boxes,* Matem. Zametki, vol. 1, pp. 9–16, 1967.
4. J.-C. Faugère, *A new efficient algorithm for computing Gröbner bases without reduction to zero (F5),* Proc. of ISSAC 2002, pp. 75 – 83, ACM Press, 2002.
5. K. Iwama, *Worst-Case Upper Bounds for kSAT,* The Bulletin of the EATCS, vol. 82, pp. 61–71, 2004.
6. V. Kolchin, *The rate of convergence to limit distributions in the classical problem of shots,* Teoriya veroyatn. i yeye primenen., vol. 11, pp. 144–156, 1966.
7. V. Kolchin, A. Sevast'yanov, and V. Chistyakov, *Random allocations,* John Wiley & Sons, 1978.
8. H. Raddum, *Solving non-linear sparse equation systems over $GF(2)$ using graphs,* University of Bergen, preprint, 2004.
9. H. Raddum, I. Semaev, *New technique for solving sparse equation systems*, Cryptology ePrint Archive, 2006/475.
10. H. Raddum, I. Semaev, *Solving MRHS linear equations*, submitted, extended abstract in Proceedings of WCC'07, 16-20 Avril 2007, Versailles, France, INRIA, 323–332.
11. I. Semaev, *On solving sparse algebraic equations over finite fields*, submitted, extended abstract in Proceedings of WCC'07, 16-20 Avril 2007, Versailles, France, INRIA, 361–370.
12. E.P.K. Tsang, *Foundations of constraint satisfaction*, Academic Press, 1993.
13. B.-Y. Yang, J-M. Chen, and N.Courtois, *On asymptotic security estimates in XL and Gröbner bases-related algebraic cryptanalysis,* in ICICS 2004, LNCS 3269, pp. 401–413, Springer-Verlag, 2004.
14. A. Zakrevskij, I. Vasilkova, *Reducing large systems of Boolean equations,* 4th Int.Workshop on Boolean Problems, Freiberg University, September, 21–22, 2000.