

On Deciding Deep Holes of Reed-Solomon Codes

Qi Cheng and Elizabeth Murray*

School of Computer Science
The University of Oklahoma
Norman, OK 73019, USA
Email: {qcheng, wumpus}@ou.edu.

Abstract. For generalized Reed-Solomon codes, it has been proved [7] that the problem of determining if a received word is a deep hole is co-NP-complete. The reduction relies on the fact that the evaluation set of the code can be exponential in the length of the code – a property that practical codes do not usually possess. In this paper, we first present a much simpler proof of the same result. We then consider the problem for standard Reed-Solomon codes, i.e. the evaluation set consists of all the nonzero elements in the field. We reduce the problem of identifying deep holes to deciding whether an absolutely irreducible hypersurface over a finite field contains a rational point whose coordinates are pairwise distinct and nonzero. By applying Cafure-Matera estimation of rational points on algebraic varieties, we prove that the received vector $(f(\alpha))_{\alpha \in \mathbf{F}_p}$ for the Reed-Solomon $[p-1, k]_p$, $k < p^{1/4-\epsilon}$, cannot be a deep hole, whenever $f(x)$ is a polynomial of degree $k+d$ for $1 \leq d < p^{3/13-\epsilon}$.

Keywords: Reed-Solomon codes, deep hole, NP-complete, algebraic surface.

1 Introduction

A signal, when transferred over a long distance, always has a possibility of being corrupted. Error-detecting and error-correcting codes alleviate the problem and make the modern communication possible. The Reed-Solomon codes are very popular in engineering a reliable channel due to their simplicity, burst error-correction capabilities, and the powerful decoding algorithms they admit.

Let \mathbf{F}_q be the finite field with q elements, where q is a prime power. The encoding process of generalized Reed-Solomon codes can be thought of as a map from $\mathbf{F}_q^k \rightarrow \mathbf{F}_q^n$, in which a message (a_1, a_2, \dots, a_k) is mapped to a vector

$$(f(x_1), f(x_2), \dots, f(x_n)),$$

* This research is partially supported by NSF Career Award CCR-0237845

where $f(x) = a_k x^{k-1} + a_{k-1} x^{k-2} + \dots + a_1 \in \mathbf{F}_q[x]$ and $\{x_1, x_2, \dots, x_n\} \subseteq \mathbf{F}_q$ is called the evaluation set. (Note that different encoding schemes are possible.)

It is not difficult to see that the set of codewords formed in this manner is a linear subspace of \mathbf{F}_q^n which has dimension k . Generalized Reed-Solomon codes are therefore *linear codes*, because they are linear subspaces of \mathbf{F}_q^n , where n is the length of a codeword.

The *Hamming distance* between two codewords is the number of coordinates in which they differ – or one can think of this as the number of modifications required to transform one vector into another. A *Hamming ball of radius m* is a set of vectors within Hamming distance m to some vector in \mathbf{F}_q^n . The minimum distance of a code is the smallest distance between any two distinct codewords, and is a measure of how many errors the code can correct or detect. The *covering radius of a code* is the maximum possible distance from any vector in \mathbf{F}_q^n to the closest codeword. A *deep hole* is a vector which achieves this maximum. The minimum distance of generalized Reed-Solomon codes is $n - k + 1$. The covering radius of generalized Reed-Solomon codes is $n - k$.

A code is useless without a *decoding algorithm*, which takes some *received word* (a vector in \mathbf{F}_q^n) and outputs a message. The message should correspond, ideally, to the codeword which is closest, with respect to Hamming distance, to the received word. If we assume that each coordinate in a received word is equally likely to be in error, then the closest codeword is the most likely to be the intended transmission.

Standard Reed-Solomon codes use \mathbf{F}_q^* as their evaluation set. If the evaluation set is \mathbf{F}_q , then the code is called an *extended Reed-Solomon code*. If the evaluation set is the set of rational points in a projective line over \mathbf{F}_q , then the code is known as a *doubly extended Reed-Solomon code*. The difference between standard Reed-Solomon codes, extended Reed-Solomon codes and doubly extended Reed-Solomon codes is not significant, but generalized Reed-Solomon codes are quite unique, as the evaluation set can be exponentially larger than the length of a codeword.

1.1 Related Work

The pursuit of efficient decoding algorithms for Reed-Solomon codes has yielded intriguing results. If the radius of a Hamming ball centered at some received word is less than half the minimum distance, there can be at most one codeword in the Hamming ball. Finding this codeword is called *unambiguous decoding*. It can be efficiently solved, see [1] for a simple algorithm.

If the radius is less than $n - \sqrt{n(k-1)}$, the problem can be solved by the Guruswami-Sudan algorithm [6], which outputs all the codewords inside a Hamming ball. If the radius is stretched further, the number of codewords in a Hamming ball may be exponential. We then study *the bounded distance decoding problem*, which outputs just one codeword in any Hamming ball of a certain radius. More importantly, we can remove the restriction on radius and investigate the *maximum likelihood decoding problem*, which is the problem of computing the closest codeword to any given vector in \mathbf{F}_q^n .

The question on decodability of Reed-Solomon codes has attracted attention recently, due to recent discoveries on the relationship between decoding Reed-Solomon codes and some number theoretical problems. Allowing exponential alphabets, Guruswami and Vardy proved that the maximum likelihood decoding is NP-complete. They essentially showed that deciding deep holes is co-NP-complete. When the evaluation set is precisely the whole field or \mathbf{F}_q^* , an NP-completeness result is hard to obtain, Cheng and Wan [3] managed to prove that the decoding problem of Reed-Solomon codes at certain radius is at least as hard as the discrete logarithm problem over finite fields. In this paper, we wish to establish an additional connection between decoding of standard Reed-Solomon codes and a classical number-theoretic problem – that of determining the number of rational points on an algebraic hypersurface.

1.2 Our Results

The decoding problem of Reed-Solomon codes can be reformulated into the problem of *curve fitting* or *noisy polynomial reconstruction*. In this problem, we are given n points

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

in \mathbf{F}_q^2 . The goal is to find polynomials of degree $k-1$ that pass as many of the n points as possible. Note that all the x -coordinates are distinct.

Given the received word $w = (y_1, y_2, \dots, y_n)$, we are particularly interested in the polynomial obtained by interpolating the n points.

$$\begin{aligned} w(x) = & y_1 \frac{(x-x_2)(x-x_3)\cdots(x-x_n)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_n)} \\ & + \cdots + y_i \frac{(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} \\ & + \cdots + y_n \frac{(x-x_1)(x-x_2)\cdots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})}. \end{aligned}$$

In this paper, we say that a polynomial $w(x)$ *generates* a vector $w \in \mathbf{F}_q^n$ if

$$w = (w(x_1), w(x_2), \dots, w(x_n)).$$

If the polynomial $w(x)$ has degree $k - 1$ or less, w must be a codeword, and vice versa (since codewords consist of the encodings of all messages of length k). If $w(x)$ has degree k , w must be a deep hole (as we will later show). What if it has degree larger than k ? Can it be a deep hole?

In this paper, we try to answer this question. If a received word is a deep hole, there is no codeword which is at distance $n - k - 1$ or closer to the received word. Hence if the distance bound is $n - k - 1$, a decoding algorithm can tell a received word is deep hole or not by checking whether there is a codeword in the Hamming ball of radius $n - k - 1$. This shows that maximum likelihood decoding of Reed-Solomon codes, as well as the bounded distance decoding at radius $n - k - 1$, is at least as hard as deciding deep holes. Observe that the bounded distance decoding at a distance of $n - k$ or more can be done efficiently. It is hoped that we can decrease the radius until we reach the domain of hard problems.

We are mainly concerned with the case when the evaluation set consists of nonzero elements of the field. Notice that for generalized Reed-Solomon code, the bounded distance decoding at distance $n - k - 1$ is NP-hard. We reduce the problem to deciding whether an absolutely irreducible hypersurface contains a rational point whose coordinates are pairwise distinct and nonzero. From the reduction, we show that if k and the degree of $w(x)$ are small, $w(x)$ cannot generate deep holes. More precisely

Theorem 1. *Let p be a prime and $1 < k < p^{1/4-\epsilon}$ be a positive integer. The vector $(w(\alpha))_{\alpha \in \mathbf{F}_p}$ is not a deep hole in the Reed-Solomon code $[p, k]_p$ if the degree of $w(x)$ is greater than k but less than $k + p^{3/13-\epsilon}$.*

Roughly speaking, the theorem indicates that a vector generated by a low degree polynomial can not be a deep hole, even though it is very far away from any codeword.

To prove the theorem, we need to estimate the number of rational points on an algebraic hypersurface over a finite field. This problem is one of the central problems in algebraic geometry and finite field theory. Weil, through his proof of the Riemann Hypothesis for function fields, provided a bound for the number of points on algebraic curves. This bound was later generalized by Weil and Lang to algebraic varieties. Schmidt [8] obtained some better bounds for absolutely irreducible hypersurfaces by

elementary means. In this paper, we will use an improved bound, obtained by Cafure and Matera [2] very recently. But first we give a new proof that deciding whether or not a received word is a deep hole is co-NP-complete. Our reduction is straight-forward and much simpler than the one constructed by Guruswami and Vardy.

2 A simple proof that the maximum likelihood decoding is NP-complete

We reduce the following finite field subset sum problem [5, Page 233] to deep hole problem of generalized Reed-Solomon codes.

Instance: A set of n elements $A = \{x_1, x_2, x_3, \dots, x_n\} \subseteq \mathbf{F}_{2^m}$, an element $b \in \mathbf{F}_{2^m}$ and a positive integer $k < n$.

Question: Is there a nonempty subset $\{x_{i_1}, x_{i_2}, \dots, x_{i_{k+1}}\} \subseteq A$ of cardinality $k + 1$ such that

$$x_{i_1} + x_{i_2} + \dots + x_{i_{k+1}} = b.$$

Now consider the generalized Reed-Solomon code $[n, k]_{2^m}$ with evaluation set A . Suppose we have a received word

$$w = (f(x_1), f(x_2), \dots, f(x_n))$$

where $f(x) = x^{k+1} - bx^k$. If the word is not a deep hole, it is at most $n - k - 1$ away from a codeword. Suppose that the codeword is generated by $t(x)$. Then $t(x)$ has degree at most $k - 1$ and $f(x) - t(x)$ has at least $k + 1$ distinct roots in A . Since $f(x) - t(x)$ is a monic polynomial with degree $k + 1$, we have

$$f(x) - t(x) = x^{k+1} - bx^k - t(x) = (x - x_{i_1})(x - x_{i_2}) \cdots (x - x_{i_{k+1}}),$$

for some $x_{i_1}, x_{i_2}, \dots, x_{i_{k+1}}$ in A . Therefore $x_{i_1} + x_{i_2} + \dots + x_{i_{k+1}} = b$.

On the other hand, if $x_{i_1} + x_{i_2} + \dots + x_{i_{k+1}} = b$, $f(x) - (x - x_{i_1})(x - x_{i_2}) \cdots (x - x_{i_{k+1}})$ generates a codeword. It shares $k + 1$ values with w , thus has distance less than $n - k - 1$ away from a codeword, so it cannot be a deep hole.

In summary, w is not a deep hole if and only if the answer to the instance of the finite field subset sum problem is “Yes”. Hence the deep hole problem is co-NP-complete.

By a similar argument, we know that the polynomials of degree k must generate deep holes. Hence

Corollary 1. *For a generalized Reed-Solomon code $[n, k]_q$, there are at least $(q - 1)q^k$ many deep holes.*

We remark that the argument cannot work for small evaluation sets, because the subset sum is easy in that case. In fact, if the evaluation set is the whole field and $k > 1$, then a polynomial of degree $k + 1$ cannot generate a deep hole.

3 A hypersurface related to deep holes

The above argument motivates us to consider vectors generated by polynomials of larger degree. We are given some received word w and we want to know whether or not it is a deep hole. The received word is generated by $w(x)$ of the form $f(x) + t(x)$ where $f(x)$ is some polynomial containing no terms of degree smaller than k , and $t(x)$ is some polynomial containing only terms of degree $k - 1$ or smaller. For purposes of determining whether or not w is a deep hole, we fix a monic $f(x)$

$$f(x) = x^{k+d} + f_{d-1}x^{k+d-1} + \dots + f_0x^k \in \mathbf{F}_q[x]$$

and let $t(x)$ vary and ask whether $f(x) + t(x)$ has $k + 1$ roots, or perhaps more.

In its essence, the question is one of finding a polynomial whose leading terms are $f(x)$, and which has as many zeroes as possible over a certain field. (As stated earlier, for $k > 1$, if $f(x)$ has degree k , then w is a deep hole. If $f(x)$ has degree $k + 1$, then it is not a deep hole.)

The most obvious way to approach this problem is to symbolically divide $f(x)$ by a polynomial that is the product of $k + 1$ distinct (but unknown) linear factors, and determine whether or not it is possible to set the the leading term of the remainder, i.e., the coefficient of x^k , equal to zero. If the leading coefficient is 0, the remainder has degree $k - 1$ or less in x , which generates a codeword. The distance between the codeword and w will be at most $n - k - 1$.

A polynomial that is the product of $k + 1$ distinct linear factors will have the elementary symmetric polynomials as coefficients,

$$\Pi = (x - x_1)(x - x_2)\dots(x - x_{k+1}) = x^{k+1} + \pi_1x^k + \pi_2x^{k-1} + \dots + \pi_{k+1},$$

where π_i is the i -th symmetric polynomial in x_1, x_2, \dots, x_{k+1} .

Since Π is monic, the remainder of the division of $f(x)$ by Π will be a polynomial in $\mathbf{F}_q[x_1, x_2, \dots, x_{k+1}][x]$ of degree less than $k + 1$. Denote

the leading coefficient of the remainder by $L_{f_0, f_1, \dots, f_{d-1}}(x_1, x_2, \dots, x_{k+1})$. This is a multivariate polynomial of degree d .

As an example, imagine the division of the polynomial x^{k+1} by Π . We can easily verify that the leading term of the remainder is $-\pi_1 x^k$. Since we can always find $k + 1$ distinct values that will satisfy $\pi_1 = 0$, we know that x^{k+1} cannot be a deep hole. But, in most cases $w(x)$ will have larger degree and contain many terms, and the remainder will be a more complex multivariate polynomial in $k + 1$ variables, rather than a linear polynomial in $k + 1$ variables. If the leading coefficient itself has a solution where all roots are distinct and invertible, then $f(x) + t(x)$ cannot generate a deep hole.

We now argue that the leading coefficient of the remainder is absolutely irreducible. We write

$$L_{f_0, f_1, \dots, f_{d-1}}(x_1, x_2, \dots, x_{k+1}) = F_d + F_{d-1} + \dots + F_0,$$

where F_i is a form containing all the terms of degree i in L . The polynomial $L_{f_0, f_1, \dots, f_{d-1}}(x_1, x_2, \dots, x_{k+1})$ is absolutely irreducible if F_d is absolutely irreducible. To see this, suppose that L can be factored as $L'L''$. Let F'_{d_1} be the form of highest degree in L' and F''_{d_2} be the form of highest degree in L'' . Then we have $F_d = F'_{d_1}F''_{d_2}$, a contradiction to the condition that F_d is absolutely irreducible.

Fortunately F_d does not depend on f_i 's.

Lemma 1. *The form of the highest degree in $L_{f_0, f_1, \dots, f_{c-1}}(x_1, x_2, \dots, x_{k+1})$ is exactly $L_{0,0,\dots,0}(x_1, x_2, \dots, x_{k+1})$.*

Proof. It can be proved by mathematical induction on c .

In the next section, we argue that the term of highest degree in the leading coefficient, which we will call $\chi_d(x_1, x_2, \dots, x_{k+1})$, is absolutely irreducible. We will actually show that $\chi_d(x_1, x_2, 1, 0, 0 \dots 0)$ is absolutely irreducible. This is because that $\chi_d(x_1, x_2, 1, 0, 0 \dots 0)$ has the same degree as $\chi_d(x_1, x_2, \dots, x_{k+1})$, if the former is irreducible, so is the latter.

Lemma 2. $\chi_d(x_1, x_2, 1, 0, 0 \dots 0) = \sum_{i+j \leq d} x_1^i x_2^j$.

Proof. We need to compute the leading coefficient of the remainder after dividing x^{k+d} by $(x - x_1)(x - x_2)(x - 1)x^{k-2}$. It is as same as the leading coefficient of the remainder after dividing x^{d+2} by $(x - x_1)(x - x_2)(x - 1)$. The remainder is a quadratic polynomial in x . When we evaluate it at x_1 , it takes the value x_1^{d+2} . When we evaluate it at x_2 , it takes the value

x_2^{d+2} . When we evaluate it at 1, it takes value 1. By interpolating, we obtain the unique polynomial satisfying these conditions. It is

$$x_1^{d+2} \frac{(x-x_2)(x-1)}{(x_1-x_2)(x_1-1)} + x_2^{d+2} \frac{(x-x_1)(x-1)}{(x_2-x_1)(x_2-1)} + \frac{(x-x_1)(x-x_2)}{(1-x_1)(1-x_2)}$$

The leading coefficient is

$$\frac{x_1^{d+2}}{(x_1-x_2)(x_1-1)} + \frac{x_2^{d+2}}{(x_2-x_1)(x_2-1)} + \frac{1}{(1-x_1)(1-x_2)},$$

which is equal to $\sum_{i+j \leq d} x_1^i x_2^j$.

4 A smooth curve

The section is devoted to the proof of the irreducibility of the bivariate polynomial $\sum_{i+j \leq d} x^i y^j$ over \mathbf{F}_p .

Lemma 3. *Let p be a prime and $d < p - 2$ be a positive integer. The curve $f(x, y) = \sum_{i+j \leq d} x^i y^j = 0$ is absolutely irreducible over \mathbf{F}_p .*

Proof. To show that $f(x, y) = \sum_{i+j \leq d} x^i y^j$ is absolutely irreducible, we actually prove a stronger statement that $f(x, y) = 0$ is a smooth algebraic curve. It can be verified by simple calculation that places on the curve at infinity are nonsingular. Hence it is sufficient to show that all the affine places on the curve are nonsingular, i.e. that the system of equations

$$\begin{cases} f(x, y) = 0 \\ \frac{\partial f}{\partial x} = 0 \\ \frac{\partial f}{\partial y} = 0 \end{cases}$$

has no solution.

First, it is convenient to write $f(x, y)$ as

$$x^d + (y+1)x^{d-1} + (y^2+y+1)x^{d-2} + \dots + (y^d + \dots + 1).$$

We write $\frac{\partial f}{\partial x}$ as:

$$dx^{d-1} + (d-1)(y+1)x^{d-2} + \dots + (y^{d-1} + \dots + 1)$$

and $\frac{\partial f}{\partial y}$ as

$$dy^{d-1} + (d-1)(x+1)y^{d-2} + \dots + (x^{d-1} + \dots + 1)$$

Assume that there is a solution (X, Y) to the system of equations. Compute $(x-y)\frac{\partial f}{\partial x}$:

$$\begin{aligned} x\frac{\partial f}{\partial x} &= dx^d + (d-1)(y+1)x^{d-1} + \dots + x(y^{d-1} + \dots + 1) \\ &= (d+1)x^d + d(y+1)x^{d-1} + \dots + (y^d + y^{d-1} + \dots + 1) - f(x, y) \\ y\frac{\partial f}{\partial x} &= dyx^{d-1} + (d-1)(y^2+y)x^{d-2} + \dots + (y^d + \dots + 1) \end{aligned}$$

Their difference is:

$$\begin{aligned} (x-y)\frac{\partial f}{\partial x} &= dx^d + [d-(y+1)]x^{d-1} + \dots - (y^d + \dots + 1) \\ &= (d+1)x^d + dx^{d-1} + (d-1)x^{d-2} + \dots + 1 - f(x, y). \end{aligned}$$

Since $f(X, Y)$ must be zero, we know that:

$$(d+1)X^d + dX^{d-1} + (d-1)X^{d-2} + \dots + 1 = 0$$

we multiply by the above X , and then subtract the original expression to get:

$$(d+1)X^{d+1} = X^d + X^{d-1} + \dots + 1. \quad (1)$$

Repeat the process on $\frac{\partial f}{\partial y}$, we get

$$(d+1)Y^{d+1} = Y^d + Y^{d-1} + \dots + 1. \quad (2)$$

This shows that neither X nor Y can be zero. Now, observe that

$$(x-y)f(x, y) = x^{d+1} + x^d + \dots + 1 - y^{d+1} - y^d - \dots - 1 = 0.$$

This means that $(d+2)X^{d+1} = (d+2)Y^{d+1}$. We also know that the right hand sides of (1) and (2) are actually $\frac{X^{d+1}-1}{X-1}$, and $\frac{Y^{d+1}-1}{Y-1}$. So multiplying both sides by $X-1$ for 1 and by $Y-1$ for 2, we obtain

$$(d+1)X^{d+2} - (d+2)X^{d+1} = -1$$

$$(d+1)Y^{d+2} - (d+2)Y^{d+1} = -1.$$

Hence we have $(d+1)X^{d+2} = (d+1)Y^{d+2}$.

Since $d + 2 < p$, this means that the characteristic of the field does not divide either $d + 1$ or $d + 2$. We have $X^{d+2} = Y^{d+2}$ and $X^{d+1} = Y^{d+1}$. Therefore $X = Y$.

At the point (X, Y) , $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ have the same form

$$(d + (d - 1) + \cdots + 1)X^{d-1} + ((d - 1) + (d - 2) + \cdots)X^{d-2} + \cdots + 1 = 0.$$

And $f(X, Y)$ can be written (since $X = Y$) as

$$(d + 1)X^d + dX^{d-1} + (d - 1)X^{d-2} + \cdots + 1 = 0.$$

If we subtract $f(X, Y)$ from $\frac{\partial f}{\partial x}(X, Y)$, we get

$$(d + 1)X^d - ((d - 1) + (d - 2) + \cdots + 1)X^{d-1} - \cdots - X = 0.$$

Divide the result by X , we have

$$\begin{aligned} & (d + 1)X^{d-1} - ((d - 1) + (d - 2) + \cdots + 1)X^{d-2} - \cdots - 1 \\ &= (d + 1)X^{d-1} + (d + (d - 1) + \cdots + 1)X^{d-1} \\ &= 0. \end{aligned}$$

This means that $\frac{(d+1)(d+2)}{2}X^{d-1} = 0$, hence $X = 0$, and this is a contradiction.

5 Estimation of number of rational points

Cafure and Matera [2, Theorem 5.2] have obtained the following estimation of number of rational points on an absolutely irreducible hypersurface:

Proposition 1. *An absolutely irreducible \mathbf{F}_q -hypersurface of degree d in \mathbf{F}_q^n contains at least*

$$q^{n-1} - (d - 1)(d - 2)q^{n-3/2} - 5d^{13/3}q^{n-2}$$

many \mathbf{F}_q -rational points.

We also use the following proposition [2, Lemma 2.2].

Proposition 2. *Suppose $f_1(x_1, x_2, \dots, x_n)$ and $f_2(x_1, x_2, \dots, x_n)$ are polynomials of degree not greater than d , and they donot have a common factor. Then the number of \mathbf{F}_q -rational solutions of*

$$f_1 = f_2 = 0$$

is at most d^2q^{n-2} .

We seek solutions of $L(x_1, x_2, \dots, x_{k+1})$ but not of

$$\prod_{1 \leq i \leq k+1} x_i \prod_{1 \leq i < j \leq k+1} (x_i - x_j) = 0.$$

We count the number of rational solutions of $L(x_1, x_2, \dots, x_{k+1})$, minus the number of rational solutions of

$$\begin{cases} L(x_1, x_2, \dots, x_{k+1})=0 \\ \prod_{1 \leq i \leq k+1} x_i \prod_{1 \leq i < j \leq k+1} (x_i - x_j)=0 \end{cases}$$

The number is greater than

$$p^k - (d-1)(d-2)p^{k-1/2} - 5d^{13/3}p^{k-1} - [\max(d, \frac{k^2+k+2}{2})]^2 p^{k-1},$$

which is greater than 0 if $d < p^{3/13-\epsilon}$ and $k < p^{1/4-\epsilon}$. This concludes the proof of the main theorem.

6 Concluding Remarks

It has been proved that for generalized Reed-Solomon codes, the bounded distance decoding at radius $n - k - 1$ is NP-hard. In this paper, we try to determine the complexity of this problem for standard Reed-Solomon codes. We reduce the problem to a problem of determining whether a hypersurface contains a rational point of distinct coordinates. While we didnot solve the problem completely, we show that for prime fields and small k , this problem is easy if a vector is generated by a small degree polynomial. In essence, we ask whether there exists a polynomial with many distinct rational roots under the restriction that some coefficients are prefixed. This problem bears an interesting comparison with the active research [4] on the construction of irreducible polynomial with some prefixed coefficients.

To solve the problem for every k and every vector, there are two apparent approaches: 1) Find a better estimation of number of rational points on the hypersurfaces. 2) Explore the specialty of the hypersurfaces. From an average argument, it is tempting to conjecture that the vectors generated by polynomials of degree k are the only possible deep holes. If so, we can completely classify deep holes. We leave it as an open problem.

References

1. E. Berlekamp and L. Welch. Error correction of algebraic block codes. U.S. Patent Number 4633470, 1986.

2. A. Cafure and G. Matera. Improved explicit estimates on the number of solutions of equations over a finite field. <http://www.arxiv.org/abs/math.NT/0405302>, 2004.
3. Qi Cheng and Daqing Wan. On the list and bounded distance decodibility of the Reed-Solomon codes (extended abstract) (FOCS). In *Proc. 45th IEEE Symp. on Foundations of Comp. Science*, pages 335–341, 2004.
4. Stephen D. Cohen. Explicit theorems on generator polynomials. *Finite Fields and Their Applications*, 2005. To appear.
5. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
6. Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.
7. Venkatesan Guruswami and Alexander Vardy. Maximum-likelihood decoding of Reed-Solomon codes is NP-hard. In *Proceeding of SODA*, 2005.
8. W. Schmidt. *Equations over Finite Fields. An Elementary Approach*, volume 536 of *Lecture Notes in Mathematics*. Springer-Verlag, 1976.