Hard Problems of Algebraic Geometry Codes

Qi Cheng

***Abstract*—The minimum distance is one of the most important combinatorial characterizations of a code. The maximum likelihood decoding problem is one of the most important algorithmic problems of a code. While these problems are known to be hard for general linear codes, the techniques used to prove their hardness often rely on the construction of artificial codes. In general, much less is known about the hardness of the specific classes of natural linear codes. In this paper, we show that both problems are NP-hard for algebraic geometry codes. We achieve this by reducing a well-known NP-complete problem to these problems using a randomized algorithm. The family of codes in the reductions is based on elliptic curves. They have positive rates, but the alphabet sizes are exponential in the block lengths.**

***Index Terms*—Algebraic-geometric codes, Complexity theory, Maximum likelihood decoding.**

I. Introduction

An $[n,k]_q$ linear error-correcting code is a linear subspace of a vector space $\mathbf{F}_q^n$, where $\mathbf{F}_q$ denotes the finite field of q elements, and k denotes the dimension of the subspace. The *Generator Matrix* for a linear code is a $k \times n$ matrix, with row rank k which defines a linear mapping from $\mathbf{F}_q^k$ (called the *message space*) to $\mathbf{F}_q^n$. Therefore, the code C is:

$$C = \{aG | a \in \mathbf{F}_q^k\}.$$

We call a vector in C a codeword. The *Hamming Distance* between two codewords x and y, is the weight (number of nonzero coordinates) of $x-y$. The minimum distance of a code is the minimum Hamming distance between any two codewords. If the code is linear, then the vector $x - y$ is a codeword, and the minimum distance of the code is equal to the minimum weight of any codeword.

Given a linear code as input, how hard is it to compute the minimum distance? This problem had been open for two decades before it was finally solved by Vardy in 1997 [15], when he proved that the problem is NP-complete. Interestingly, determining whether a code contains a codeword of a given weight was known to be NP-complete much earlier [3].

Dumer et.al. [5] studied the hardness of approximating the minimum distance of a linear code. They showed that the minimum distance of a linear code is not approximable within any constant factor in random polynomial time, unless NP=RP. The codes used in their work and that of Vardy [15] are artificially designed. Their results exhibit that it is hard to compute the minimum distance for the *general* linear codes, but say nothing specific about any of the well-studied and widely-deployed codes, such as the Reed-Solomon codes, the Reed-Muller codes, the BCH codes and the algebraic geometry codes.

The *maximum likelihood decoding* problem, is one of the central problems in algorithmic coding theory. For any vector y in $\mathbf{F}_q^n$, it asks for a codeword x to minimize the distance between x and y. Given that a received word is equally likely to contain an error in any position, codewords that are closest to the received word (i.e. differ in fewer coordinates) are most likely to encode the intended message. This problem is proved to be NP-hard for general linear codes [3]. Proving NP-hardness for classes of useful codes is more difficult and subtle. The only result of this kind known to date is the result of [7] on the NP-completeness of maximum likelihood decoding for *generalized* Reed-Solomon codes, where the sizes of the alphabets are exponential in the lengths of the codes. A related result in [4] shows that decoding of Reed-Solomon codes at certain radius is at least as hard as discrete logarithm problem over finite fields.

In this paper, we prove that the minimum distance problem and the maximum likelihood decoding problem are NP-hard for a natural class of codes, namely, the algebraic-geometry codes. The algebraic geometry codes can be seen as a generalization of the Reed-Solomon codes. While the study of algebraic geometry codes began as a purely mathematical pursuit, an increased understanding of their unique combinatorial properties promises that they will find real-world applications in the foreseeable future. In combinatorics, it is often hard to explicitly construct an object which is, in certain aspects, better than a random object. A family of algebraic geometry codes is one of a few bright spots, where we can explicitly construct a code having more codewords than a random code given the block length and the minimum distance. Moreover, given proper representations, these codes possess a polynomial time list decoding algorithm [8], which corrects errors well beyond half of the minimum distance. In contrast, a random code usually does not have a good decoding algorithm due to the lack of algebraic structure.

Remark 1: Since any linear code is a *weakly* algebraic geometry code [13], we only consider algebraic geometry codes in the strict sense in this paper. All of our results apply to strongly algebraic geometry codes.

Proving the NP-hardness of the maximum likelihood decoding of algebraic geometry codes (MLDAGC) an-

School of Computer Science, the University of Oklahoma, Norman, OK 73019, USA. Email: qcheng@cs.ou.edu. This research is partially supported by NSF Career Award CCR-0237845

swers the most important question about the decodability of this class of codes. Proving the NP-hardness of the minimum distance problem for algebraic geometry codes (MDPAGC) is also well motivated. The designed distance, which is a lower bound of the minimum distance, can be easily obtained from the description of the codes. Less attention is paid to the problem of computing the exact minimum distance. Although minimum distances of Reed-Solomon codes can be easily computed, our result shows that the minimum distance problem of algebraic geometry codes are very hard, even in the case of elliptic codes.

Remark 2: The most interesting family of algebraic geometry codes has a fixed alphabet. The codes in our results have alphabets of exponential size. Nonetheless, we observe that all the known decoding algorithms for algebraic geometry codes are not sensitive to the size of the alphabets. Our results indicate that if a polynomial time maximum likelihood decoding algorithm for algebraic geometry codes does exist, it can only work for codes with a small alphabet size.

A nice surprise about our proofs is their conceptual simplicity. We use the subset sum problem directly, thus all of the results on the preprocessing subset sum problem can be readily carried over to the preprocessing maximum likelyhood decoding decoding of algebraic geometry codes. However our reductions are randomized, which we would prefer to avoid. The need for randomization seems to occur in places where we deal with number theory and primes.

Our reduction always maps a "Yes" instance to a "Yes" instance, and maps a "No" instance to a "No" instance in *expected polynomial time*. The reductions in [5] are *reverse unfaithful random reductions*, which always maps a "No" instance to a "No" instance, but with a small probability, maps a "Yes" instance to a "No" instance.

II. Elliptic curves

The Reed-Solomon code of block length n and dimension k is obtained by evaluating polynomials of degree $k-1$ at a set of n elements in a finite field. For a linear $[n,k]_q$ code, the Singleton bound asserts that $d \leq n-k+1$. The Reed-Solomon codes are optimal, in that they satisfy the Singleton bound with equality. It is trivial to read the minimum distance of Reed-Solomon codes from the block length and the dimension.

Algebraic geometry codes are natural generalizations of the Reed-Solomon codes. Let K be a function field over a finite field $\mathbf{F}$. Let $A_1, A_2, \cdots, A_n, B_1, B_2, \cdots, B_m$ be $\mathbf{F}$-rational places. Let $a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_m$ be positive integers. Given a divisor $A = \sum_{i=1}^{n} a_i A_i - \sum_{i=1}^{m} b_i B_i$, define $L(A)$ to be the set of functions with poles only at $A_1, A_2, \cdots, A_n$ with multiplicities at most $a_1, a_2, \cdots, a_n$ respectively, and with zeros at $B_1, B_2, \cdots, B_m$ with multiplicities at least $b_1, b_2, \cdots, b_m$ respectively. The functions in $L(A)$ form a linear space over the field $\mathbf{F}$. It has dimension no less than $deg(A)-g+1$, where g is the genus of the function field, and $deg(A) = \sum_{i=1}^{n} a_i - \sum_{i=1}^{m} b_i$. For the divisor A, we can construct a linear code whose codewords are obtained by evaluating the functions in $L(A)$ at rational places $P_1, P_2, \cdots, P_n$, where $\{P_1, P_2, \cdots, P_n\} \cap \{A_1, A_2, \cdots, A_n, B_1, B_2, \cdots, B_m\} = \emptyset$. We call the code algebraic geometric if $deg(A) < n$. We call it strongly algebraic geometric if $2g - 2 < deg(A) < n$. The designed distance of the code is $n - deg(A)$. For a function f in the function field, we denote its divisor by (f), the pole part of the divisor by $(f)_\infty$ and the zero part of the divisor by $(f)_0$. Hence $(f) = (f)_0 - (f)_\infty$.

To prove that computing minimum distances of algebraic geometry codes is NP-hard, we use codes defined by curves of genus one, i.e., elliptic curves. We first review some facts about elliptic curves. An elliptic curve is a smooth cubic curve. Let $\mathbf{F}$ be a field. If the characteristic of $\mathbf{F}$ is neither 2 nor 3, we may assume that an elliptic curve E is given by an equation

$$y^2 = x^3 + ax + b, \qquad a, b \in \mathbf{F}. \tag{1}$$

The discriminant of this curve is defined as $-16(4a^3 + 27b^2)$. It is essentially the discriminant of the polynomial $x^3 + ax + b$. It should be non-zero for the curve to be smooth. For detailed information about elliptic curves, we refer the reader to Silverman's book [14]. The set of $\mathbf{F}$-rational points on the elliptic curve consists of the solution set over $\mathbf{F}$ of the equation plus a point at infinity, denoted by O. We use $E(\mathbf{F})$ to denote the group. These points form an abelian group with the infinity point as the identity. From now on, we only consider elliptic curves over finite fields $\mathbf{F}_q$ of characteristic greater than 3 and assume that curves are given in form (1). The following properties of elliptic curves are relevant to our result.

Proposition 1: [10] Let $P_1, P_2, \cdots, P_n, P$ be elements in $E(\mathbf{F}_q)$ distinct from O. If $m_1P_1 + m_2P_2 + \cdots + m_nP_n = P$, where m_i, $1 \leq i \leq n$, are positive integers, then there is a function having zeros at $P_1, P_2, \cdots, P_n$, with multiplicities $m_1, m_2, \cdots, m_n$ respectively, a pole at P with multiplicity 1 and a pole at O with multiplicity $m_1 + m_2 + \cdots + m_n - 1$. We can compute the function in time polynomial in $m_1 + m_2 + \cdots + m_n$ and $\log q$.

Since $(x)_\infty = 2O$, $(y)_\infty = 3O$, and consequently, $(x^i)_\infty = 2iO$, $(x^{i-1}y)_\infty = (2i+1)O$, we can compute a basis for $L(\alpha O)$ in polynomial time, and it contains only monomials. If $Q \neq O$, can we compute a basis for $L(Q + \alpha O)$? Since a basis of $L(\alpha O)$ can be computed easily, we only need to find a function $f' \in L(Q + \alpha O) - L(\alpha O)$. Then $f_1, f_2, \cdots, f_{k-1}$ and f' form a basis for $L(Q+(k-1)O)$. It is fairly easy to find such a function. We can simply pick one point $Q' \notin \{Q, O\}$ in randomized polynomial time, and then compute $Q'' = Q - Q'$. Let l_1 be the line passing through Q' and Q'', let l_2 be the line passing through Q and $-Q$. We then

set $f' = l_1/l_2$. Note that $(f') = Q' + Q'' + (-Q) - O - Q - (-Q) = Q' + Q'' - O - Q$. In conclusion:

Proposition 2: There exists a randomized algorithm to find a base of $L(\alpha O)$ and $L(Q + \alpha O)$ in time polynomial in α and $\log q$.

Proposition 3: Let A be a divisor. If $deg(A) \geq 1$, then dimension of $L(A)$ is $deg(A)$.

Proposition 4: $q-2\sqrt{q}+1 \leq |E(\mathbf{F}_q)| \leq q+2\sqrt{q}+1$.

Lenstra [11] showed that the order of a random elliptic curve over $\mathbf{F}_q$ is distributed almost uniformly in the range $[q-2\sqrt{q}+1, q+2\sqrt{q}+1]$. Heath-Brown [9] showed that for most of q, there are many primes in the range $[q-2\sqrt{q}+1, q+2\sqrt{q}+1]$. Combining these two celebrated results, we have

Proposition 5: [6] Given a positive integer N, there is a randomized algorithm finding a prime $p > N$, an elliptic curve over $\mathbf{F}_p$ and a point G on the curve which has a prime order greater than N. The algorithm runs in expected time polynomial in $\log N$.

It seems hard to derandomize the above algorithm. In fact, even an efficient deterministic algorithm to find a prime bigger than a given number is not known. The problem was listed as open in [1]. On the other hand, once the prime, the curve and the point are found, we can test in deterministic polynomial time that they satisfy the requirements. For practical purposes, there is an efficient method based on the theory of complex multiplication to construct an elliptic curve of a given order.

III. The NP-hardness proof of the MDPAGC

We reduce the following well known subset sum problem to the problem of computing minimum distances of algebraic geometry codes.

Instance: A set $A = \{a_1, a_2, a_3, \cdots, a_n\}$ of n positive integers, a positive integer b and a positive integer $k < n$.

Question: Is there a nonempty subset $\{a_{i_1}, a_{i_2}, \cdots, a_{i_k}\} \subseteq A$ of cardinality k such that

$$a_{i_1} + a_{i_2} + \cdots + a_{i_k} = b.$$

First we prove that an elliptic curve version of the problem is NP-hard.

Lemma 1: The following problem (*elliptic curve subset sum problem*) is NP-hard:

Instance: A prime p, an elliptic curve C over $\mathbf{F}_p$, a set of points $A = \{P_1, P_2, P_3, \cdots, P_n, Q\}$ on the curve and a positive integer $k < n$.

Question: Is there a nonempty subset $\{P_{i_1}, P_{i_2}, \cdots, P_{i_k}\} \subseteq A$ of cardinality k such that

$$P_{i_1} + P_{i_2} + \cdots + P_{i_k} = Q$$

Proof: We reduce the subset sum problem to this one. Denote $a_1+a_2+a_3+\cdots+a_n+b$ by N. Applying Proposition 5, we find a prime $p > N$, an elliptic curve over $\mathbf{F}_p$ and a point G on the curve of prime order $q > N$ in randomized polynomial time. Then set

$$Q = bG, P_1 = a_1G, P_2 = a_2G, \cdots, P_n = a_nG.$$

We have that

$$P_{i_1} + P_{i_2} + \cdots + P_{i_k} = Q,$$

if and only if

$$a_{i_1} + a_{i_2} + \cdots + a_{i_k} \equiv b \pmod{q},$$

if and only if

$$a_{i_1} + a_{i_2} + \cdots + a_{i_k} = b.$$

■

Theorem 1: Given an instance of the elliptic curve subset sum problem, we can in randomized polynomial time, construct an algebraic geometry $[n, k]_p$ code with $p = O(q^2)$ such that if the answer to the elliptic curve subset sum problem is "YES", then the code has minimum distance $n - k$. If the answer to the prime field subset sum problem is "NO", then the code has minimum distance $n - k + 1$.

Proof: Given an instance of elliptic curve subset sum problem, we consider an algebraic geometry code generated by evaluating functions in $L(Q+(k-1)O)$ at $P_1, P_2, \cdots$ and P_n. By the Singleton bound, we know that the minimum distance is at most $n - k + 1$. This code has designed distance $n - k$, thus the minimum distance is at least $n - k$. Let $f_1, f_2, \cdots, f_k$ be a basis of $L(Q + (k - 1)O)$, the generator matrix of the code is

$$\begin{pmatrix} f_1(P_1) & f_1(P_2) & \dots & f_1(P_n) \\ f_2(P_1) & f_2(P_2) & \dots & f_2(P_n) \\ \dots & \dots & \dots & \dots \\ f_k(P_1) & f_k(P_2) & \dots & f_k(P_n) \end{pmatrix}$$

The parameters of the code can be computed in randomized polynomial time by Proposition 2.

If there exists a subset $\{P_{i_1}, P_{i_2}, \cdots, P_{i_k}\} \subseteq \{P_1, P_2, \cdots, P_n\}$ such that $P_{i_1} + P_{i_2} + \cdots + P_{i_k} = Q$ in $E(\mathbf{F}_p)$. Then there exists a function f having zeros at P_{i_1}, P_{i_2}, $\cdots, P_{i_k}$ with single multiplicity, a pole at Q with single multiplicity, and a pole at O with multiplicity $k - 1$. We have $f \in L(Q + (k - 1)O)$. Such a function is unique up to a constant factor. The codeword corresponding to f has weight $n-k$, because it has k zeros in $\{P_1, P_2. \cdots, P_n\}$. Hence the minimum distance of the code is $n - k$.

In the other direction, if the minimum weight of the codewords is $n - k$, there exists a function $f \in L(Q + (k - 1)O)$ that has zeros at k many points in $P_1, P_2, \cdots, P_n$. Denote them by $P_{i_1}, P_{i_2}, \cdots, P_{i_k}$. Since it can have no more than k poles, counting multiplicities, it must have exactly k zeros, and all the zeros have single multiplicity. Thus it must have exactly k poles as well. It has a pole at Q with multiplicity 1 and a pole at O with multiplicity $k - 1$. That is to say

$(f) = P_{i_1} + P_{i_2} + \cdots + P_{i_k} - Q - (k-1)O$. Hence in $E(\mathbf{F}_p)$

$$P_{i_1} + P_{i_2} + \cdots + P_{i_k} = Q.$$

■

Corollary 1: If there is a polynomial time Las Vegas algorithm to compute the minimum distance of an algebraic geometry code, then $NP \subseteq ZPP$. If there is a polynomial time randomized algorithm to compute the minimum distance of an algebraic geometry code, then $NP \subseteq RP$.

Corollary 2: Deciding whether an algebraic geometry code is maximum distance separable is NP-hard under a randomized reduction.

IV. A time complexity lower bound for computing the minimum distance

For the above analysis, it is easy to see that we can in time $2^n(\log q)^{O(1)}$ compute the minimum distance of an elliptic $[n,k]_q$ code. Does there exist a better algorithm? If a problem is NP-hard, we do not expect to find an algorithm solving it in polynomial time, not even in subexponential time. However, for NP-hard problems, sometimes we can find exponential algorithms beating the trivial exhaustive search. What can we do in the case of the minimum distance problem of algebraic geometry codes? We can ask the same question for general linear codes as well: can we compute the minimum distance in time $2^{cn}(\log q)^{O(1)}$ for some small c?

Ajtai et.al. [2] have studied the problem. They proposed an algorithm that solves the problem in time $2^{O(n)}$ if the field size is bounded by a polynomial in n. The exact constant hidden in big-O is not calculated in their paper.

The elliptic curve discrete logarithm problem (ECDLP) is to compute l such that $Q = lP$, given $P, Q \in E(\mathbf{F}_q)$. It is assumed in the elliptic curve cryptography that there is no algorithm which runs in time q^c for $c < 1/2$ to solve ECDLP in $\mathbf{F}_q$. But it is not believed to be NP-hard. Since it is obviously an NP-easy problem, there must exist a randomized polynomial time reduction from the ECDLP to any NP-hard problem, including the minimum distance problem of an algebraic geometry code. In this section, we present a succinct reduction. We reduce ECDLP over $\mathbf{F}_q$ to the problem of computing the minimum distance of algebraic geometry $[n,k]_q$ codes, where $n \leq \lfloor \log q \rfloor$. The reduction gives rise a lower bound on the time complexity of computing the minimum distance of linear codes under a cryptographic hardness assumption.

Theorem 2: For any constant $c > 0$, if there is an algorithm which in expected time $2^{cn}(\log q)^{O(1)}$ computes the minimum distance of any linear $[n,k]_q$ code, then the ECDLP over $\mathbf{F}_q$ can be solved in expected time q^c.

Proof: Suppose that we need to compute the discrete logarithm of Q base P on elliptic curve $E(\mathbf{F}_q)$. W.l.o.g, we assume that P has a prime order p. Note that we must have $p \leq q + 1 - 2\sqrt{q}$ by Proposition 4.

Denote the largest even number which is not bigger than $\lfloor \log p \rfloor$ by n. Randomly select a positive integer $r < p$, compute $R = rQ$. With probability $\binom{n}{n/2}/2^n > 1/n^{O(1)}$, the discrete logarithm of R is an integer that when written in binary, has exactly $n/2$ ones and $n/2$ zeros.

Now consider the code C generated by evaluating functions in $L(R + (n/2-1)O)$ at $P_0 = P, P_1 = 2P, P_2 = 2^2P, \cdots, P_{n-1} = 2^{n-1}P$. The minimum distance of the code is $n/2$ iff R can be written as a sum of $n/2$ points from $P_0, P_1, \cdots, P_{n-1}$. Denote the set of these $n/2$ points by D. Let C_i be the code generated by evaluating functions in $L(R+(n/2-1)O)$ at $P_0, P_1, \cdots, P_{i-1}, P_{i+1}, \cdots, P_{n-1}$. We can find D by asking the question where the minimum distance of C_i, for $1 \leq i \leq n$, is $n/2$. Basically, $P_i \in D$ iff the answer for C_i is "No". Assume that we find $D = \{P_{i_1}, P_{i_2}, \cdots, P_{i_{n/2}}\}$. Then

$$\log_P R = \sum_{1 \leq j \leq n/2} 2^{i_j}.$$

Hence $\log_P Q = r^{-1} \log_P R \pmod p$. The discrete logarithm of Q base P is solved. ■

V. The maximum likelihood decoding for AG-codes is NP-hard

The dimension of linear space $L((k-1)O)$ over $\mathbf{F}_p$ is $k-1$ for an elliptic curve E. The dimension of linear space $L(Q + (k-1)O)$, $Q \neq O$, is k. Let $f_1, f_2, \cdots, f_{k-1}$ be a basis for $L((k-1)O)$, and f' be a function in $L(Q + (k-1)O) - L((k-1)O)$. They can be founded quickly by Proposition 2.

Lemma 2: Consider the code generated by evaluating functions in $L((k-1)O)$ at $P_1, P_2, \cdots, P_n$. Suppose the received word is $R = (f'(P_1), f'(P_2), \cdots, f'(P_n))$, where f' is defined above. Then

1) the distance from R to the code is either $n-k+1$ or $n-k$
2) the distance from R to the code is $n-k$ iff there is a subset $\{P_{i_1}, \cdots, P_{i_k}\}$ of $\{P_1, P_2, \cdots, P_n\}$ such that

$$P_{i_1} + P_{i_2} + \cdots + P_{i_k} = Q$$

Proof:

It is clear that R is not a codeword, since if $f' \in L(Q + (k-1)O)$ takes the same values as a function in $L((k-1)O)$ at n distinct points, it must be equal to the function, but f' has a pole at Q.

If the distance is less than $n-k$, it means that there is a function $f \in L((k-1)O)$ such that $f' - f$ has more than k distinct zeros in $\{P_1, P_2, \cdots, P_n\}$. But $f' - f \in L(Q + (k-1)O)$, which has at most k poles. A contradiction.

If the distance from R to the code is $n-k$, there is a function $f \in L((k-1)O)$ such that $f' - f$ has k

distinct zeros. Let them be $P_{i_1}, \cdots, P_{i_k}$. The function $f' - f$ must have a pole at Q with multiplicity 1 and a pole at O with multiplicity $k-1$. Therefore, we have $(f'-f) = P_{i_1} + \cdots + P_{i_k} - Q - (k-1)O$ and in $E(\mathbf{F}_p)$

$$P_{i_1} + \cdots + P_{i_k} = Q.$$

In the other direction, if there is a subset $P_{i_1}, \cdots, P_{i_k}$ of $P_1, P_2, \cdots, P_n$ such that

$$P_{i_1} + P_{i_2} + \cdots + P_{i_k} = Q$$

This implies that there is a function g such that

$$(g) = P_{i_1} + \cdots + P_{i_k} - Q - (k-1)O.$$

It is clear that $g \in L(Q + (k-1)O)$, thus $g = f + af'$, where $f \in L((k-1)O)$ and $a \in \mathbf{F}_p^*$. The vector R is at distance $n-k$ away from the codeword obtained by evaluating the function $-f/a$ at $P_1, P_2, \cdots, P_n$.

To prove that the distance is at most $n-k+1$, compute $P' = Q - P_1 - P_2 - \cdots - P_{k-1}$. If $P' \in \{P_k, P_{k+1}, \cdots, P_n\}$, then we have shown that the distance from R to the code is $n-k$. Assume that it is not the case. Then there exists a function g' such that

$$(g') = P_{i_1} + \cdots + P_{i_{k-1}} + P' - Q - (k-1)O.$$

Since $g' \in L(Q + (k-1)O)$, we have that $g' = af' + f$ for some $f \in L((k-1)O)$ and $a \in \mathbf{F}_p^*$. This shows that the distance from R to the code is at most $n-k+1$. ■

Theorem 3: Given a received vector, computing the distance from the vector to an elliptic code is NP-hard under a randomized reduction. Therefore, the maximum likelihood decoding problem for algebraic geometry codes is NP-hard under a randomized reduction.

Proof: Given an instance of the elliptic curve subset sum problem, let f' be a function in $L(Q + (k-1)O) - L((k-1)O)$. Now consider an algebraic geometry code generated by evaluating functions in $L((k-1)O)$ at $P_1, P_2, \cdots, P_n$. According to Lemma 2, the answer to the elliptic curve subset sum instance is "Yes", iff the distance from $R = (f'(P_1), f'(P_2), \cdots, f'(P_n))$ to the code is $n-k$. ■

The preprocessing maximum likelyhood decoding problem asks whether there exist polynomial time maximum likelyhood decoding algorithms *dependent* on the codes. Applying the result about the preprocessing subset sum problem [12], we get

Corollary 3: There is a sequence of algebraic geometry codes $C_1, C_2, \cdots, C_i, \cdots$, where $C_i \in [i, k]_{q_i}$, such that the existence of polynomial size circuits which solve their maximum likelihood decoding problems implies that $NP \subseteq P/poly$.

VI. Concluding remarks

In this paper, we prove that computing minimum distances and the maximum likelihood decoding are NP-hard for algebraic geometry codes. Our results rule out the possibility of polynomial time solutions for these two problems, unless $NP = ZPP$.

The Reed-Solomon codes can be thought of as a special case of algebraic geometry codes, in which we use the rational function field. Let O be the infinity point on the projective line. The functions $1, x, x^2, \cdots, x^k$ form a basis for $L(kO)$. In [4], the authors study Hamming balls centered at the vectors $(r(x)/h(x))_{x \in \mathbf{F}_q}$, where r and h are polynomials, in order to prove that the bounded distance decoding for the Reed-Solomon codes is hard. The function $f(x)/h(x)$ has poles at a point other than O. In the proof of Lemma 2, we use f' to generate a received word, it has poles at a place other than O. We suspect that further exploration of this connection between rational functions with a different pole and decoding problems would prove fruitful.

Our results use algebraic geometry codes based on elliptic curves. In many ways, the elliptic codes are very similar to the Reed-Solomon codes. Intuitively we expect that the decoding problem for elliptic codes is the easiest among all algebraic geometry codes. We leave it as an open problem to prove that both problems are NP-hard for codes based on curves of any fixed genus. We conjecture that the maximum likelihood decoding is NP-hard even for a family of algebraic geometry codes with a fixed alphabet, and leave it as an open problem.

Acknowledgments

We thank Daqing Wan and Elizabeth Murray for helpful discussions.

References

[1] Len Adleman. Algorithmic number theory-the complexity contribution. In *Proc. 35th IEEE Symp. on Foundations of Comp. Science*, pages 88–113, 1994.

[2] Miklos Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proc. 33th ACM Symp. on Theory of Computing*, pages 601–610, 2001.

[3] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions of Information Theory*, 24(3):384–386, 1978.

[4] 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.

[5] Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Transactions on Information Theory*, 49(1):22–37, 2003.

[6] S. Goldwasser and J. Kilian. Primality testing based on elliptic curves. *Journal of the ACM*, 46(4):450–472, 1999.

[7] V. Guruswami and A. Vardy. Maximum-likelihood decoding of Reed-Solomon codes is NP-hard. *IEEE Transactions on Information Theory*, 51(7):2249–2256, 2005.

[8] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.

[9] D. R. Heath-Brown. The differences between consecutive primes. *J. London Math. Soc.*, 2(18):7–13, 1978.

[10] Ming-Deh Huang and Doug Ierardi. Efficient algorithms for the Riemann-Roch problem and for addition in the Jacobian of a curve. *Journal of Symbolic Computation*, 18:519–539, 1994.

[11] H. W. Lenstra. Factoring integers with elliptic curves. *Annals of Mathematics*, 126:649–673, 1987.

[12] Antoine Lobstein. The hardness of solving subset sum with preprocessing. *IEEE Transactions on Information Theory*, 36(4):943–946, 1990.

[13] R. Pellikaan, B.-Z. Shen, and G.J.M. van Wee. Which linear codes are algebraic-geometric? *IEEE Transactions on Information Theory*, 37(3):583–602, 1991.

[14] Joseph H. Silverman. *The arithmetic of elliptic curves*. Springer-Verlag, 1986.

[15] Alexander Vardy. The intractability of computing the minimum distance of a code. *IEEE Trans. Inform. Theory*, 43(6):1757–1766, 1997.

Qi Cheng received the B.S. degree from Nankai University in 1992, M.S. degree from Fudan University in 1995 and the Ph.D. degree in computer science from University of Southern California in 2001.

He joined the University of Oklahoma in Norman, OK in 2001, where he is now an Associate Professor of Computer science. His research interests include cryptography, coding theory and theoretical computer science.