Code-based Cryptosystem from Quasi-Cyclic Elliptic Codes

Fangguo Zhang[1,2] ⋆ and Zhuoran Zhang[1,2]

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
[2] Guangdong Key Laboratory of Information Security, Guangzhou 510006, China

Abstract. With the fast development of quantum computation, code-based cryptography arises public concern as a candidate of post quantum cryptography. However, the large key-size becomes a main drawback such that the code-based schemes seldom become practical although they performed pretty well on the speed of both encryption and decryption algorithm. Algebraic geometry codes was considered to be a good solution to reduce the size of keys, but because of its special construction, there have lots of attacks against them. In this paper, we propose a public key encryption scheme based on elliptic codes which can resist the known attacks. By using automorphism on the rational points of the elliptic curve, we construct quasi-cyclic elliptic codes, which reduce the key size further. We apply the list-decoding algorithm to decryption thus more errors beyond half of the minimum distance of the code could be correct, which is the key point to resist the known attacks for AG codes based cryptosystem.

Keywords: code-based cryptography, post quantum cryptography, quasi-cyclic code, elliptic code, list-decoding

1 Introduction

Since the introduction of public key cryptography in 1976 [12], many cryptosystems have been proposed. Most of the commonly used public key cryptosystems are based on the hardness of factoring or the presumed intractability of the discrete logarithm problem. However, with the discovering of Shor Algorithm [43] and the rapid development of quantum computer, the above problems together with many other problems which are thought to be difficult to solve, become not hard any more. Thus, how to build cryptosystems that can resist the attack from quantum computer, i.e. post-quantum cryptosystems, raises the researchers concern. There are several kinds of post-quantum cryptography, and the principal available techniques are code-based cryptography, lattice-based cryptography, multivariate cryptography and hash-based cryptography etc. Among them, the

⋆ Corresponding author, email: isszhfg@mail.sysu.edu.cn

original McEliece encryption scheme is a very strong candidate as one of the future post-quantum standards for public-key encryption.

The code-based McEliece system [29], whose security relies on the hardness of decoding a random linear code, is one of the best-known public key cryptosystem, and has already resisted 40 years of cryptanalysis since its proposing in 1978. Niederreiter [34] gave a variant of McEliece, and a significant amount of research went into analysing and improving them. One line of research was concerned with improving the direct decoding attacks that McEliece had outlined in his original paper, and with choosing the parameters that would maximize resistance against these attacks. Another line of research was concerned with modifying McEliece's construction in order to obtain a more powerful system. A third line was concerned with structural analysis, i.e. the study of the structure of the underlying codes in order to devise attacks against such cryptosystems [31].

There are many hard problems in coding theory, including general decoding problem, syndrome decoding problem, finding the minimum distance of a code, finding the minimum weight codeword and so on. [5] showed that the general decoding problem for linear codes and the general problem of finding the minimum weight codeword are both NP-complete. The problem of computing the minimum distance of a binary linear code is NP-hard, and the corresponding decision problem is NP-complete according to [47]. In addition, [20] prove that maximum-likelihood decoding is NP-hard for the family of Reed-Solomon codes. Besides, it has been proved that for elliptic codes, minimum distance problem and maximum likelihood decoding problem are NP-hard [8].

The code-based cryptography presents many advantages: it is very fast for both encryption and decryption and the best known attacks are exponential in the length of the code. However, due to the large key size required to reach a good security level, no practical application of codes-based cryptography is known to us. Nowadays, a very popular trend in code-based cryptography is to decrease the public-key size by focusing on subclasses of alternant/Goppa codes which admit a very compact public matrix, typically quasi-cyclic(QC) or quasi-dyadic(QD) generator matrices.

The first proposal who use quasi-cyclic codes can be traced back to [16] by Gaborit where quasi-cyclic BCH codes are suggested. This proposal was broken in [35], essentially because the number of possible keys was too low. There are also many other proposals based on quasi-cyclic codes and attacks against them. Usually, the attacks first build an algebraic model of the key recovery attack and using the Groebner basis techniques to drastically reduce the number of variables in the system when compares to the polynomial system associated to unstructured alternant or Goppa codes [2]. However, the security of the quasi-cyclic codes does not been break totally. In fact, the security closely rely on the chosen codes. In the submissions of NIST Post-Quantum cryptography standard, many code-based proposals use quasi-cyclic codes to reduce the key size and can still keep secure.

Algebraic geometry (AG) code was proposed by Goppa [17] in 1977 and was introduced into cryptography in 1996 by Janwa and Moreno [24]. Their original

idea to use AG codes was to decrease the so large block length of public key of McEliece scheme. Meanwhile, the AG codes not only contain the Goppa codes as a subclass but give immense choice because now one can vary the field, the genus, the curves of particular genus, and the rational points that are constituents of the divisors generation the codes. Besides, AG codes have structure that can be constructed by the divisor and the rational point set other than the generator matrix, which can decrease their storage space [7, 23, 31, 36, 42].

Unfortunately, when it comes to the AG codes, their special structure becomes a drawback as well, which results in many cryptanalysis towards it. In 1992, Sidelnikov and Shestakov [44] discovered a deterministic polynomial time structural attack against Niederreiter's proposal to use Reed-Solomon codes, i.e. AG codes with genus $g = 0$. In 2007, Minder [31] analysed the codes defined on elliptic curve whose genus $g = 1$ is not safe. Faure [15] then generalized this work into hyperelliptic curve, i.e. $g = 2$. Thus the codes defined on curves with genus $g \leq 2$ seems are all broken. However, their attack needs to find the minimum weight word in the giving code in the first step, which is considered to be a hard problem [8] if the code is not a maximum distance separable code. In 2014, Márquez-Corbella *et al.* proved that the structure of the curve can be recovered from the only knowledge of a generator matrix of the code [27,28], but the corresponding decoding algorithm is lacked. Recently, Pellikaan *et al.* [10] proposed a decoding attack use Error-Correct-Pairing (ECP) decoding algorithm and is efficient on codes from curves of arbitrary genus based on their previous work [9,22,38]. These attacks seems to warning us, AG code is not a good choice to construct cryptosystem.

However, after study all the attacks above, we find the fact that they all hold on the assumption that there is no more than half of the minimum distance errors occur. This may rise from that most decoding algorithm has error correct bounding $(d-1)/2$ where d denotes the minimum distance of the code. In 1999, Guruswami and Sudan [19] proposed a list decoding algorithm for both RS and AG codes which can correct more than $(d-1)/2$ errors in polynomial time. At the same time, we noticed that the information-set-decoding (ISD) algorithm which inspired nearly all general decoding algorithms, has a complexity bound connect tightly with the weight of errors [33]. Thus, as long as we choose rational parameters, especially the number of error weights, we can build a security code-based cryptosystem.

Our contributions: In this paper, we reconsider the application of algebraic geometry codes, especially elliptic codes, in cryptography, and propose a public-key encryption scheme based on elliptic codes. We introduce list decoding of elliptic codes into our scheme to prevent it from the known attacks. Moreover, we apply Tong's construction of quasi-cyclic elliptic codes [46] into our scheme to decrease the public key size further.

Organization: The rest of paper is organized as follows. In section 2, we review some preliminaries that will be used later. In section 3, we present our cryptosystem based on quasi-cyclic elliptic code. In section 4, we show the security analysis together with the parameters we recognized. In section 5, we

analyse the efficiency of our scheme. Last but not the least, section 6 draws the conclusion to this paper.

2 Preliminaries

In this section, we present the notions of coding theory that are prerequisite for the following chapters as well as basic knowledge about code-based cryptography.

2.1 Linear Codes and Code-Based Cryptography

We now recall some basic definitions for linear codes and code-based cryptography.

An $[n,k]_q$ linear error-correcting code $\mathcal{C}$ is a linear subspace of a vector space $\mathbb{F}_q^n$, where $\mathbb{F}_q$ 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 rank k which defines a linear mapping from $\mathbb{F}_q^k$ (called the message space) to $\mathbb{F}_q^n$. Namely, the code $\mathcal{C}$ is

$$\mathcal{C} = C(G) = \{xG \mid x \in \mathbb{F}_q^k\}.$$

If $\mathcal{C}$ is the kernel of a matrix $H \in \mathbb{F}_q^{(n-k)\times k}$, we call H a parity check matrix of $\mathcal{C}$, i.e.

$$\mathcal{C} = C^{\perp}(H) = Ker(H) = \{y \in \mathbb{F}_q^n | Hy = \mathbf{0}\}$$

We call a vector in $\mathcal{C}$ a codeword.

Given a codeword $\boldsymbol{c} = (c_1, c_2, \ldots, c_n) \in \mathbb{F}_q^n$, its Hamming weight $\mathsf{wt}(\boldsymbol{c})$ is defined to be the number of non-zero coordinates, i.e. $\mathsf{wt}(\boldsymbol{c}) = |\{i \mid c_i \neq 0, 1 \leq i \leq n\}|$. The distance of two codewords $\boldsymbol{c}_1, \boldsymbol{c}_2$, denoted by $d(\boldsymbol{c}_1, \boldsymbol{c}_2)$ counts the number of coordinates in which they differ. The minimum distance $d(\mathcal{C})$ of code $\mathcal{C}$ is the minimal value of the distance between any two different codewords. By the linearity of $\mathcal{C}$, we know that $d(\mathcal{C})$ is determined by the minimum Hamming weight among all non-zero codewords in $\mathcal{C}$, i.e.

$$d(\mathcal{C}) = min\{\mathsf{wt}(\boldsymbol{c}) \mid \boldsymbol{c} \in \mathcal{C} \setminus \{0\}\}.$$

If a linear $[n,k]_q$ code has d as the minimum distance, then $\mathcal{C}$ is called a $[n,k,d]_q$ linear code.

If $\boldsymbol{c}$ is a codeword and $\boldsymbol{c} + \boldsymbol{e}$ is the received word, then we call $\boldsymbol{e}$ the error vector and $\{i|e_i \neq 0\}$ the set of error positions and $\mathsf{wt}(\boldsymbol{e})$ is the number of errors of the received word. If $\boldsymbol{r}$ is the received word and the distance of $\boldsymbol{r}$ to the code $\mathcal{C}$ is t', then there exists a codeword $\boldsymbol{c}'$ and an error vector $\boldsymbol{e}'$ such that $\boldsymbol{r} = \boldsymbol{c}' + \boldsymbol{e}'$ and $\mathsf{wt}(\boldsymbol{e}') = t'$. If the number of errors is at most $(d-1)/2$, then it is sure that $\boldsymbol{c} = \boldsymbol{c}'$ and $\boldsymbol{e} = \boldsymbol{e}'$. In other words, the nearest codeword to $\boldsymbol{r}$ is unique when $\boldsymbol{r}$ has distance at most $(d-1)/2$ to $\mathcal{C}$.

Nowadays, most code-based cryptography are variants of either McEliece [29] public-key encryption system or Niederreiter [34]. We give a brief introduction of them together with the security assumptions here.

The McEliece system was proposed in 1978. Although in the original description, the secret key of the McEliece public-key encryption scheme is a Goppa code , the secret key could be drawn from any subclass of the class of the alternate codes. The trapdoor for the McEliece cryptosystem is the knowledge of an efficient error correcting algorithm for the chosen code class together with a permutation. Algorithm 1 shows the McEliece PKE scheme as follows:

Fig. 1. McEliece Public-key Encryption Scheme

Key Generation:	**Encryption:**
G: $k \times n$ generator matrix of code $\mathcal{C}$ with error correcting capability t	plaintext $\boldsymbol{m} \in \mathbb{F}^k$
S: $k \times k$ random non-singular matrix	$\boldsymbol{e} \leftarrow_{\$} \mathbb{F}^n$ of weight t
P: $n \times n$ random permutation matrix	$\boldsymbol{c} \leftarrow \boldsymbol{m}\mathsf{G}^{pub} \oplus \boldsymbol{e}$
$\mathsf{G}^{pub} \leftarrow \mathsf{SGP}$	**Decryption:**
$\mathsf{C.Decode}$: decoding algorithm for $\mathcal{C}$	$\boldsymbol{c}\mathsf{P}^{-1} = (\boldsymbol{m}\mathsf{S})\mathsf{G} \oplus \boldsymbol{z}\mathsf{P}^{-1}$
$pk \leftarrow < \mathsf{G}^{pub}, t >$	$\boldsymbol{m}\mathsf{SG} = \mathsf{C.Decode}(\boldsymbol{c}\mathsf{P}^{-1})$
$sk \leftarrow < \mathsf{S}, \mathsf{P}, \mathsf{C.Decode} >$	$\boldsymbol{m} = \mathsf{C.Decode}(\boldsymbol{c}\mathsf{P}^{-1})\mathsf{S}^{-1}\mathsf{G}^{-1}$

The Niederreiter's public-key cryptosystem [34], which can be seen as the dual variant of the McEliece PKC, was proposed in 1986. In difference to the McEliece cryptosystem, instead of representing the message as a codeword, Niederreiter proposed to encode it into the error vector. We summarize it in Figure 2.

Fig. 2. Niederreiter Public-key Encryption Scheme

Key Generation:	
H: $n \times (n-k)$ check matrix of code $\mathcal{C}$ with error correcting capability t	**Encryption:**
P: $n \times n$ random permutation matrix	$\boldsymbol{e} \leftarrow \phi_{n,t}(\boldsymbol{m}) \in \{0,1\}^n$, $\mathsf{wt}(\boldsymbol{e}) = t$
M: $k \times k$ matrix such that MHP is systematic	$\boldsymbol{c} = \mathsf{H}^{pub}\boldsymbol{e}$
$\mathsf{H}^{pub} \leftarrow \mathsf{MHP}$	**Decryption:**
$\mathsf{C.SDecode}$: syndrome decoding algorithm for $\mathcal{C}$	$\mathsf{M}^{-1}\boldsymbol{c} = \mathsf{HP}\boldsymbol{e}^T$
$pk \leftarrow < \mathsf{H}^{pub}, t >$	$\mathsf{P}\boldsymbol{e}^T = \mathsf{C.SDecode}(\mathsf{M}^{-1}\boldsymbol{c})$
$sk \leftarrow < \mathsf{M}, \mathsf{P}, \mathsf{C.SDecoding} >$	$\boldsymbol{e}^T = \mathsf{P}^{-1} \cdot \mathsf{C.SDecode}(\mathsf{M}^{-1}\mathsf{c})$

The advantage of this dual variant is the smaller public key size since it is sufficient to store the redundant part of the matrix H^{pub}. The disadvantage is the fact that the mapping $\phi_{n,t}$ slows down encryption and decryption.

2.2 Algebraic Geometry Code and Corresponding List-Decoding Algorithm

Algebraic geometry (AG) code was invented by Goppa [17] in 1977 as a natural generalization of the Reed-Solomon codes. We now give some notions will be used

below. Let $\mathbb{F}_q$ be a finite field with q elements and $\mathcal{X}$ be a smooth projective geometrically connected curve over a finite field $\mathbb{F}_q$ of genus g. The function field of $\mathcal{X}$ is denoted by $\mathbb{F}_q(\mathcal{X})$.

A divisor D on a curve $\mathcal{X}$ is a formal sum of points $D = \Sigma_P n_P P$ on the curve $\mathcal{X}$, where $n_P \in \mathbb{Z} \setminus \{0\}$ for a finite number of points on $\mathcal{X}$. Here n_P denotes the multiplicity of the point P on the curve. The degree of a divisor $D = \Sigma_P n_P P$ is defined as the sum of n_P, i.e., $deg(D) := \Sigma_P n_P$. The support of a divisor $supp(D)$ is the set of points with non-zero coefficients. A divisor is called effective if all coefficients are non-negative.

For each point $P \in \mathcal{X}$ and any $f \in \mathbb{F}_q(\mathcal{X}) \setminus \{0\}$, we can abstract the notion of evaluation of f at P (denoted by $v_P(f)$) by local parameter and discrete valuation function $v_P : \mathbb{F}_q(\mathcal{X}) \to \mathbb{Z} \cup \{\infty\}$. A point P is said to be a zero of multiplicity m if $v_P(f) = m > 0$, a pole of multiplicity $-m$ if $v_P(f) = m < 0$.

Any function $f \in \mathbb{F}_q(\mathcal{X}) \setminus \{0\}$ can be associated with a so-called principal divisor. The principle divisor of $f \in \mathbb{F}_q(\mathcal{X})$ is defined as $div(f) := \Sigma_P v_P(f) P$.

Let $G = \Sigma_P n_P P$ be any divisor of degree k on $\mathcal{X}$. Denote by $\mathcal{L}(G)$ all rational functions $f \in \mathbb{F}_q(\mathcal{X})$ such that the divisor $div(f) + G$ is effective, together with the zero function, i.e.,

$$\mathcal{L}(G) := \{f \mid div(f) + G \text{ is effective} \cup \{0\}\}.$$

By the Riemann-Roch theorem, $\mathcal{L}(G)$ is a vector space over $\mathbb{F}_q$ of finite dimension and its dimension is given by $dim(\mathcal{L}(G)) := k - g + 1$, where g is the genus of $\mathcal{X}$.

Given an irreducible curve $\mathcal{X}$ and the function field $\mathbb{F}_q(\mathcal{X})$ defined over $\mathcal{X}$, let $P_1, P_2, \ldots, P_n$ be distinct rational points on $\mathcal{X}$. The n points determine a divisor $D := P_1 + P_2 + \ldots + P_n$. Let G be an arbitrary divisor on $\mathcal{X}$ such that $\{P_1, P_2, \ldots, P_n\} \cap supp(G) = \emptyset$. An AG code $\mathcal{C}(D, G)$ is defined by the following injective mapping $\mathsf{ev} : \mathcal{L}(G) \to \mathbb{F}_q^n$ with

$$\mathsf{ev}(f) := (f(P_1), f(P_2), \ldots, f(P_n))$$

Hence $\mathcal{C}(D, G) = \mathsf{image}(\mathsf{ev})$. If $G = \Sigma_P n_P P$ is a divisor of degree k, then $\mathcal{C}(D, G)$ is an $[n, k - g + 1, d]_q$ code and $d \geq n - k + 1 - g$. The basic properties of AG codes can be found in [23].

List decoding is a powerful decoding algorithm with a long history. For any $[n, k, d]$ linear code, a well-known fact is that if the number of errors t satisfies $t \leq \lceil (d-1)/2 \rceil$, then there must exist a unique codeword within distance $\lceil (d-1)/2 \rceil$ from the received vector. Meanwhile, if $t > (d-1)/2$, unique decoding is usually impossible. In 1999, Guruswami and Sudan [18] proposed a list decoding algorithm for both RS and AG codes. The algorithm is able to efficiently output a list of codewords which lie in the sphere of radius up to $t = n - \sqrt{nk}$ centered around the received vector. More precisely, the list decoding algorithm $\mathsf{C.ListDecode}$ takes as input a linear $[n, k]$ code $\mathcal{C}$, a received vector $\boldsymbol{r}$ and a parameter $t \leq n - \sqrt{nk}$, and it outputs a list of codewords whose Hamming distances to $\boldsymbol{r}$ are at most t. Up to now, the list decoding algorithm is one of the most powerful decoding methods for AG codes.

The Guruswami-Sudan list decoding consists of three steps: initialization, interpolation and root finding. The details can be find in [18] and [19]. Here we give a brief description refer to [48].

The Guruswami-Sudan List Decoding Algorithm: C.ListDecode

- **Input:**
  An AG-code $\mathcal{C}_{\mathcal{L}}(D, G)$ determined by curve $\mathcal{X}$ over $\mathbb{F}_q$ and divisors $G = \alpha Q$ and D, a received vector $\boldsymbol{r} = (r_1, \ldots, r_n)$ and an error bound t, which determines the maximal number of coordinates in which a codeword disagrees with vector $\boldsymbol{r}$ in order for the codeword to be included on the output list.
- **Output:**
  A list $\Omega_{\boldsymbol{r}}$ of codewords such that $\mathsf{dis}(\boldsymbol{r}, \boldsymbol{c}) \leq t$.
- **Initialization:**
  0.1 $\Omega_{\boldsymbol{r}} := \emptyset$.
  0.2 Compute list decoding parameters l from n, t and g, where $l \geq \alpha$.
  0.3 Fix a **pole basis** $\{\phi_{j_1} : 1 \leq j_1 \leq l - g + 1\}$ of $\mathcal{L}(lQ)$ such that ϕ_{j_1} has at most $j_1 + g - 1$ poles at Q.
  0.4 For each P_i, $1 \leq i \leq n$, find a **zero basis** $\{\psi_{j_3,P_i} : 1 \leq j_3 \leq l - g + 1\}$ of $\mathcal{L}(lQ)$ such that P_i is a zero of ψ_{j_3,P_i} with multiplicity (or at least) $j_3 - 1$.
  0.5 Compute the set $\{a_{P_i,j_1,j_3} \in \mathbb{F}_q : 1 \leq i \leq n, 1 \leq j_1, j_3 \leq l - g + 1\}$ such that for every i and every j_1, we have $\psi_{j_1} = \Sigma_{j_3} a_{P_i,j_1,j_3} \psi_{j_3,P_i}$.
- **Interpolation:**
  Set $s = \frac{l-g}{\alpha}$. Find a non-zero polynomial $H \in \mathcal{L}(lQ)[T]$ of the form

$$H[T] = \sum_{j_2=0}^{s} \sum_{j_1=1}^{l-g+1-\alpha j_2} h_{j_1,j_2} \phi_{j_1} T^{j_2}$$

- **Root Finding:**
  Find all roots $h \in \mathcal{L}(\alpha Q) \subseteq \mathcal{L}(lQ)$ of $H[T]$. For each h, check if $h(P_i) = r_i$ for at least $n - t$ values of $i \in \{1, 2, \ldots, n\}$, and if so, put h in $\Omega_{\boldsymbol{r}}$.
- **Return** $\Omega_{\boldsymbol{r}}$.

2.3 The Construction of Quasi-Cyclic Elliptic Code

Consider the curve defined by the projective solutions to the Weierstrass equation $\mathcal{E} : Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$. If the parameters a_i are such that the curve is smooth, then it is an elliptic curve. If $\mathrm{char}(\mathbb{F}_q) \neq 2, 3$, the Weierstrass equation can be taken in the form

$$\mathcal{E} : Y^2 = x^3 + aX + b$$

up to a coordinate transformation of the form

$$X = u^2X' + r,\ Y = u^3Y' + su^2X' + w$$

with $r, s, u, w \in \mathbb{F}_q, u \neq 0$. Specifically, if the equation has form $Y^2 = X^3 + b$ or $Y^2 = X^3 + aX$, their only automorphism are of the from $(X, Y) \mapsto (u^2X, u^3Y)$ with $u^6 = 1$ and $u^4 = 1$ respectively. More details in [45].

Let $\mathcal{E}$ be an elliptic curve over $\mathbb{F}_q$ and $\mathbb{F}_q(\mathcal{E})$ be the elliptic function field, then there exists an additive abelian group $\mathcal{E}(\mathbb{F}_q)$ with the group operation defined by the "chord-and-tangent" rule on $\mathcal{E}$.

Let $P_1, P_2, \ldots, P_n \in \mathcal{E}(\mathbb{F}_q)$. Define $D := P_1 + P_2 + \ldots + P_n$ be divisors on $\mathcal{E}$. Let G be another divisor on $\mathcal{E}$ such that $0 < deg(G) = k < n$ and $supp(D) \cap supp(G) = \emptyset$. The elliptic code $\mathcal{C}(D, G)$ is defined by G and D with

$$\mathcal{C}(D, G) := \{(f(P_1), \ldots, f(P_n) \mid f \in \mathcal{L}(G)\} \subseteq \mathbb{F}_q^n.$$

The construction of quasi-cyclic elliptic code is proposed by Tong and Ding [46]. Here we summarize the needed facts as follows.

Let $\mathcal{E}$ be an elliptic curve over $\mathbb{F}_q$ with coefficients a, b. Let $\mathcal{O}$ be the infinity point of $\mathcal{E}$, $\mathrm{Aut}(\mathcal{E})$ be the automorphism group of $\mathcal{E}$.

Let $\mathcal{C}$ be a linear code with length lm over $\mathbb{F}_q$. Let

$$c = (c_{0,0}, c_{0,1}, \ldots, c_{0,l-1}, c_{1,0}, \ldots, c_{1,l-1}, \ldots, c_{m-1,0}, \ldots, c_{m-1,l-1})$$

denotes a codeword in $\mathcal{C}$. Then

$$c' = (c_{0,l-1}, c_{0,1}, \ldots, c_{0,l-2}, \ldots, c_{m-1,l-1}, c_{m-1,0}, \ldots, c_{m-1,l-2},) \in \mathcal{C}.$$

$\mathcal{C}$ is called a quasi-cyclic(QC) code over $\mathbb{F}_q$ of length lm and index l, and m is called the co-index of $\mathcal{C}$.

Let $\sigma \in \mathrm{Aut}(\mathcal{E})$, $\sigma(\mathcal{O}) = \mathcal{O}$. The action on the $P_1, P_2, \ldots, P_n \in \mathcal{E}(\mathbb{F}_q)$ gives rise to orbits that contains $\mathrm{ord}(\sigma)$ elements, and some orbits which one contains t elements, where $t|\mathrm{ord}(\sigma)$. From results of algebraic geometry codes, we have

Theorem 1 [46] *Let $\sigma \in \mathrm{Aut}(\mathcal{E})$ with $\mathrm{ord}(\mathcal{O}) = \mathcal{O}$. Let $S_1, \ldots, S_l$ be l distinct orbits of length $\mathrm{ord}(\sigma)$. The divisor D and G are defined by*

$$D = \sum_{i=1}^{l} \sum_{P_i \in S_i} P_i, , G = k\mathcal{O}$$

where $0 < k < l\mathrm{ord}(\sigma)$. Algebraic geometry codes

$$\mathcal{C}_{\mathcal{L}}(D, G) = (f(P_1), f(P_2), \ldots, f(P_l), \ldots,$$
$$f(\sigma^{\mathrm{ord}(\sigma)-1}P_1), f(\sigma^{\mathrm{ord}(\sigma)-1}P_2), \ldots, f(\sigma^{\mathrm{ord}(\sigma)-1}P_l)), f \in \mathcal{L}(G)$$

are $[l\mathrm{ord}(\sigma), k]$ QC NMDS codes with co-index $\mathrm{ord}(\sigma)$, where $\mathcal{L}(G)$ is the Riemann-Roch space associated to G.

Obviously, every quasi cyclic code $\mathcal{C}$ must have a quasi cyclic generator matrix. To find this QC generator matrix, we only need to find some codeword $c_1, \ldots, c_t$ in $\mathcal{C}$, such that their quasi cyclic shift are linear independent. Once this QC generator matrix is found, we can use the first row of each block to recover the generator matrix, and thus the $n \times k$ public generator can be decreased into a $n \times (k/\mathrm{ord}(\sigma))$ matrix with compact from .

3 The McEliece-Based Cryptosystem from Quasi-Cyclic Elliptic Codes

In this section, we will propose our scheme based on the quasi-cyclic elliptic codes and McEliece encryption scheme. We first give the construction of a basic scheme, and then give a more efficient scheme with key-encapsulation mechanism. Last but not least, we show how to transform the basic scheme into the dual Niederriter version.

3.1 The Basic Scheme

Our PKE scheme based on QC elliptic codes can be summarized as follows:

- **Set up**(1^λ)
  Generates the global parameters $\mathbf{param} = (q, n, k, t)$, where q is the size of the finite field, n is the length of the code and k is the dimension if the code. Denote $t = t_0 + t_1$ as the number of errors that are add to the codeword, where $t_0 = (d-1)/2 = (n-k-1)/2$ and $t_1 < n - \sqrt{nk} - t_0$ such that $\mathsf{C.ListDecode}$ can correct t errors.
- **KeyGen(param)**
  Take **param** as input, and output the key pair $(\mathsf{pk}, \mathsf{sk})$.
  Firstly construct a finite field $\mathbb{F}_q$ with q elements, where q is a prime number. Afterwards randomly choose an element $b \in \mathbb{F}_q$, then the elliptic curve is given by $\mathcal{E} : Y^2 = X^3 + b$. The restriction on b is that there must exist an automorphism function σ on $\mathcal{E}$ with $\mathrm{ord}(\sigma) = 6$.
  Secondly, choose $l = n/6$ rational points $P_D = \{P_1, P_2, \ldots, P_l \mid P_i \neq \mathcal{O}\}$ on $\mathcal{E}$ on different orbits corresponding to σ. Set

$$D = \Sigma_{i=1}^{l} \Sigma_{j=0}^{j=5} \sigma^j(P_i)$$

  and $G = k\mathcal{O}$. Then $\mathcal{C}(D, G)$ is a quasi-cyclic elliptic curve code defined on $\mathcal{E}$ over $\mathbb{F}_q$. Let G^{pub} be the generator matrix of $\mathcal{C}$ with compact form.

$$\mathsf{pk} \leftarrow (\mathsf{G}^{pub}, t)$$
$$\mathsf{sk} \leftarrow (\mathcal{E}, P_D, \sigma)$$

- **Encrypt**$(\mathsf{pk}, \boldsymbol{m})$
  Take the public key pk and message $\boldsymbol{m} \in \mathbb{F}_q^k$ as input, and output the cipher-text $\boldsymbol{c}$.
  Firstly, randomly choose a vector $\boldsymbol{r} \in \mathbb{F}_q^k$, and another random vector $\boldsymbol{e}$ in $\mathbb{F}_q^n$ of weight t.
  Then the cipher-text is calculated as

$$\boldsymbol{c_1} \leftarrow \boldsymbol{r}\mathsf{G} + \boldsymbol{e}$$
$$\boldsymbol{c_2} \leftarrow \boldsymbol{m} + \boldsymbol{r}$$
$$\boldsymbol{c} \leftarrow \boldsymbol{c_1} || \boldsymbol{c_2}$$

– **Decrypt**$(\mathsf{sk}, \boldsymbol{c})$

Take the secret key sk and cipher-text $\boldsymbol{c}$ as input, and output the message $\boldsymbol{m}$.

Once get the cipher-text, depart it into two parts with length n and k respectively to get $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$.

Then run the list-decoding algorithm to calculate

$$\boldsymbol{r} \leftarrow \mathsf{C.ListDecode}(\boldsymbol{c_1})$$

And finally will get

$$\boldsymbol{m} \leftarrow \boldsymbol{c_2} - \boldsymbol{r}$$

Correctness: Notice that the correctness of our scheme relies on the success of list decoding algorithm about $\mathcal{C}$. To avoid the situation that there are more than one codeword are returned in the list, we can add some redundancies to the original message. There are many analysis about the number of codeword in the output list, like [30] for Reed-Solomon codes and [48] for elliptic codes etc. Acutually, as showed in [6], in CCA2-secure variants of McElieces system there is no difficulty in identifying which codeword is a valid message. Once the decoding algorithm success and a unique codeword is decided, then for a given message $\boldsymbol{m}$ and its corresponding ciphertext $\boldsymbol{c}$, we have

$$\boldsymbol{c_2} - \mathsf{C.ListDecode}(\boldsymbol{c_1}) = \boldsymbol{c_2} - \boldsymbol{r} = \boldsymbol{m} + \boldsymbol{r} - \boldsymbol{r} = \boldsymbol{m},$$

i.e.

$$\mathbf{Decrypt}(\mathsf{sk}, (\mathbf{Encrypt}(\mathsf{pk}, \boldsymbol{m})) = \boldsymbol{m},$$

which shows the correctness of our scheme.

The quasi-cyclic generator matrix of the underlying code can be stored in a brief from, which leads to a smaller size of the public key. Here we use list decoding algorithm as a subroutine in the decryption algorithm, because the traditional unique decoding algorithm is impossible to correct more than $(d-1)/2$ errors. Our basic scheme can be transformed into a CCA2-secure version by the universal method as mentioned in [25].

3.2 A More Efficient Scheme

To make our scheme more practical, we propose the corresponding key encapsulation mechanism (KEM) refer to [21] as follows.

Alice and Bob want to share a common session secret key K. Bob publishes his public key $\mathsf{pk} = (\mathsf{G}^{pub}, t)$, and his secret key is denoted as $\mathsf{sk} = (\mathcal{E}, P_D, \sigma)$. Besides, choose security hash functions $\mathcal{H}, \mathcal{K}, \mathcal{F}$.

– **Encap:**

Alice randomly chooses a vector $\boldsymbol{m} \in \mathbb{F}_q^n$. Then run the **Encrypt** algorithm with Bob's public key pk and $(\boldsymbol{m}||\mathcal{H}(\boldsymbol{m}))$. The output cipher-text is denoted as $\boldsymbol{c}$. Set

$$\boldsymbol{d} \leftarrow \mathcal{F}(\boldsymbol{m})$$

Alice sends $(\boldsymbol{c}, \boldsymbol{d})$ to Bob.
The session key is defined as

$$K \leftarrow \mathcal{K}(\boldsymbol{m}||\boldsymbol{c})$$

– **Decap:**
Bob receives $(\boldsymbol{c}, \boldsymbol{d})$. Then run the **Decrypt** algorithm with his secret key pk and $\boldsymbol{c}$. Denote the output as $(\boldsymbol{m}^*, \mathcal{H}(\boldsymbol{m}^*))$. Bob computes $\boldsymbol{c}^* \leftarrow$ **Encrypt**$(\mathsf{pk}, \boldsymbol{m}^*||\mathcal{H}(\boldsymbol{m}^*))$ and $\boldsymbol{d}^* \leftarrow \mathcal{F}(\boldsymbol{m}^*)$.
If $\boldsymbol{c}^* = \boldsymbol{c}$ and $\boldsymbol{d}^* = \boldsymbol{d}$, Bob computes the session key

$$K \leftarrow \mathcal{K}(\boldsymbol{m}^*||\boldsymbol{c}^*)$$

Else return *false*.

According to [21], the above KEM version of our scheme is IND-CCA2.

3.3 Transformation into Niederreiter Version

As mentioned before, Niederreiter encryption scheme is actually a dual version of McEliece scheme, so we can easily transform our scheme into Niederreiter version. It is well-known that a decoding algorithm can be transformed into a syndrome decoding algorithm, and list decoding is no exception. [3] shows the details for syndrome list decoding. We denote the syndrome decoding algorithm for code $\mathcal{C}$ as $\mathsf{C.SListDecoding}$.

– **Set up**(1^λ)
Generates the global parameters **param** $= (q, n, k, t, \mathcal{H})$, where q is the size of the finite field, n is the length of the code, and k is the dimension if the code. Denote $t = t_0 + t_1$ as the number of errors that are add to the codeword, where $t_0 = (d-1)/2 = (n-k-1)/2$ and $t_1 < n - \sqrt{nk} - t_0$ such that there exists an efficient syndrome list decoding algorithm $\mathsf{C.SListDecoding}$ can correct t errors. A security hash function $\mathcal{H} : \{0,1\}^* \leftarrow \{0,1\}^k$ is selected as well.
– **KeyGen(param)**
Take **param** as input, and output the key pair $(\mathsf{pk}, \mathsf{sk})$.
Firstly construct a finite field $\mathbb{F}_q$ with q elements, where q is a prime number. Afterwards randomly choose an element $b \in \mathbb{F}_q$, then the elliptic curve is given by $\mathcal{E} : Y^2 = X^3 + b$. The restriction on b is that there must exist an automorphism function σ on $\mathcal{E}$ with $\mathrm{ord}(\sigma) = 6$.
Secondly, choose $l = n/6$ rational points $P_D = \{P_1, P_2, \ldots, P_l \mid P_i \neq \mathcal{O}\}$ on $\mathcal{E}$ on different orbits corresponding to σ. Set

$$D = \Sigma_{i=1}^{l} \Sigma_{j=0}^{j=5} \sigma^j(P_i)$$

and $G = k\mathcal{O}$. Then $\mathcal{C}(D, G)$ is a quasi-cyclic elliptic curve code defined on $\mathcal{E}$ over $\mathbb{F}_q$. Let H^{pub} be the parity check matrix of $\mathcal{C}$ with compact form.

$$\mathsf{pk} \leftarrow (\mathsf{H}^{pub}, t)$$
$$\mathsf{sk} \leftarrow (\mathcal{E}, P_D, \sigma)$$

– **Encrypt**$(\mathsf{pk}, \boldsymbol{m})$
  Take the public key pk and message $\boldsymbol{m} \in \mathbb{F}_q^n$ as input, and output the cipher-text $\boldsymbol{c}$.
  Choose a random vector $\boldsymbol{z} \in \mathbb{F}_q^{n-k}$ of weight t, and calculate

$$\boldsymbol{c_1} = \mathsf{H}\boldsymbol{z}^T$$
$$\boldsymbol{c_2} = \boldsymbol{m} + \mathcal{H}(\boldsymbol{z})$$
$$\boldsymbol{c} = \boldsymbol{c_1} || \boldsymbol{c_2}$$

– **Decrypt**$(\mathsf{sk}, \boldsymbol{c})$
  Take the secret key sk and cipher-text $\boldsymbol{c}$ as input, and output the message $\boldsymbol{m}$.
  Once get the cipher-text $\boldsymbol{c}$, depart it into to two parts with equal length ($\boldsymbol{c_1}$ and $\boldsymbol{c_2}$). Then run the syndrome list-decoding algorithm to get

$$\boldsymbol{z} \leftarrow \mathsf{C.SListDecode}(\boldsymbol{c_1})$$

  And finally get

$$\boldsymbol{m} \leftarrow \boldsymbol{c_2} - \mathcal{H}(\boldsymbol{z}).$$

Here we choose a random vector with weight t and add the hash of it on the message instead of using the mapping $\phi_{n,t}$ to encode the message into a error vector, thus the scheme will be more efficient.

4 Security Analysis and Parameters

The two most important types of attacks against code-based cryptosystems are structural attacks and decoding attacks. Structural attacks exploit structural weaknesses in the construction, and then attempt to recover the secret key. Decoding attacks are used to decrypt a given ciphertext. In this section, we will show how our system resist the known attacks and then give our parameters.

4.1 Information-Set-Decoding

Information-Set-Decoding is an approach introduced by Prange [41]. The idea is to find a set of coordinates of a garbled vector which are error-free and such that the restriction of the codes generator matrix to these positions is invertible. Then, the original message can be computed by multiplying the encrypted vector by the inverse of the submatrix. Peters [10] generalised the ISD algorithm over $\mathbb{F}_2$ to $\mathbb{F}_q$, afterwards Niebuhr *et al.* [33] optimized it and show a lower bounds for their ISD algorithm.

Let n be the length of the code $\mathcal{C}$ over $\mathbb{F}_q$, k be the dimension and $r = n - k$ be the co-dimension. To correct t errors, the lower bound for the expected cost in the binary operation of the algorithm is

$$WF_{qISD}(n,k,t,q) = min_p \frac{1}{\sqrt{q-1}} \cdot \frac{2l \min\left(\binom{n}{t}(q-1)^t, q^r\right)}{\lambda_q \binom{r-l}{t-p}\binom{k+l}{p}(q-1)^t} \cdot \sqrt{\binom{k+l}{p}(q-1)^p}$$

with $l = \log_q\left(K_q\lambda_q\sqrt{\binom{k}{p}(q-1)^{p-1}}\cdot\ln(q)/2\right)$ and $\lambda_q = 1-\exp(-1) \approx 0.63$. Noticed that the functions above is associated with n, k and t very tightly. Thus so long as we choose appropriate parameters such that the complexity of the above algorithm is beyond the security level, our scheme will reach the security level.

Example 1 To reach 2^{128} security level we choose parameters $q = 809$ and $[n, k] = [372, 156]$, then $r = n - k = 216$. Set $t = 125 < n - \sqrt{nk}$. Take them into the equation above, we found that when $p = 1$ the right side of the equation gets the minimum value.

$$WF_{qISD}(372, 156, 125, 1021) = \frac{1}{q-1}\cdot\frac{2l\binom{n}{t}}{\lambda_q\binom{r-l}{t-1}\binom{k+l}{1}}\cdot\sqrt{\binom{k+l}{1}(q-1)}$$

$$= \frac{1}{808}\cdot\frac{2l\binom{372}{125}}{\lambda_q\binom{215-l}{124}}(156+l)\cdot\sqrt{(156+l)808} \approx 2^{128.955},$$

here $l = \log_q\left(K_q\lambda_q\sqrt{\binom{k}{p}(q-1)^{p-1}}\cdot\ln(q)/2\right) \approx 1.313$.

4.2 Structure Attack

Because of the specially algebraic structure of AG codes, many researchers try to recover the secret key from the parameters and public key.

Minder [31] claimed that they devised an effective structural attack against the McEliece cryptosystem based on algebraic geometry codes defined over elliptic curves. This attack is inspired by an algorithm due to Sidelnikov and Shestakov [44] which solves the corresponding problem for Reed-Solomon codes.

However, the first step of attack, i.e. recovering the group structure, whose idea is to use the fact that minimum weight codewords correspond to functions whose divisor is exactly known, is unlikely to be done. Although we have known that the minimum distance of the used elliptic codes is $n-k$, as is said in [31], the asymptotic (in n) approximation of the probability that a fixed word of weight w shows still remains

$$2^{n[(1-R)H_q(\frac{\omega}{1-R})-H_q(\omega)]}$$

where $R = k/n$, $\omega = w/n$ and

$$H_q(x) = \begin{cases} x\log_q(q-1) - x\log_q(x) - (1-x)\log_q(1-x) & \text{if } 0 < x < 1-q^{-1} \\ 0 & \text{if } x = 0 \end{cases}$$

which means finding the needed minimum weight words has a exponent complexity. Moreover, Cheng [8] proved that the minimum distance problem is NP-hard. Since a codeword of minimum weight uniquely determine the minimum distance of this linear code, it indicated that it is unlikely to find a minimum weight word in elliptic codes in polynomial time. As a result, their attack is not efficient enough to break our scheme.

In [28] and [27], Márquze-Corbella *et al.* showed an attack which can recover an equivalent algebraic geometry code with the underlying one. They proved that the structure of the code can be recovered from the only knowledge of a generator matrix of the code. Although they showed efficient computational approach to the rational points and divisor finding algorithm, decoding algorithm from the obtained code's representation is still lacking. Thus, this result does not lead to an efficient attack.

4.3 Error Correcting Pair (ECP) Decoding

The attack with Error-Correcting-Pair was proposed by Pellikaan *et al.* in [10]. The ECP finding algorithm was able to be computed in $O(n^4)$ operations in $\mathbb{F}_q$, which allows the attacker to decrypt any encrypt message in $O(n^3)$ under the assumption that the users also use error correcting pairs.

Given a positive integer t, a t-ECP for a linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is a pair of linear codes (A, B) in $\mathbb{F}_q^n$ satisfying the following conditions:
(1) $(A * B) \subseteq \mathcal{C}^{\perp}$
(2) $d(B^{\perp}) > t$
(3) $k(A) > t$
(4) $d(A) + d(\mathcal{C}) > n$
where $A * B$ denotes the Schur product of A and B.

For a public matrix of AG code $\mathcal{C}_{\mathcal{L}}(D, G)^{\perp}$, suppose $D = P_1 + \cdots + P_n$. one can compute an ECP for $\mathcal{C}_{\mathcal{L}}(D - P, G - P)$, where $P \in \{P_1, \ldots, P_n\}$, i.e. an ECP for $\mathcal{C}_{\mathcal{L}}(D, G)^{\perp}$ punctured at one position. Thus the decoding can be can be performed by first correcting errors on the punctured code and then correct an erasure, with the help of P-Filtrations technique.

The decoding algorithm comes from [37]. We first give some definitions and then show the algorithm.

Let $\mathcal{C}$ be a linear code over $\mathbb{F}_q$. Define the syndrome map of the code $\mathcal{C}$ by

$$s : \mathbb{F}_q^n \to (\mathcal{C}^{\perp})^{\vee}$$

$$\boldsymbol{w} \mapsto (\boldsymbol{v} \mapsto \langle \boldsymbol{v}, \boldsymbol{w} \rangle)$$

where $\mathcal{C}^{\vee}$ denotes the vector space of $\mathbb{F}_q$ linear functionals on $\mathcal{C}$.

Let A, B and C be linear codes in $\mathbb{F}_q^n$. Define the error locating map E_w of a received word $\boldsymbol{w}$ with respect to the code C, by

$$E_w : A \to B^{\vee}$$

$$\boldsymbol{a} \mapsto (\boldsymbol{b} \mapsto \langle \boldsymbol{w}, \boldsymbol{a} * \boldsymbol{b} \rangle)$$

Suppose $I = i_1, \ldots, i_t$, where $1 \leq i_1 < \ldots < i_t \leq n$. Let A be a linear code in $\mathbb{F}_q^n$. Define

$$A(I) = \{a \in A \mid a_i = 0 \text{for all } i \in I\}$$

Define the projection map

$$\pi_I : \mathbb{F}_q^n \to \mathbb{F}_q^t$$

by $\pi_I(w) = (w_{i_1}, \ldots, w_{i_t})$. Define $A_I = \pi_I(A)$. Let $\boldsymbol{e} \in \mathbb{F}_q^n$. We will denote $\pi_I(e * A)$ by eA_I.

Define the inclusion map

$$i_I : \mathbb{F}_q^t \to \mathbb{F}_q^n$$

by inserting w_j at the i_jth coordinate for all $j = 1, \ldots, t$ and zeros everywhere else, for $w \in \mathbb{F}_q^n$. Note that $\pi_I \circ i_I$ is the identity map on $\mathbb{F}_q^t$. Define the restricted syndrome map

$$s_I : \mathbb{F}_q^t \to (C^\perp)^\vee$$

by the composition $s_I = s \circ i_I$.

The ECP Decoding Algorithm [37]

(1.1) Compute $\mathrm{Ker}(E_w)$.
(1.2) If $\mathrm{Ker}(E_w) = 0$, then goto (3.2).
(1.3) If $\mathrm{Ker}(E_w) \neq 0$, then choose a non-zero element $\boldsymbol{a}$ of $\mathrm{Ker}(E_w)$. Let $J = z(\boldsymbol{a}) = \{i \mid a_i = 0\}$.

(2.1) Compute the space of solutions of $s_J(\boldsymbol{x}) = s(\boldsymbol{w})$.
(2.2) If $s_J(\boldsymbol{x}) = s(\boldsymbol{w})$ has no or more than one solution then goto (3.2).
(2.3) If $s_J(\boldsymbol{x}) = s(\boldsymbol{w})$ has the unique solution $\boldsymbol{x}_0$, then compute $wt(\boldsymbol{x}_0)$.
(2.4) If $wt(\boldsymbol{x}_0) > t$, then goto (3.2).

(3.1) Print: "The received word is decoded by:"; Print $\boldsymbol{w} - i_J(\boldsymbol{x}_0)$; then goto (4.1).
(3.2) Print: "The received word has more than t errors ".

(4.1) End.

Simply speaking, one sets up a system of linear equations with the help of the vector space A and B. The set of zeros of a non-zero solution of these equation contains the error positions. Solving a set of linear equations involving the syndrome of the received word gives the error values.

A t-error correcting pair can correct any word with at most t errors. However, there is a limitation to t that $t \leq (d^* - 1)/2$ where d^* is the design distance for the code, which means that if there are more than $(d^* - 1)/2$ errors added to the codeword, the ECP decoding algorithm will fail because the dimension of A is exactly the maximum number of errors, and $dim(A) = t + g < n - k - t$ by definition. [37] also proved a corollary that if a linear code $\mathcal{C}$ has a t-ECP, then $t \leq \lfloor (d(\mathcal{C}) - 1)/2 \rfloor$. Having noticed that, we add more than $(d^* - 1)/2$ random errors in the encryption algorithm to defence the ECP attack, and use List-Decoding algorithm in the decryption algorithm to recover the codewords.

Then there comes another combined attack that first exhaustive search for the added errors, then use ECP-attack, i.e. split the error $\boldsymbol{e}$ into $\boldsymbol{e_1}$ and $\boldsymbol{e_2}$ such that $\boldsymbol{e_1} + \boldsymbol{e_2} = \boldsymbol{e}$ and $wt(\boldsymbol{e_1}) = \frac{d^*-1}{2} = t_0$. Denote the weight of $\boldsymbol{e_2}$ by t_1 with $t_1 = t - t_0$. The random errors may happen on any position, then there are $\binom{n}{t_1}$ possibilities for the error position, and $(q-1)^{t_1}$ possibilities for the error value, which inspire us to make sure our parameters will hold

$$\binom{n}{t_1}(q-1)^{t_1}$$

beyond the security level.

4.4 Other attacks

Exhaustive search to recover the structure of codes There are 3 parameters are needed to recover the code, i.e. b for elliptic curve, D and σ for elliptic code. Another divisor $G = k\mathcal{O}$ is actually public.

The elliptic curve has one parameter in $\mathbb{F}_q$, there are at most 6 automorphism functions, and at least $1+q-2\sqrt{q}$ points on a curve over $\mathbb{F}_q$ and can be sort out by the orbits of automorphism σ. Hence to recover the AG code, the attacker has to try

$$6q\binom{(1+q-2\sqrt{q})/6}{n/6}$$

times.

Attacks against quasi-cyclic codes There are also some attacks against QC codes. [14] gives algebraic cryptanalysis using Groebner basis of the QC McEliece variant based on [4] for the special structure of both Goppa codes and large order of quasi cyclicity. This attack relies on the construction of a specific party-check matrix of the underlying code, which differs from our QC elliptic codes. [35] attacks cryptosystems based on QC-BCH codes and QC-LDPC codes, mainly because those codes provide a low number of possible keys. We can choose appropriate parameters to make sure the key space is large enough to defence this attack.

4.5 Proposition of Parameters

In the following tables we use notations

1^λ: security level;

q: the prime to generate the finite field;

n: length of the quasi-cyclic code;

k: dimension of the quasi-cyclic code;

t_0: $t_0 = (d-1)/2$ is the basic weight of error vector;

t_1: the added weight of error vector;

t: $t = t_0 + t_1$ is the total weight of error vector, which can be decode by a list-decoding algorithm;

We give our suggested parameters in **Table 1**.

Now we give a example with security level 2^{128} to show that how our scheme resist the known attacks.

Example 2 We choose $q = 811$, $n = 372$, $k = 156$ and $t = 125$, the corresponding quasi-cyclic elliptic codes has minimum distance $d = n - k = 216$, the

Table 1. Parameters for Quasi-Cyclic Elliptic Codes based PKE

1^λ	q	n	k	t_0	t_1	t
2^{128}	811	372	156	108	17	125
2^{196}	1103	564	228	168	28	196
2^{256}	1493	714	342	180	32	212

extra number of errors is $t_0 = 17 < n - \sqrt{nk} - (d-1)/2$. We choose the elliptic curve parameter $b = 13$, and there are 793 points on the curve

$$\mathcal{E} : Y^2 = X^3 + 13$$

There are six automorphism functions on $\mathcal{E}$, and we choose $\sigma : (X, Y) \mapsto (130X, 810Y)$ with $\mathrm{ord}(\sigma) = 6$. Choose rational points

P_D = {(368 , 704), (788 , 245), (660 , 84), (14 , 18), (98 , 88), (364 , 808), (92 , 366), (214 , 400), (784 , 627), (37 , 437), (536 , 354), (112 , 794), (717 , 338), (268 , 305), (209 , 717), (800 , 149), (446 , 438), (52 , 72), (335 , 102), (516 , 136), (627 , 283), (686 , 764), (456 , 213), (541 , 226), (642 , 613), (506 , 410), (741 , 103), (142 , 188), (83 , 171), (550 , 532), (569 , 361), (546 , 298), (421 , 191), (226 , 481), (152 , 512), (462 , 474), (257 , 53), (377 , 489), (519 , 25), (7 , 60), (790 , 22), (320 , 619), (340 , 549), (416 , 667), (716 , 26), (658 , 385), (10 , 536), (682 , 64), (729 , 526), (325 , 131), (286, 692), (698 , 286), (323 , 577), (397 , 439), (515 , 231), (484 , 534), (168 , 718), (197 ,781), (148 , 628), (67 , 742), (571 , 157), (70 , 631)}

then set

$$D = \Sigma_{i=1}^{l} \Sigma_{j=0}^{j=5} \sigma^j(P_i),\ G = k\mathcal{O}.$$

Thus we construct a QC code $\mathcal{C}(D, G)$ with a 372×26 generator matrix in compact from

$$\mathbf{G}^{pub} = \begin{pmatrix} 649 & 595 & 743 & 199 & 761 & 229 & 303 & 525 & 703 & 330 & \dots & 632 & 621 & 172 & 661 \\ 583 & 30 & 682 & 702 & 66 & 721 & 126 & 287 & 630 & 358 & \dots & 386 & 422 & 372 & 110 \\ 554 & 97 & 576 & 368 & 463 & 586 & 102 & 229 & 676 & 331 & \dots & 398 & 182 & 261 & 251 \\ 581 & 314 & 479 & 365 & 293 & 700 & 292 & 43 & 449 & 436 & \dots & 78 & 465 & 18 & 693 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 197 & 139 & 785 & 12 & 89 & 264 & 388 & 209 & 398 & 254 & \dots & 500 & 656 & 807 & 193 \\ 552 & 252 & 149 & 181 & 510 & 88 & 5 & 522 & 777 & 670 & \dots & 338 & 130 & 617 & 3 \end{pmatrix}$$

The public key is $\mathsf{pk} = (\mathsf{G}^{pub}, t)$ and secret key is $\mathsf{sk} = (b, P_D, \sigma)$.

Randomly generates a vector

$\boldsymbol{r}$ =(691, 260, 260, 508, 152, 171, 101, 324, 323, 691, 577, 58, 423, 231, 21, 500, 235, 491, 63, 72, 57, 774, 676, 187, 687, 254, 67 , 263, 21, 761, 505, 603, 442, 43, 534, 559, 750, 667, 805, 600, 15, 736, 431, 723, 708, 736, 404, 378, 84, 32, 403, 318, 338, 259, 439, 348, 351, 463, 604, 73, 195, 398, 626, 10, 789, 602, 707, 302, 267, 301, 392, 239, 159, 195, 126, 702, 121, 71, 554, 316, 789, 536, 644, 160, 324, 386, 788, 160, 528, 693, 763, 198, 580, 469, 374, 538, 762, 795, 180, 427, 408, 511, 673, 715, 560, 165, 183, 65, 626, 572, 647, 645, 137, 672, 327, 365, 94, 752, 328, 437, 392, 291, 93, 728, 91, 513, 161, 94, 213, 237, 444, 473, 25, 202, 756, 478, 279, 377, 515, 444, 675, 681, 558, 281, 151, 532, 489, 489, 290, 331, 227, 272, 618, 410, 328, 357),

and a error vector of weight t

$\boldsymbol{e}$ =(83, 0, 237, 87, 327, 0, 534, 639, 0, 485, 0, 0, 434, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 614, 342, 0, 0, 435, 0, 126, 0, 542, 0, 547, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 185, 124, 21, 0, 0, 0, 0, 0, 65, 0, 0, 0, 0, 236, 0, 665, 0, 659, 0, 408, 465, 0, 0, 0, 0, 246, 0, 0, 0, 0, 0, 106, 782, 0, 0, 0, 563, 626, 0, 0, 0, 499, 746, 0, 0, 0, 0, 0, 0, 610, 651, 0, 307, 0, 0, 797, 0, 0, 0, 243, 0, 0, 0, 0, 0, 799, 0, 0, 0, 557, 53, 0, 314, 588,

447, 258, 0, 379, 100, 714, 201, 0, 0, 0, 0, 0, 465, 0, 0, 487, 806, 0, 0, 186, 702, 0, 0, 267, 718, 635, 691, 0, 608, 185, 492, 0, 0, 33, 0, 0, 0, 631, 0, 560, 0, 536, 0, 0, 0, 0, 0, 0, 0, 96, 661, 354, 0, 0, 0, 0, 0, 491, 0, 0, 0, 262, 0, 451, 0, 7, 0, 444, 0, 0, 0, 0, 0, 0, 0, 0, 715, 0, 0, 0, 0, 361, 0, 0, 0, 0, 22, 509, 518, 0, 421, 618, 0, 792, 0, 0, 0, 0, 0, 95, 0, 0, 0, 0, 255, 0, 0, 391, 0, 0, 0, 0, 558, 0, 0, 63, 0, 542, 0, 0, 0, 0, 0, 800, 0, 0, 287, 0, 0, 768, 0, 0, 0, 0, 0, 297, 725, 0, 0, 0, 483, 72, 271, 0, 0, 0, 0, 0, 450, 0, 0, 132, 0, 149, 6, 0, 0, 0, 729, 0, 0, 0, 0, 229, 0, 0, 0, 0, 18, 0, 346, 0, 0, 0, 0, 0, 0, 0, 331, 0, 578, 0, 0, 0, 0, 224, 0, 0, 0, 0, 0, 0, 0, 0, 728, 0, 389, 440, 425, 0, 808, 0, 0, 0, 200, 707, 0, 0, 755, 0, 0, 426, 0, 0, 223, 0, 0, 0, 284, 627, 0, 94, 0, 0, 719, 0, 606, 65, 0, 0, 0, 0, 0, 0, 0, 0, 143, 0, 0, 0, 137, 0, 662, 0, 0, 0, 229, 0, 690, 0, 0, 0).

Suppose the message is

$\boldsymbol{m}$ =(458 , 740 , 766 , 174 , 543 , 45 , 15 , 746 , 343 , 151 , 801 , 644 , 566 , 59 , 481 , 435 , 228 , 603 , 313 , 164 , 547 , 426 , 728 , 720 , 705 , 664, 22 , 703 , 26 , 424 , 360 , 367 , 373 , 564 , 605 , 61 , 109 , 490 , 436 , 368 , 706 , 497 , 773 , 73 , 398 , 70 , 295 , 553 , 255 , 433 , 173 , 439, 564 , 790 , 281 , 718 , 746 , 659 , 723 , 538 , 606 , 225 , 60 , 215 , 748 , 387 , 286 , 551 , 47 , 725 , 305 , 496 , 745 , 75 , 473 , 419 , 113 , 216, 1 , 611 , 85 , 559 , 114 , 186 , 384 , 790 , 52 , 430 , 551 , 68 , 431 , 766 , 449 , 6 , 461 , 126 , 631 , 86 , 806 , 320 , 325 , 259 , 658 , 73 , 217 , 435, 719 , 398 , 563 , 289 , 39 , 673 , 494 , 715 , 18 , 659 , 388 , 645 , 765 , 83 , 102 , 740 , 75 , 285 , 236 , 427 , 292 , 539 , 174 , 236 , 191 , 156, 371 , 701 , 568 , 4 , 164 , 576 , 337 , 416 , 67 , 450 , 560 , 497 , 281 , 118 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0).

The cipher-text is

$\boldsymbol{c} = \boldsymbol{c1} \,||\, \boldsymbol{c2} = (\boldsymbol{r}\mathsf{G} + \boldsymbol{e}) \,||\, (\boldsymbol{m} + \boldsymbol{r})$

= (127 , 340 , 695 , 400 , 601 , 406 , 608 , 21 , 765 , 536 , 109 , 568 , 791 , 329 , 690 , 520 , 708 , 555 , 578 , 462 , 751 , 317 , 532 , 471 , 141 , 478 88 , 709 , 776 , 296 , 505 , 638 , 567 , 420 , 115 , 667 , 479 , 144 , 46 , 750 , 671 , 174 , 230 , 38 , 766 , 473 , 79 , 584 , 162 , 481 , 679 , 53 , 403 , 392 ,726 , 80 , 577 , 425 , 37 , 711 , 611 , 140 , 497 , 649 , 719 , 41 , 572 , 707 , 169 , 261 , 626 , 508 , 687 , 731 , 618 , 577 , 708 , 320 , 708 , 188 , 61 , 191 , 199 , 537 , 419 , 508 , 493 , 253 , 798 , 139 , 231 , 610 , 94 , 358 , 129 , 251 , 507 , 693 , 795 , 512 , 729 , 52 , 116 , 718 , 74 , 69 , 50 , 26 , 230 , 383 , 26 , 62 , 422 , 240 , 743 , 310 , 439 , 805 , 600 , 626 , 775 , 388 , 206 , 531 , 697 , 387 , 206 , 138 , 53 , 310 , 505 , 131 , 691 , 170 , 625 , 244 , 638 , 269 , 455 , 465 , 147 , 547 , 274 , 329 , 175 , 71 , 716 , 93 , 534 , 665, 470 , 71 , 747 , 376 , 4 , 254 , 706 , 651 , 445 , 178 , 505 , 526 , 579 , 474 , 363 , 693 , 335 , 167 , 745 , 782 , 605 , 361 , 109 , 377 , 83 , 479 , 103 , 55 , 455 , 800 , 549 , 163 , 158 , 738 , 15 , 402 , 261 , 759 , 682 , 142 , 320 , 802 , 176 , 561 , 599 , 640 , 479 , 6 , 667 , 671 , 431 , 605 , 246 , 792 , 225 , 756 , 78 , 559 , 513 , 158 , 468 , 364 , 288 , 92 , 392 , 26 , 328 , 521 , 511 , 541 , 142 , 2 , 510 , 179 , 424 , 314 , 0 , 648 , 481 , 326 , 241 , 758 , 479 , 592 , 366 , 22 , 289 , 232 , 444 , 176 , 211 , 575 , 137 , 146 , 432 , 62 , 127 , 187 , 282 , 781 , 274 96 , 83 , 722 , 5 , 576 , 33 , 530 , 563 , 784 , 724 , 697 , 555 , 550 , 641 , 51 , 132 , 510 , 595 , 476 , 45 , 286 , 71 , 205 , 35 , 241 , 443 , 703 , 646 , 68 , 316 180 , 501 , 27 , 370 , 753 , 580 , 106 , 260 , 243 , 440 , 224 , 477 , 478 , 544 , 346 , 655 , 133 , 133 , 266 , 340 , 274 , 501 , 3 , 549 , 179 , 685 , 210 , 53 142 , 410 , 560 , 130 , 286 , 523 , 645 , 187 , 361 , 127 , 751 , 12 , 763 , 449 , 265 , 749 , 80 , 798 , 678 , 174 , 719 , 284 , 312, 135 , 781 , 251 , 322 267 , 196 , 640 , 387 , 124 , 390 , 509 , 784 , 582 , 611 , 355 , 473 , 389 , 705 , 466 , 701 , 15 , 95 , 389 , 410 , 522 , 263 , 762 , 405 , 779 , 92 640 , 372 522 , 590 , 341 , 616 , 808 , 348 , 233 , 179)
|| (338 , 189 , 215 , 682 , 695 , 216 , 116 , 259 , 666 , 31 , 567 , 702 , 178 , 290 , 502 , 124 , 463 , 283 , 376 , 236 , 604 , 389 , 593 , 96 , 581 , 107 , 89 , 155 , 47 , 374 , 54 , 159 , 4 , 607 , 328 , 620 , 48 , 346 , 430 , 157 , 721 , 422 , 393 , 796 , 295 , 806 , 699 , 120 , 339 , 465 , 576 , 757 , 91 , 238 , 720 , 255, 286 , 311 , 516 , 611 , 801 , 623 , 686 , 225 , 726 , 178 , 182 , 42 , 314 , 215 , 697 , 735 , 93 ,

270 , 599 , 310 , 234 , 287 , 555 , 116 , 63 , 284 , 758 , 346 , 708 , 365 , 29 , 590 , 268 , 761 , 383 , 153 , 218 , 475 , 24 , 664 , 582 , 70 , 175 , 747 , 733 , 770 , 520 , 788 , 777 , 600 , 91 , 463 , 378 , 50 , 686 , 507 , 631 , 576 , 345 , 213 , 482 , 586 , 282 , 520 , 494 , 220 , 168 , 202 , 327 , 129 , 453 , 633 , 387 , 473 , 635 , 629 , 396 , 92 , 513 , 482 , 443 , 142 , 41 , 49 , 742 , 320 , 307 , 778 , 432 , 650 , 489 , 489 , 290 , 331 , 227 , 272 , 618 , 410 , 328 , 357).

Obviously we can get that $\mathbf{Decrypt}(\mathsf{sk}, \boldsymbol{c}) = \boldsymbol{m}$.

The lower bound of information set decoding is showed in **Example 1** where

$$WF_{qISD} > 2^{128.95}.$$

To apply the structure attacks, one need to find the minimum weight codeword of the underlying code, whose computational complexity is about

$$O(\binom{n}{n-k}) > 2^{360}.$$

Meanwhile, the exhaustive search will cost

$$6q\binom{(1+q-2\sqrt{q})/6}{n/6} > 2^{140.05}.$$

Last but not least, the ECP decoding attack relies on the unique decoding algorithm. Since we add more than $t_0 = (d-1)/2$ errors to the codeword, the direct ECP-decoding algorithm will fail. The combined ECP attack will cost

$$\binom{n}{t_1}(q-1)^{t_1} > 2^{260.54}.$$

5 Efficiency Analysis

Here we show how the proposed scheme preforms in the size of key pairs, and the speed of encryption and decryption.

The size of public key can be calculated as the size of first row of each block and the size of error number. Each element of the matrix is an element of $\mathbb{F}_q$ with $\log q$ bits, and for each block with 6 rows we only need 1 row to recover it, thus we need $(n \cdot k/6) \cdot \log q + \log t$ bits to save the public key. The secret key is a triple of $(\mathcal{E}, D, \sigma)$, where $\mathcal{E}$ is a Weierstrass equation with one parameter, $D = \Sigma_{i=1}^{l} \Sigma_{P_i \in S_i} P_i$ is the divisor, and σ is an automorphism with order 6 of $\mathcal{E}$. There are $n/6$ orbits S_i, and every orbit can be recover from one rational point and the automorphism function σ. Since the $\mathcal{E}$, i.e. the relationship between the coordinates of points is known, we only need 1 coordinate to recover each point, which leads to size of D is $n \log q/6$ bits. The automorphism function is in fact a mapping between the coordinates, and with the same reason, the mapping between 1 coordinate with 1 parameters is enough to recover the whole mapping, which only needs $\log q$ bits. Thus the size of secret key is $2 \log q + n \log q/6$ bits.

Thanks to the special structure of quasi-cyclic elliptic codes, we reduce the size of both public key and secret key a lot compare to the other variants of

McEliece based cryptosystem. Although the encryption algorithm is almost as fast as the original scheme, the speed of decryption because of the using of list-decoding is sacrificed.

Here we compare the key size (in bits) of our scheme with some proposals to NIST Post-Quantum Cryptography Standardization in **Table 2**.

Table 2. Key-size(bits) for Quasi-Cyclic Elliptic Codes based PKE

1^λ	Scheme	Public-key	Secret-key	KEM message
2^{128}	QC-EC	67709	448	5720
	Big Quake	203856	118176	1608
	RQC	11928	11928	12952
2^{192}	QC-EC	150029	672	8408
	Big Quake	673056	246880	3248
	RQC	21928	21928	22952
2^{256}	QC-EC	284891	847	10508
	Big Quake	1198400	334432	3936
	RQC	28080	28080	29140

6 Conclusion

We construct a public-key encryption system based on quasi-cyclic elliptic codes, which can resist all attacks against algebraic geometry codes as far as we know. The knowledge of building quasi-cyclic codes from elliptic curves helps us to decrease the size of both public key and secret key. To resist the attacks, we add errors beyond the half of the minimum distance of the code, and then use list-decoding algorithm as a subroutine to decode the ciphertext. Our cryptosystem performs good on the storage size, but the decrypt speed is sacrificed to ensure the security. Obviously, with the development of list-decoding algorithm, especially list-decoding algorithm towards quasi-cyclic algebraic geometry codes, the decryption speed of our scheme will be accelerate.

We noticed that many geometrically curves have automorphism with large order, so it is worthwhile work to explore better schemes based on algebraic geometry codes.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61672550) and the National Key R& D Program of China(2017YFB0802503).

References

1. Abatangelo V, Larato B. Near-MDS codes arising from algebraic curves. Discrete mathematics, 2005, 301(1): 5-19.
2. Bardet M, Barelli E, Blazy O, et al. BIG QUAKE: BInary Goppa QUAsi-cyclic Key Encapsulation (2017). https://hal.archives-ouvertes.fr/hal-01671866/document
3. Beelen P , Hholdt, Tom. The Decoding of Algebraic Geometry Codes[M]. Advances In Algebraic Geometry Codes. 2008.
4. Berger T P , Cayrel P L , Gaborit P , et al. Reducing Key Length of the McEliece Cryptosystem. International Conference on Progress in Cryptology-africacrypt. D-BLP, 2009.
5. Berlekamp E R , Mceliece R J , Tilborg H C A V . On the inherent intractability of certain coding problems. IEEE Trans.inf.theory, 1978, 24(3):384-386.
6. Bernstein D J , Lange T , Peters C . Attacking and Defending the McEliece Cryptosystem. International Workshop on Post-quantum Cryptography. Springer-Verlag, 2008.
7. Chabaud F. On the security of some cryptosystems based on error-correcting codes. Workshop on the Theory and Application of of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1994: 131-139.
8. Cheng Q. Hard problems of algebraic geometry codes. IEEE Transactions on Information Theory, 2008, 54(1): 402-406.
9. Couvreur A, Mrquez-Corbella I, Pellikaan R. A polynomial time attack against algebraic geometry code based public key cryptosystems. Information Theory (ISIT), 2014 IEEE International Symposium on. IEEE, 2014: 1446-1450.
10. Couvreur A, Mrquez-Corbella I, Pellikaan R. Cryptanalysis of McEliece cryptosystem based on algebraic geometry codes and their subcodes. IEEE Transactions on Information Theory, 2017, 63(8): 5404-5418.
11. Deneuville J C , Gaborit P , Gilles Z. Ouroboros: A Simple, Secure and Efficient Key Exchange Protocol Based on Coding Theory[C]. International Workshop on Post-quantum Cryptography. Springer, Cham, 2017.
12. Diffie W, Hellman M. New directions in cryptography. IEEE transactions on Information Theory, 1976, 22(6): 644-654.
13. Esmaeili M, Yari S. Generalized quasi-cyclic codes: structural properties and code construction. Applicable Algebra in Engineering, Communication and Computing, 2009, 20(2): 159-173.
14. Faugre J C , Otmani A , Perret L , et al. Algebraic Cryptanalysis of McEliece Variants with Compact Keys. Advances in Cryptology EUROCRYPT 2010. Springer Berlin Heidelberg, 2011.
15. Faure C, Minder L. Cryptanalysis of the McEliece cryptosystem over hyperelliptic codes. In ACCT 2008, pages 99107, 2008.
16. Gaborit P. Shorter keys for code based cryptography. International Workshop on Coding and Cryptography WCC 2005. Bergen, Norway: ACM Press, 2005, pp. 8191.
17. Goppa V D . Codes associated with divisors. Probl.peredachi Inf, 1977, 13(1):3339.
18. Guruswami V, Sudan M. Improved decoding of reed-solomon and algebraic-geometry codes, IEEE Trans. Inf. Theory, vol. 45, no. 6, pp. 1757-1767, 1999.
19. Guruswami V, Sudan M. On representations of algebraic-geometric codes for list decoding. European Symposium on Algorithms. Springer, Berlin, Heidelberg, 2000: 244-255.

20. Guruswami V , Vardy A . Maximum-likelihood decoding of Reed-Solomon codes is NP-hard. IEEE Transactions on Information Theory, 2004, 51(7):2249-2256.
21. Hofheinz D, Hvelmanns K, Kiltz E. A modular analysis of the Fujisaki-Okamoto transformation. Theory of Cryptography Conference. Springer, Cham, 2017: 341-371.
22. Hoholdt T, Pellikaan R. On the decoding of algebraic-geometric codes. IEEE Transactions on Information Theory, 1995, 41(6): 1589-1614.
23. Hoholdt T, Van Lint J H, Pellikaan R. Algebraic geometry codes. Handbook of coding theory, 1998, 1(Part 1): 871-961.
24. Janwa H, Moreno O. McEliece public key cryptosystems using algebraic-geometric codes. Designs, Codes and Cryptography, 1996, 8(3): 293-307.
25. Katz J, Lindell Y. Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series). Chapman & Hall/CRC, 2007.
26. Li J, Wan D, Zhang J. On the minimum distance of elliptic curve codes. arXiv preprint arXiv:1501.01138, 2015.
27. Márquez-Corbella I, Martínez-Moro E, Pellikaan R, et al. Computational aspects of retrieving a representation of an algebraic geometry code. Journal of Symbolic Computation, 2014, 64: 67-87.
28. Márquez-Corbella I, Martínez-Moro E, Pellikaan R. On the unique representation of very strong algebraic geometry codes. Designs, Codes and Cryptography, 2014, 70(1-2): 215-230.
29. Mceliece R J. A public-key cryptosystem based on algebraic. Coding Thv, 1978, 4244: 114-116.
30. McEliece R J. The Guruswami–Sudan Decoding Algorithm for Reed–Solomon Codes. Interplanetary Network Progress Report, 2003, 153.
31. Minder L. Cryptography based on Error Correcting Codes. PhD thesis, EPFL, Lausanne, 2007.
32. Niebuhr R. Attacking and Defending Code-based Cryptosystems . Technische Universitt, 2012.
33. Niebuhr R, Cayrel P L, Bulygin S, et al. On lower bounds for information set decoding over Fq. Proceedings of the 2nd International Conference on Symbolic Computation and Cryptography, SCC. 2010, 10: 143-157.
34. Niederreiter H. Knapsack-type cryptosystems and algebraic coding theory. Prob. Control and Inf. Theory, 1986, 15(2): 159-166.
35. Otmani A , Tillich J P , Dallot L. Cryptanalysis of Two McEliece Cryptosystems Based on Quasi-Cyclic Codes. Mathematics in Computer Science, 2010, 3(2):129-140.
36. Overbeck R, Sendrier N. Code-based cryptography. Post-quantum cryptography. Springer, Berlin, Heidelberg, 2009: 95-145.
37. Pellikaan R. On decoding by error location and dependent sets of error positions. Discrete Mathematics, 1992, 106: 369-381.
38. Pellikaan R. On the efficient decoding of algebraic-geometric codes. Eurocode92. Springer, Vienna, 1993: 231-253.
39. Pellikaan R. On the existence of error-correcting pairs. Journal of Statistical Planning and Inference, 1996, 51(2): 229-242.
40. Peters C. Information-set decoding for linear codes over Fq. International Workshop on Post-Quantum Cryptography. Springer, Berlin, Heidelberg, 2010: 81-94.
41. Prange E . The use of information sets in decoding cyclic codes. Ire Transactions on Information Theory, 1962, 8(5):5-9.
42. Sendrier N. Code-based cryptography: State of the art & perspectives. IEEE Security and Privacy, 2017, 15(4): 44-50.

43. Shor P W. Algorithms for quantum computation: discrete logarithms and factoring. IEEE Symposium on Foundations of Computer Science. 1994:124-134.
44. Sidelnikov V M, Shestakov S O. On insecurity of cryptosystems based on generalized Reed-Solomon codes. Discrete Mathematics and Applications, 1992, 2(4): 439-444.
45. Silverman J H . The Arithmetic of Elliptic Curves. Inventiones Mathematicae, 1974, 23(3-4):179-206.
46. Tong H, Ding Y. Quasi-cyclic NMDS codes. Finite Fields and Their Applications, 2013, 24: 45-54.
47. Vardy A . The intractability of computing the minimum distance of a code. IEEE Press, 1997.
48. Zhang F, Liu S. Solving ECDLP via List Decoding. IACR Cryptology ePrint Archive, Report2018/795, 2018. https://eprint.iacr.org/2018/795.pdf