UM-SJTU Joint Institute
Ve475 Introduction to Cryptography

Project 1
Error Correcting Codes and Cyptography

Zhu Chen	ID:5143709145
Shi Yucheng	ID:5143709089
Xu Rishang	ID:5143709097
Zeng Zhi	ID:5143709108

Date: July 4, 2017

Abstract

Error correcting code is wideley used in physical layer of the internet protocal stack. Its objective is to ensure the bit streams that we transfer through thousands-of-mile channels to be correct. The coding scheme and the hardness of decoding in the error correcting code make it applicable to public key cryptography, which is usually in the application layer in the internet protocol layer. In this paper we are going to introduce the mechanism of error correcting code and coding theory, then show the relationship between the error correcting code and public key cryptography, and finally introduce two kinds of Cryptosystems based on coding theory.

Contents

1 Introduction

1.1 Shannon Capacity Theorem and Coding

People have never stopped trying to make the communication more and more efficient. They want to shorten the time spent on conmunicating so as to improve the productivity. This concern became extremely common at the beginning of Information Era. The popularity of Internet causes the fact that there are nearly 2^{216} bits transferred per microsecond nowadays. Moreover, with the longer distance of transmission, people start to study how to make the information reach each other correctly.

The first approach, which is a huge progress, is to use the digital signal {0,1} to represent signals. The long-distance-analoge data trasmitting will face a problem of distortion and attenuation. However, the digital signals can be transferred using modulation and signal amplifier so as to produce less errors.

Although this helped a lot, there were still some unavoidable errors because we need to transfer data through huge routers to more than 200000KM away, there have to be some errors caused by confliction and harrasing. People have tried to increase the power so as to lessen the error, but it didn't work very well. In 1948 Shannon [1] presented a capacity theory that was again a huge breakthrough. He mentioned that we needn't correct the error by inproving the power, we can modify the bits we are tranferring (coding). As long as the tranmit rate didn't exceed certain threshold, we can theoretically achieve the reliable communication. This threshold is given:

$$C = W_c \log_2(1 + \mathrm{SNR})$$

where the C is named the Shannon Capacity in bits/sec, W_c is the bandwidth of the channel (how wide the range of frequency can the channel stand), and SNR is the source-noise rate, which determines the potential noise in the channel.

Shannon provided us with a threshold, but he didn't mention how to modify the bits we are transferring, this field of study later became Information Theory and Coding Theory is concerned with bit stuffing and modifying so as to minimize the errors.

1.2 Error Detecting Code (EDC) and Error Correcting Code (ECC)

Error Detecting code (EDC) and Error Correcting code (ECC) are two of the major types of coding. The objective is to detect, and correct code, respectively. Before we focus on the ECC, we will first introduce EDC.

1.2.1 EDC and Single Parity Check

In EDC, we just need to detect t errors in the code and the further handling of the code will move on to the higher protocol layer. Single Parity Check is a scheme to detect single error in on code blocks.

Assume we split the bit stream into 7 bits per block $b_1, b_2, ..., b_7$ and add one check bits b_8 into the block. 8 bits form a block, and the b_8 satisfies:

$$b_8 = \sum_{i=1}^{7} b_i$$

Note that the bit can only be 0 or 1. So

$$\sum_{i=1}^{8} b_i = 2\sum_{i=1}^{7} b_i = 0$$

Suppose there is an error in the block, this will cause the sum to be 1 instead of 0, thus letting us detect one error successfully.

However, if there are even number of errors, the sum will still be zero, and we failed to detect the error.

1.2.2 ECC and Hamming Code

Error correcting code can ensure us to correct t errors in one code block.

Hamming (n, k) code [1] is one of the most famous code scheme. It uses m check bits per block to detect and correct single error per block. k is the number of information bits per block, and $k = 2^m - 1 - m$. n is the number of bits per block and $n = 2^m - 1$.

We will take Hamming $(7, 4)$ code as an example. First we will generate an $m \times n$ matrix where each columns are orthogonal to each other and all the

nonzero mutually orthogonal columns except 0 columns appears only once in the matrix. In Hamming (7,4) code we can get:

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Here we put the columns will one 1 to the right and form a Indentity Matrix at the right. Then we remove it and right multiply the remaining matrix with the column of information bit. Then we can get the check bits. For example:

$$\begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

The relationship between the m check bits and b check bits are

$$\begin{cases} m_1 = b_1 + b_2 + b_3 \\ m_2 = b_1 + b_2 + \quad b_4 \\ m_3 = b_1 + \quad b_3 + b_4 \end{cases}$$

The relationship between the m check bits and b check bits are

Then we can get the codeword $\mathbf{c} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ m_1 \\ m_2 \\ m_3 \end{pmatrix}$

Since

$$\begin{cases} m_1 + m_1 = b_1 + b_2 + b_3 + \quad m_1 \quad = 0 \\ m_2 + m_2 = b_1 + b_2 + \quad b_4 + \quad m_2 \quad = 0 \\ m_3 + m_3 = b_1 + \quad b_3 + b_4 + \quad m_3 = 0 \end{cases}$$

We can get

$$\mathbf{Hc} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ m_1 \\ m_2 \\ m_3 \end{pmatrix} = 0$$

Suppose there is one error in **c**. Denote it by matrix

$$\mathbf{e} = \mathbf{c'} - \mathbf{c},$$

where c' is the error code block and c is the original block. We can see that in **e** the corresponding error bit is 1 and others are 0. For example, if b_1 is the error bits. Then

$$\mathbf{Hc'} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} [\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ m_1 \\ m_2 \\ m_3 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

From the checking matrix **H** we can get the column $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ corresponds to the b_1, thus helping us find the error bit because each columns are different.

1.3 Cryptography Based on Error Correcting Codes

1.3.1 Public Key Cryptography

The Error correcting code can be used in Public Key Cryptography. Although as a student in Ve475 we are familiar with the Public Key Cryptography, we should also briefly mention it for wide range of readers.

The traditional cryptosystem is commonly *symmetric cyptosystems*. That is, we use the same key to encrypt and decrypt the message. But the problem is that we have some trouble when we are giving key to the other. If malicious attacker get the key through the channel, then all the message will be known.

The public key cryptography is widely used now and is *assymmetric cryptosystem*. This cryptosystem has the following elements:

1. **Private Key and Public key**: The key generator will generate this two keys K_{pub} and K_{priv}. The private key is kept only by the owner, and the public key is tranferred and known by everybody.
2. **Encryption**: Everyone can encrypt message using the public key. We can get the ciphertext c by a function:
$$c = \text{enc}(K_{\text{pub}}, m)$$
3. **Decryption**: The decryption is performed by owners with the private key. The plaintext m is given by:
$$m = \text{dec}(K_{\text{priv}}, \text{enc}(K_{\text{pub}}, m))$$

This cryptosystem is very secure because the attack can never get the private key. Since the decoding procedure and the encoding procedure is using different key, we can regard the procedure as asymmetric.

There are three types of public cryptosystems invented till now [3] : number theory based cryptography, lattice based cryptography, and correcting codes based cryptography. In the above examples in 1.2 we can see that the coding and decoding procedure is like the public key cryptography and we will mainly focus on this kind of public cryptosystems

1.3.2 History of ECC Based Cryptology and Current Situations

Using the error correcting code into the public key cryptography is an old approach. The first cryptosystem is presented by McEliece in 1978.[**?**](we

will discuss it in details later). It is very efficient and has not been broken yet because of its large public keys.

After the first cryptosystem established, three kinds of researches were done to improve the cryptosystem. One is to improve the decoding scheme and choose the best parameter to resist the attacks. Another one to make it more powerful cryptosystem to make it faster to decrypt without being unsafe. The third one is to research on the structure of the codes in order to devise some attacks on these systems. The three kinds of research are related to each other.

In 1994, SideInikov [2] proposed a McEliece type cryptosystem that uses Reed-Muller codes (we will mention later) as the family of codes.

Although there are several powerful designs of attack and improvement of the McEliece system, the security of this system is not well understood nowadays. It is to be further studied that what properties should a code have so that it is structurally safe to be used.

1.3.3 Our Focuses

In this paper we will introduce McEliece Type Cryptosystem and one of the variants - The Sidelnikov Cryptosystem. We will first introduce some background knowledge behind these two cryptosystems, then introduce their encrypt and decrypt methed, and finally introduce some attacks against them.

2 Linear Code and its Application on Cryptography

2.1 Generalization of Hamming Code –Linear Code

2.1.1 Basic Definitions

Definitions: Let $\mathbb{F}$ be a field called *alphabet*. An $[n, k]$ block code $\mathcal{C}$ is a subfield defined over an *alphabet* $\mathbb{F}_q^n$ with q symbols. It is a set of q^k vectors(*codewords*) with length n, where n is the **length** of the code and k is the **dimension** of the code. *Binary linear code* is code defined on $\mathbb{F}_2$. [10]

Definition: In general, the *weight* of a code is its distance to the origin **0**.[10] For simplicity, in section 2.1, we use *code* to denote binary linear code. Therefore, the weight of code is the number of ones it contains. For example, the weight of [1,0,1,0,1,0,1] is 4.

Definition: The *minimum weight* d of a code is the smallest weight for any nonzero codeword in the code. A [n,k] code with minimum weight d is denoted as [n,k,d].[10]

Definition: Since a linear block code $\mathcal{C}$ is a k-dimentional vector space, there exist k linearly independent vectors $g_0, g_1, \cdots, g_{k-1}$ such that every codeword c in $\mathcal{C}$ can be represented as a linear combination of these vector,

$$c = m_0 g_0 + m_1 g_1 + \cdots + m_{k-1} g_{k-1}$$

. We can derive a $k \times n$ matrix G,

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix}$$

$$\mathbf{m} = [m_0, m_1, \cdots, m_{k-1}]$$

$$codeword = \mathbf{m}G$$

. [10]

2.1.2 Problem Definitions

1. ***Encoding Problem*** Constructing a map $\mathbb{F} \to \mathcal{C}$, where $\mathcal{C}$ is a [n,k] code. This is easy, through generator matrix G, by encoding: $x \to xG$.
2. ***Decoding Problem*** This is notoriously hard in general. For $x \in \mathbb{F}^{\times}$, any vector $y \in \mathcal{C}$ such that

   $$d(x, y) = \min_{z \in \mathcal{C}} d(x, z)$$

   . y is called a maximum likelihood decoding of x.

2.1.3 Parity Check Matrix and Syndrome decoding

Syndrome decoding is method to decode words in $\mathbb{F}$. It is based on parity check matrix.

Definition: The dual space to an [n, k] code $\mathcal{C}$ is the [n, n-k] dual code of $\mathcal{C}$, denoted as $\mathcal{C}^{\perp}$. As a vector space, $\mathcal{C}^{\perp}$ has a basis which denoted as H, the parity check matrix of code $\mathcal{C}$. We have

$$GH^T = \mathbf{0}$$

.

Theorem: A vector v is a codeword if and only if

$$vH^T = \mathbf{0}$$

.

Error Detection and Correction: Let x be the original message and $c = xG$. Assume the receiver got message

$$\mathbf{r} = c + \mathbf{e}$$

, where $\mathbf{e}$ is the error vector. Let H be the parity check matrix. The syndrome is given by

$$s = \mathbf{r}H^T = \mathbf{c} + \mathbf{e}H^T = \mathbf{e}H^T$$

. If the code is well constructed(Hamming code [7,4,3]), there exists a bijection between s and $\mathbf{e}$. Therefore, by computing syndrome s, we can calculate the error $\mathbf{e}$.

Definition: A code is called a t-error-correcting code if there exists a fast algorithm that can correct t errors.

For example, hamming code is one-error-correcting code, because it can correct 1 error by syndrome decoding. We replace previous symbol of [n,k,d] code to [n,k,t] code.

2.2 Public Crytosystem based on Linear Code

Let G be the generator matrix of an [n,k,t] code $\mathcal{C}$. Select a $k \times k$ invertible matrix S and a $n \times n$ permutation matrix P.

$$G' = SGP$$

.

1. Bob use G' as public key.
2. Alice want to send message m to Bob. She send $m' = mG' + \mathbf{e}$ to Bob, where $\mathbf{e}$ is n-vector with weight t.
3. Bob computes

$$s = m'P^{-1} = mSGPP^{-1} + eP{-}1 = m'SG + eP^{-1}$$

. Need to notice that $m'SG+eP^{-1}$ is a t-errors code in $\mathcal{C}$, because mSG is a codeword in $\mathcal{C}$ and eP^{-1} has weight t. Since the t-error decoding method exists for code $\mathcal{C}$, Bob can decode it to get mS.

4. Bob knows S and the inversion of S, he can easily computes m.

Example: Here is an example of a public Cryptosystem based on [7,4,1(t)], where

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$S = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Bob makes the public generator matrix

$$G' = SGP = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

If Alice want to send the message $m = [1, 1, 0, 1]$ to Bob, she first construct a weight 1 error vector, say $\mathbf{e} = [0, 0, 0, 0, 1, 0, 0]$ and computes

$$c = mG' + \mathbf{e} = [0, 1, 1, 0, 1, 1, 0]$$

After receiving c, Bob fist compute $c' = cP^{-1} = mSG + eP^{-1}$, where

$$P^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$c' = [1, 0, 0, 0, 1, 1, 1]$. The parity check matrix is

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

The syndrome of c' is $[0, 0, 1]$, which means a error in the last digit. So $mSG = [1, 0, 0, 0, 1, 1, 0]$, which is the same as the first vector in G. Therefore, $mS = [1, 0, 0, 0]^T$, Since S is invertible,

$$S^{-1} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

. So $m = [1\ 0\ 0\ 0]^T \cdot S^{-1} = [1\ 1\ 0\ 1]$

2.2.1 Cryptoanalysis on the Previous Cryptosystem

Attacks based on finding S, G, P is not realistic, since their number have the same growth rate as factorization. Attack based on correcting errors seems more appealing. There are limited error combination, for [7,4,1(t)], there are only 7 possible error vector **e'**. Try decoding $c = mG' + e - e'$ with the 7 possible **e'** will recover the plaintext. However, for piratical cryptosystem like McEliece cryptosystem, they use Goppa code[1024,524,50], the possible error combination is $\binom{1024}{50}$, this strategy will not work. Possible attack to McEliece cryptosystem such as *information set decoding* based on *low weight word finding algorithm* will be mentioned later.

3 McEliece Cryptosystem

3.1 Algebraic Codes

In this section, we first restate some preliminaries in algebraic geometry textbook[8] and then discuss two families of algebraic codes: Reed-Muller codes and algebraic geometry codes.

3.1.1 Reed-Muller Codes

Let $\mathbb{F}_2$ be the Galois field of two elements, we define a *boolean function* to be a member of the ring

$$\mathcal{B}(\{v_1, \ldots, v_m\}) := \mathbb{F}_2[v_1, \ldots, v_m]/(v_1^2 - v_1, \ldots, v_m^2 - v_m).$$

The degree of a boolean function $f \in \mathcal{B}(\{v_1, \ldots, v_m\})$ is the degree of its lowest-degree representative in $\mathbb{F}_2[v_1, \ldots, v_m]$, denoted by $\deg(f)$. We also define $\mathcal{B}(r, \{v_1, \ldots, v_m\}) := \{f \in \mathcal{B}(\{v_1, \ldots, v_m\}), \deg(f) \leq r\}$ to be the $\mathbb{F}_2$-vector space of all boolean functions of degree less than or equal to r. By counting all the possible monomials, it follows immediately the dimension of this $\mathbb{F}_2$-vector space is

$$\dim(\mathcal{B}(r, \{v_1, \ldots, v_m\})) := \sum_{i=0}^{r} \binom{m}{i}.$$

Note that for each $f \in \mathcal{B}(\{v_1, \ldots, v_m\})$, f takes values on 2^m positions. We can then associate a binary word of length 2^m obtained by evaluating f on the 2^m possible positions. We denote $\mathcal{R}(r, m)$ to be the set obtained by evaluating all the boolean functions in $\mathcal{B}(r, \{v_1, v_2, \ldots, v_m\})$.

We see that $\mathcal{R}(r, m)$ is actually a linear code because it's linear in each entry, which follows from the fact that $f \in \mathcal{B}(r, \{v_1, \ldots, v_m\})$ is linear, as well as the fact that the evaluation map, let say

$$E_v : \mathcal{B}(r, \{v_1, \ldots, v_m\}) \to \mathbb{F}_2^{2^m},$$

is linear. We also claim that the evaluation map E_v has a trivial kernal. The proof is basically induction on r[7]. Therefore, there exists a bijection between

$$\phi : \mathcal{B}(r, \{v_1, \ldots, v_m\}) \to \mathcal{R}(r, m).$$

The code r, m then has parameters

$$n = 2^m, \quad k = \dim(\mathcal{B}(r, \{v_1, \ldots, v_m\})) = \sum_{i=0}^{r} \binom{m}{i}.$$

Furthermore, we can show by induction that the minimun distance of $\mathcal{R}(r, m)$ is

$$d = 2^{m-r}[7].$$

From the definition of $\mathcal{B}(r, \{v_1, \ldots, v_m\})$ and the since we identify $\mathcal{B}(r, \{v_1, \ldots, v_m\})$ with $\mathcal{R}(r, m)$, we also have

$$\mathcal{R}(0, m) \subset \mathcal{R}(1, m) \subset \ldots \subset \mathcal{R}(r, m).$$

We also observe that the dual code of $\mathcal{R}(r, m)$ is $\mathcal{R}(m - r - 1, m)[7]$, namely,

$$\mathcal{R}(r, m)^{\perp} = \mathcal{R}(m - r - 1, m)$$

3.1.2 Algebraic Geometry Codes

To understandthe definition of algebraic geometry codes, we need first some preliminaries.

Let $\mathbb{F}$ be a field, $\overline{\mathbb{F}}$ be its algebraic closure, we denote the *affine sapce* $\mathbb{A}^n(\overline{\mathbb{F}})$. Note that the affine space is actually $\overline{\mathbb{F}}^n$. Let $n = 2$, an *plane affine algebraic curve* over $\mathbb{F}$ is a set of points of $\overline{\mathbb{F}}^2$ whose coordinates are zeros of some bivariate polynomials with coefficients in $\mathbb{F}$.

The *projective space* $\mathbb{P}^n(\overline{\mathbb{F}})$ is the set

$$(\overline{\mathbb{F}}^{n+1} \backslash \{0\}) / \sim,$$

where $\sim$ is the equivalence relation defined by $(x_0, \ldots, x_n) \sim (y_0, \ldots, y_n)$ iff there exists some $\lambda \neq 0, \lambda \in \overline{\mathbb{F}}$ such that $(x_0, \ldots, x_n) = (\lambda y_0, \ldots, \lambda y_n)$. The equivalence class is often denoted as $(X_0 : X_1 : \ldots : X_n)$.

A *projective algebraic curve* in $\mathbb{P}^2(\overline{\mathbb{F}})$ is the zero set in $\mathbb{P}^2(\overline{\mathbb{F}})$ in some homogeneuous polynomial $f \in \overline{\mathbb{F}}[X, Y, Z]$.

For some irreducible projective curve $\chi \in \mathbb{P}^2(\overline{\mathbb{F}})$ given by $f(X, Y, Z) = 0$, we can also denote the fraction space of χ, namely,

$$\overline{\mathbb{F}}(\chi) = \big(\big\{\frac{f}{g}, \mid \deg(f) = \deg(g), f, g \text{ are homo}\big\} \backslash \sim\big) \cup \{0\},$$

where $(f,g) \sim (f',g') \Leftrightarrow fg' = gf'$. We refer to the definition of *order* or *multiplicity* at a point P in literature and the usual rules that

$$\begin{aligned} \mathrm{ord}_P(f \cdot g) &= \mathrm{ord}_P(f) + \mathrm{ord}_P(g) \\ \mathrm{ord}_P(f+g) &\geq \min\{\mathrm{ord}_P(f), \mathrm{ord}_P(g)\} \end{aligned}$$

Let χ be a smooth projective curve. A *divisor* on χ is a finite formal sum of points on χ. If $n_P = 0$ for almost every $P \in \chi$, $\sum_{P\in\chi} n_P P$ is a divisor. The set of divisors of χ actually is a group and denoted by $\mathrm{Div}(\chi)$.

The degree of the divisor $\Delta = \sum_{p\in\chi} n_P P$ is defined to be

$$\deg(\Delta) := \sum_{P\in\chi} n_P.$$

The *support* of the divisor Δ, is the set of points on which it is nonzero[8], *i.e.,* $\mathrm{supp}(\Delta) := \{P \in \chi \mid n_P \neq 0\}$. For f be a nonzero rational function. $\mathrm{ord}_P(f)$ is nonzero only for a finite number of points (zeros and infinity), so we can define a divisor $\mathrm{div}(f) := \sum_{P\in\chi} \mathrm{ord}_P P$. An divisor which is the divisor of some rational function is said to be *principle*. Thus $\deg(\mathrm{div}(f)) := \sum_{P\in\chi} \mathrm{ord}_P = 0$, namely all the principle divisors has degree 0.

We say two divisors Δ_1, Δ_2 are *linearly equivalent*, if $\Delta_1 - \Delta_2$ is principle. (note $\Delta_1 - \Delta_2$ is also a divisor.) Using this equivalence class, we define the *divisor class group*

$$\mathrm{Cl}\,\chi := \mathrm{Div}\,\chi \backslash \sim .$$

In particular, if we restrict to divisors of degree 0, the *Jocobian*, namely we get

$$\mathrm{Cl}^0(\chi) := \{\Delta \in \mathrm{Div}\,\chi \mid \deg(\Delta) = 0\}.$$

We mention that the dimension g of $\mathrm{Cl}^0(\chi)$ is the *genus* of the curve χ.

Define a partial ordering $\succeq$ by

$$\Delta_1 \succeq \Delta_2 \Leftrightarrow n_P \geq m_P \forall P \in \chi,$$

where $\Delta_1 = \sum_{P\in\chi} n_P P$, $\Delta_2 = \sum_{P\in\chi} m_P P$ are two divisors.

The *linear space* of a divisor Δ is

$$\mathcal{L}(\Delta) := \{f \in \overline{\mathbb{F}}(\chi) \mid \mathrm{div}(f) + \Delta \succeq 0\} \cup \{0\}.$$

Note that the linear space is a $\overline{\mathbb{F}}-$ vector space.

We then proceed to construct algebraic geometry code (AG code). Let χ be a smooth irreducible curve defined over F. For $P_1, \ldots, P_n$ distinct rational points. The evaluation map at $P_1, \ldots, P_n$,

$$\begin{aligned} E_{P_1,\ldots,P_n} : \overline{\mathbb{F}}(\chi) &\to \overline{\mathbb{F}}^n \\ f &\mapsto (f(P_1), \ldots, f(P_n)) \end{aligned}$$

is $\overline{\mathbb{F}}-$linear. Let Δ be a divisor defined over $\mathbb{F}$ such that

$$\mathrm{supp}(\Delta) \cup \{P_1, \ldots, P_n\} = \emptyset.$$

The algebraic geometry code of divisor Δ and $P_1, \ldots, P_n$ over χ is defined as

$$\mathrm{AGC}(\chi, (P_1, \ldots, P_n)) = E_{P_1,\ldots,P_n}(\mathcal{L}(\Delta) \cup \mathbb{F}(\chi)).$$

The linearity of AG code is much the same thing as in Reed-Muller code.

By the *Riemann-Roch* theorem, $\dim(\mathcal{L}(\Delta)) = \deg(\Delta) + 1 - g$, since Δ is defined over $\mathbb{F}$, $\dim(\mathcal{L}(\Delta)) \cap \mathbb{F}(\chi) = \dim(\mathcal{L}(\Delta)) = \deg(\Delta) + 1 - g$, we remain to show $E_{P_1,\ldots,P_n}$ is injective. Let $f \in \mathcal{L}(\Delta)$ such that $f(P_1) = \cdots = f(P_{\deg(\Delta)+1}) = 0$, then $f \in \mathcal{L}(\Delta - P_1 - \cdots - P_{\deg(\Delta)+1})$. The degree is $\deg \Delta - (\deg(\Delta) + 1) < 0$, so $f = 0$. Thus any $f \in \mathcal{L}$ creates a word of weight at least $n - \deg(\Delta)$, the kernel of f is trivial, we are done.

To summarize, the $\mathrm{AGC}(\chi, \Delta, (P_1, \ldots, P_n))$ has dimension $k = \deg(\Delta) + 1 - g$ and $d \geq n - \deg(\Delta)$.

It's worth noticing that *Reed−Solomon* code, let's say $\mathrm{GRS}(k, (\alpha_1, \ldots, \alpha_n))$, is an algebraic geometry codes with χ be the projective line $Y = 0$ in $\mathbb{P}^2(\overline{\mathbb{F}})$, with $|\mathbb{F}| + 1$ rational points defined by $P_i = (\alpha_i : 0 : 1)$.

Also, the Elliptic codes is another AG code example with the underlying algebraic curve being an elliptic curve (genus $g = 1$).

3.2 The McEliece Cryptosystem

3.2.1 Basic Theorem

McEliece cryptosystem is an asymmetric encryption algorithm put forward by McEliece in 1978. Here we have some concept about it. Let C be an $[n, k]$ Goppa code over $\mathbb{F} = GF(2)$. Equation $2\delta(n) = \frac{2(1-R)}{\lceil \log_2 n \rceil}$ asserts that these codes are t-error-correcting. In some texts, a code is said to be t-error-correcting whenever its minimum distance d satisfies $d \geq 2t + 1$.

Let G be the $k \times n$ generator matrix of this Goppa code. Let S be a $k \times k$ random invertible matrix with coefficients in $\mathbb{F}$. The matrix AG is a generator matrix for the same code. Let σ be a permutation over $\{1, 2, \cdots, n\}$, and let P be the corresponding $n \times n$ permutation matrix. The matrix $G_{\text{pub}} := SGP$ is a generator matrix of C^{σ}

3.2.2 Key Generation

1. All users in a McEliece deployment share a set of common security parameters: n, k, t.
2. Select a binary (n, k)-linear code C capable of correcting t errors. This code must possess an efficient decoding algorithm and generates a $k \times n$ generator matrix G for the code C.
3. Select a random $k \times k$ binary non-singular matrix S.
4. Select a random $n \times n$ permutation matrix P.
5. Compute the $k \times n$ matrix $G_{\text{pub}} = SGP$. Therefore, public key is (G_{pub}, t); her private key is (S, G, P).

3.2.3 Encryption

1. Encode the message m as a binary string of length k.
2. Computes the vector $y = mG_{pub}$.
3. Generates a random n-bit vector z containing exactly t ones (a vector of length n and weight t.
4. Computes the ciphertext as $c = y + z$.

3.2.4 Decryption

1. Compute the inverse of P (i.e. P^{-1}).
2. Computes $\hat{c} = cP^{-1}$.
3. Use the decoding algorithm for the code C to decode $\hat{c}$ to $\hat{m}$.
4. Compute $m = \hat{m}S^{-1}$.

3.2.5 Proof of Correctness of Decryption

Now we are to verify that the we can use the decryption scheme mentioned before to correctly recover the plaintext.

$\hat{c} = cP^{-1} = m\hat{G}P^{-1} + zP^{-1} = mSG + zP^{-1}$ The Goppa code can correct up to t errors, and we can see that zP^{-1}, which is another permutation of G, we can use the decoding scheme to correctly find $\hat{m} = mS$. Then we can easily get m by $\hat{m}S^{-1} = m$.

3.3 Hardness of Assumptions on McEliece Cryptosystem

There are two major assumptions on McEliece Cryptosystem[5]

3.3.1 Hardness of General Linear decoding

First, the algorithm is based on the hardness of decoding a general linear code (which is known to be NP-hard). if a t-error-correcting $[n, k]$-code is used, it should be hard to correct t errors in a general $[n, k]$ linear code given by some generator matrix G_{pub}. For a description of the private key, an error-correcting code is selected for which an efficient decoding algorithm is known, and which is able to correct t errors. However, for McEliece cryptosystem, the public key is derived from the private key by disguising the selected code as a general linear code. For this, the code's generator matrix G is related by two randomly selected invertible matrices S and P.
On the other hand, Dumer, Micciancio and Sudan have shown that for the bounded decoding scenario, decoding is hard when the maximum weight is $> 1/2d^{1+\varepsilon}$, where $d = d(n)$ is the minimum distance of the family.

3.3.2 The Code Reconstruction Problem

The second hardness assumption of McEliece-type cryptosystems is that, given a generator matrix G, it is hard to reconstruct the underlying code. We define that an attack that exploits the structure of the underlying code a structural attack. The syndrome decoding problem was proved to be equivalent to the Learning Parity with Noise (LPN) problem and also NP-complete. To guarantee the security, the best currently known attack is still ineffective.

For example, a low rate Goppa code should be used in the McEliece cryptosystem in order to avoid attack..

3.4 An Attack and Defenses on McEliece Cryptosystem

3.4.1 Stern's Attack and Transformation into Low-weight Word Finding Problem

In order to attack against the McEliece crpytosystem, we can either use exhausting search to find the S,G,P matrix and decode it, or using information set decoding to find out the plaintext without knowing the generating matrix G. The former method is like brute-force searching and is very time consuming. So we will focus on the information set decoding, which is first presented by McEliece himself. Nowadays there are many variants of McEliece's attack and here we introduce Stern's variant[9] because many new attacks against this cryptosystem is based on this variant.

Stern transformed the attack into a problem of finding a low-weight codeword. We will first show that solving the low-weight codeword is the same as solving the McEliece attack.

Our objective is to find y in the step 2 in the encryption procedure, and then use $m = G_{pub}^{-1}$ to solve plaintext m. As mentioned in section 2.1. The cipertext c has hamming distance t with y. We can regard G as a generater, and the generated codewords forms a vector space $\mathcal{C}$. Obviously $y \in \mathcal{C}$. and the all the element $x \in \mathcal{C}$ has minimum hamming distance $2t + 1$.

The crucial step is to add the row vector, the ciphertext c at the bottom of G_{pub}, forming a new generating matrix. The resulting vector space will be $\mathcal{C} \bigcup \{\mathcal{C} - c\}$, where $\mathcal{C} - c$ is formed by all the element $\mathcal{C}$ minus the row vector c. Since all the element in $\mathcal{C}$ has hamming weight larger than $2t + 1$, the only element with hamming distance t (The low weight word)in $\mathcal{C} \bigcup \{\mathcal{C} - c\}$ is $y - c = z$, thus we can find $y = c - z = c + z$

3.4.2 Low-weight Word Finding

Stern [9] showed how to determine the low-weight word in $\mathcal{C} \bigcup \{\mathcal{C} - c\}$.

1. He first found the $(n - k) * n$ parity check matrix H, this is very easy and is shown in the example of Hamming code in section 1 and 2.1.

2. Then he randomly chose $(n-k)$ columns of H and randomly selected l columns and formed a vector Z. l is a optimizing parameter. Then he partitioned the remaining k vectors into the X and Y

3. Stern then find X, Y, Z such that there are exactly p nonzero bits in X, p nonzero bits in Y and p nonzero bits in Z. And exactly $w-2p$ nonzero bits in the remaining columns. p is a optimizing parameter, and w is the low weight we need to find.

4. After that, Stern performed transformation to H such that the $(n-k)$ columns are identity matrix. (We don't consider the invertible matrix here).

5. Now that this $(n-k) \times (n-k)$ submatrix of H is the identity matrix, each of the selected $n-k$ columns corresponds to a unique row, (the row where that column has a 1 in the submatrix). We can see that the set Z of l columns correspond to the set of l rows. For every subset A of size p of X, he got the sum of the columns in A for each of those l rows, obtaining an $l-bit$ vector $\pi(A)$. In the same way we obtain $\pi(B)$

6. For each collision such that $\pi(A) = \pi(B)$, he computed the sum of all the $2p$ columns in $A \bigcup B$. If the weight of this vector is $w-2p$, those $w-2p$ columns combined with A and B, form the code word with weight w.

3.4.3 Defense on McEliece Cryptosystem

Here we list three mean ways to defending the attack to McEliece Cryptosystem.[4]

1. The most obvious way to defend McEliece's cryptosystem is to increase n, the length of the code used in the cryptosystem. We comment that allowing values of n between powers of 2 allows considerably better optimization of the McEliece public-key size. Besides from a mild growth in decoding time, there is no interference to the key generator using a Goppa code defined through a field $\mathbb{F}_{2^d}$ of size much larger than n.

2. It is a efficient way to use list decoding to increase the number of errors t. The very recent paper has introduced a list-decoding algorithm for classical irreducible binary Goppa codes, exactly the codes used in McEliece's cryptosystem. This algorithm allows the receiver to efficiently decode approximately $n-1-\sqrt{n(n-2t-2)} \geq t$ errors. However, if our sender, introducing correspondingly more errors, the

attacker is then faced with a more difficult problem of decoding the additional errors.

3. Analysis and optimization of parameters is significant. We now propose concrete parameters $[n, k]$ for various security levels in CCA-secure variants of the McEliece cryptosystem. Recall that public keys in these variants are systematic generator matrices occupying $k(n-k)$ bits.

4 Sidelnikov's Cryptosystem

4.1 Introduction of Sidelnikov's Cryptosystem[2]

Consider a binary Reed-Muller code of order r and length N $= 2^m$ (RM_r code) is constituted by vectors of the form $\Omega_f = (f(\alpha_1) \cdots f(\alpha_n))$, where $f(x)$ is a Boolean function of m variables with non-linearity order at most r, and $\{\alpha_1 \cdots \alpha_n\} = (\mathbb{F}_2)^m$ is the set of all binary vectors of length m, which is also a linear m-dimensional space over the two-element field $\mathbb{F}_2$. The number of information bits (dimension) of the RM_r code over $\mathbb{F}_2$ is

$$K(r) = \sum_{j=0}^{r} \binom{m}{j}$$

and its code distance is $d = 2^{m-r}$. Let R be the given $k(r) \times N$ generator matrix for an RM_r code of length N.[2] Denote by ε_r the ensemble of all matrices of the form $E = HR\Gamma$, where H is taken from the set of all $k(r) \times k(r)$ non-singular matrices over $\mathbb{F}_2$ and Γ is taken from the set of all $N \times N$ permutation matrices, i.e., the matrices with entries in $\mathbb{F}_2$ and a unique non-zero entry in each row and in each column. Define the automorphism group G_r of the RM_r code as the set of all permutation matrices Γ such that $R\,\Gamma = HR$. It is evident that

$$\mid \varepsilon_r \mid = h_k N! \mid G_r \mid^{-1}$$

, where h_k is the number of all non-singular binary $k \times k$ matrices, $k = k(r)$, $h_k = (2^k - 1)(2^k - 2) \cdots (2^k - 2^{k-1})$, $N!$ is the number of distinct matrices Γ, and $\mid G_r \mid$ is the cardinality of the automorphism group. Since that G_r is the general affine group of the space $(\mathbb{F}_2)^m$, hence $G_r = 2^m(2^m-1) \cdots (2^m-2^{m-1})$. The transmission of a secret message from a user y to a user x is preceded by the following actions. The user x chooses matrices $H = H_x$ and $\Gamma = \Gamma_x$, at random with equal probabilities and independently of the choice of other

users. Then he computes the matrix $E_x = H_x R\Gamma_x$ from the ensemble ε_r. The matrix E_x is the public key, and the matrices $H_x,\ \Gamma_x$ are his private key. In Sidelnikov's cryptosystem, there is another advanced type being considered the transmission rate is only slightly lower than in the original cryptosystem with ensemble ε_r. The generator matrix E is composed by u distinct $k \times N$ matrices $E_1, \cdots, E_u$. For eahc matrices $E_i, i = 1, \cdots, u$, are of the form $E_i = H_i R$ (R is the same generator matrix of the RM_r code), where $H_1, \cdots, H_u$ are non-singular $k \times k$ matrices with entries in the field $\mathbb{F}_2$. The public keys are $k \times uN$ and matrix E_x in the form:

$$E = || \ E_1 \cdots E_u \ || \ \Gamma$$

, where Γ is a $uN \times uN$ permutation matrix. The same as the previous model, we can calculate $\varepsilon_{u,r}$ as:

$$| \ \varepsilon_{u,r} \ | = (uN)!(h_k)^u | \ G_r \ |^{-u}(u!)^{-1}$$

In particular, the information transmitted in the cryptosystem considered is a vector e of length uN and weight at most t_u and the cipher text is in the form of $c = eD$, where D is the check matrix of the code $K(E)$ where D and E are related by the equality $ED^T = 0$, where D^T is the transpose of D.
To encryption, the user x builds the vectors $b = aE + e$ and $d = b\Gamma^{-1}$. The vector d is a sequence $(d'_1, \cdots, d'_u)$ of u corrupted RM_r code vectors of length N so that

$$d'_i = d_i + e_i,\ d_i = aH_iR,\ i = 1, 2, \cdots, u$$

and

$$w(e_1) + \cdots + w(e_u) = w(e) \leqslant t_U,\ (e_1, \cdots, e_u) = e\Gamma^{-1}$$

In terms of decryption, let $t_u = ut + u - 1$, where t is the maximum number of errors, which can be corrected by a decoding algorithm for the RM_r code of length N. Then $w(e_i) \leqslant t$ for at least one i. Thus, the decoding algorithm, when applied sequentially to each d'_i, properly recovers at least one vector e_i, namely, the vector with the minimal weight. Knowing $e_i = d'_i + aH_iR$, we may figure out the information vector a. Then all the remaining vectors $e_1, \cdots, e_u$ can be recovered.

4.2 An Attack Against Sidelnikov's Cryptosystem

In this section we will briefly introduce an attack against Sidelnikov's Cryptosystem, which makes Reed-Muller Code being a bad choice [6]. Let σ be a permuation of {1,2,..,n}, we can denote the code word C^σ as the

permutated code word in C corresponding to σ. Let the code word in be C^σ $(x_1, x_2, ..., x_n) \in C^\sigma$, then $(x_{\sigma^{-1}(1)}, x_{\sigma^{-1}(2)}, ..., x_{\sigma^{-1}(n)}) \in C$

Let σ be an unknown permutation and let $C = R(r, m)^\sigma$.

1. Find enough codewords C such that they belongs to $R(r, m)^\sigma$ and they form a basis $R(r, m)^\sigma$
2. Repeat the previous step with decreasing r until we can get the $R(1, m)^\sigma$
3. Determine the permutation τ such that $R(1, m)^{\tau(\sigma)} = R(1, m)$, and then $R(1, m)^{\tau(\sigma)} = R(1, m)$, we can decode the whole message.

According to Lorenze [6], the operating time complexity of the attacking algorithm is:

$$O(P(n))e^{O(P(\log n))}$$

where $P(n)$ is the polynomial of n.We can see that the complexity is in subexponential complexity, based on which we can tell that the attack is relatively efficient.

5 Conclusion

In this project we conducted a deep research on the error correcting code and coding theory, and then we build a connection between the coding theory and the cryptography. Finally we introduce two cryptosystems which base on certain coding scheme and then provide the defense or attack on these systems. Finding the relationship between the cryptography and coding theory helps us profoundly comprehend both error correcting code, which is normally at the bottom of the internet protocol stacks (The physical layer and datalink layer), and the public key cryptography, which is usually on top of the protocal stacks (The application layer).

References

[1] Communication Networks, 2nd Edition, Alberto Leon-Garcia and Indra Widjaja, McGraw Hill.

[2] V. M. Sidelnikov, A public-key cryptosystem based on binary Reed-Muller codes, Discrete Mathematics and Applications, 4 No. 3, 1994

[3] Minder, Lorenz. "Cryptography based on error correcting codes." algo. epfl. ch (2007).

[4] Bernstein, Daniel J.; Lange, Tanja; Peters, Christiane (8 August 2008). "Attacking and defending the McEliece cryptosystem". Proc. 2nd International Workshop on Post-Quantum Cryptography. Lecture Notes In Computer Science. 5299: 31–46.

[5] David B.M., Nascimento A.C.A., Müller-Quade J. (2012) Universally Composable Oblivious Transfer from Lossy Encryption and the McEliece Assumptions. In: Smith A. (eds) Information Theoretic Security. ICITS 2012. Lecture Notes in Computer Science, vol 7412. Springer, Berlin, Heidelber

[6] Minder, Lorenz, and Amin Shokrollahi. "Cryptanalysis of the Sidelnikov cryptosystem." Advances in Cryptology-EUROCRYPT 2007 (2007): 347-360.

[7] Cryptography Based on Error Correcting Codes, Lorenz Minder, 2007.

[8] Elementary Algebraic Geometry, Klaus Hulek, 2003.

[9] Stern, J.: A method for finding codewords of small weight. In: Cohen, G., Wolf- mann, J. (eds.) Coding Theory and Applications 1988. LNCS, vol. 388, pp. 106–113. Springer, Heidelberg (1989)

[10] Moon, Todd K. "Error correction coding." Mathematical Methods and Algorithms. Jhon Wiley and Son (2005).