GENERAL STRUCTURE AND PROPERTIES OF CYCLIC CODES IN GF(2)

ABSTRACT

This article provides a comprehensive analysis of cyclic codes in GF(2), focusing on their general structure and key properties. Cyclic codes are characterized by their generator polynomials, which define their structure and play a crucial role in encryption and decryption processes. The cyclical shift property, inherent in cyclic codes, facilitates efficient implementation using shift register circuits, making them practical for real-world applications. The discussion highlights the algebraic properties that distinguish cyclic codes from other linear block codes, emphasizing their ability to detect and correct errors.

Key words: cyclic codes; algebraic properties; generator function.

1. INTRODUCTION

Cyclic codes are defined by generator polynomials, which play a crucial role in their structure. Cyclic codes possess several algebraic properties that distinguish them from other types of linear block codes. One of the fundamental structural characteristics of cyclic codes is their cyclical shift property. The ability of cyclic codes to detect and correct errors is a key property. Cyclic codes exhibit a range of code lengths and dimensions, which affect their storage and transmission efficiency. Various performance metrics, such as the code rate, error detection and correction capabilities, and decoding complexity, characterize the operational properties of cyclic codes. The efficiency and complexity of decoding algorithms, such as syndrome decoding or Berlekamp-Massey algorithm, are critical properties of cyclic codes. Cyclic codes find applications in diverse areas, including telecommunications, storage systems, and digital communication.

In 1978, McEliece proposed the first code-based cryptosystem. This system is a general encryption setting for coding theory. Cyclic codes form an important class of linear codes by means of error correcting. They have a very interesting algebraic structure.

Binary cyclic codes were first introduced by Prange in 1957, and have been the topic of hundreds of papers since. A lot of development have been done on cyclic codes.

In a study by Calkavar.S.(2013) he investigateg the minimal codes words in the binary cyclic codes and obtained that:

Let C be an $[n, k]$-cyclic code over F_2 with generator polynomial g(x) = $g_0 + g_1 x + \cdots g_{n-k} x^{n-k}$ of degree n-k. In the $[n, k]$-binary cyclic codes C generated by g(x) , there are altogether 2^k- 2 minimal codewords. He concluded that these results can be used for the secret sharing based on the binary cyclic codes.

Petrenko et al (2019), developed an encryption method based on cyclic BCH codes. They used RSA encryption algorithm and error correcting codes. Efficient method of constructing code-based cryptosystems was developed by Calkavur $ Guzeltepe (2022). This approach is based on the One Time Pad cryptosystem.

2. PRELIMINARIES

Definittion:

A k × n generator matrix G obtained by writing the base vectors of the code C as rows of G is called a generator matrix of the linear $[n, k]$-code C.

Definition:

A code C is cyclic if C is a linear code, any cyclic shift of a codeword is also a codeword, i.e whenever $a_0\, a_1 \cdots a_{n-1} \in$ C, then also $a_{0-1} a_0\, a_1 \cdots a_{n-2} \in$ C.

Theorem

A code C in $F_q[x]/x^n - 1$ is a cyclic code if and only if C satisfies the following two conditions:a(x), b(x) $\in$ C $\leftrightarrow$ a(x)+b(x) $\in$ C , a(x) $\in$ C and r(x) $\in F_q[x]/x^n - 1 \rightarrow$r(x)a(x)$\in$ C,

Where we denote by $F_q[x]/x^n - 1$ the set of polynomials in $F_q[x]$ of degree less than deg $x^n - 1$.

Basics

Cyclic code has a generating function. If C $= C_0, C_1, \ldots, C_{n-1}$ is a codeword, its generating function is defined to be the polynomial

C(x) = $C_0 + C_1 x + \ldots + C_{n-1} x^{n-1}$, where x is indeterminate.

If p and m are integers with m > 0,then "p mod m" denotes the remainder obtained when p is divided by m; thus p mod m is the unique integer r such that p - r is divisible by m, and $0 \leq r \leq m - 1$. Similarly, if P(x) and Q(x) are polynomials, P(x) mod M(x) denotes the remainder when P(x) is divided by M(x);thus P(x) mod M(x) is the unique polynomial R(x) such that P(x) - R(x) is divisible by M(x), and R(x)< deg M(x).

The following Lemma lists the most important properties of the mod operator for polynomials.

Lemma 1

. (i) If deg P(x) < deg M(x), then P(x) mod M(x) = P(x).

(ii) If M(x)| P(x), then P(x) mod M(x) = 0.

(iii) (P(x) + Q(x)) mod M(x) = P(x) mod M(x) + Q(x) mod M(x).

(iv) (P(x)Q(x)) mod M(x)=(P(x)(Q(x) mod M(x))) mod M(x).

(v) If M(x)| N(x), then (P(x) mod N(x)) mod M(x) = P(x) mod M(x).

3. GENERAL STRUCTURE OF CYCLIC CODES OVER GF_2

a) Cyclic Shift Invariance

Cyclic Shift Invariance is a fundamental and defining property of cyclic codes in coding theory. This property implies that if you take any codeword from a cyclic code and cyclically shift its bits (rotate the bits either to the left or right), the resulting sequence of bits will also be a valid codeword of the same code. This characteristic is particularly useful for applications that involve circular data structures or cyclic interferences, such as data storage on circular media.

When a codeword is cyclically shifted to the left, the first bit becomes the last bit of the new codeword. For example, a left shift of the codeword 1101 results in 1011.Conversely, in a cyclic right shift, the last bit moves to the first position. Thus, a right shift of 1101 yields 1110.

In algebraic terms, cyclic codes can be represented using polynomials over a finite field. Each codeword can be associated with a polynomial, where the coefficients of the polynomial correspond to the bits of the codeword.

For a codeword represented by the polynomial C(x) = $C_0 + C_1x + \ldots + C_{n-1}x^{n-1}$, a cyclic shift can be modeled by multiplying $C(x)$ by x (modulo x^n-1). The modulo operation ensures that the polynomial remains within the degree limits suitable for the code's length, n. Given that a cyclic code is defined by a generator polynomial $g(x)$ that divides $x^n - 1$ (where n is the length of the codewords), any codeword $C(x)$ in the code can be expressed as $C(x) = a(x)\square g(x)$ for some polynomial $a(x)$. This multiplication, and the fact that $g(x)$ divides x^n-1, ensures that the cyclic shift of any codeword is also a multiple of $g(x)$ and therefore remains a valid codeword.

One of the fundamental structural characteristics of cyclic codes is their cyclical shift property. This property states that if $c(x)$ is a codeword, then $x\square c(x)$ is also a codeword.

Theorem 2

If C = ($C_0, C_1, \ldots, C_{n-1}$) is a codeword with generating function C(x) = $C_0 + C_1x + \ldots + C_{n-1}x^{n-1}$, then the generating function C^R(x) for the right cyclic shift C^R is given by the formula C^R(x) = xC(x) mod $(x^n - 1)$

Proof: Since C(x) = $C_0 + C_1x + \ldots + C_{n-1}x^{n-1}$, we have

xC(x) =C_0x + … + $C_{n-2}x^{n-1} + C_{n-1}x^n$

$C^R(x) = C_{n-1} + C_0x + \cdots + C_{n-2}x^{n-1}$. Hence xC(x) − $C^R(x)$= $C_{n-1}(x^n - 1)$. Since deg $C^R(x)$< deg$(x^n - 1)$, and xC(x) − $C^R(x)$ is a multiple of $(x^n - 1)$, the result follows from the definition of the mod operation.

(b) codewords as multiple of g(x)

Every codeword c(x) C can be expressed as c(x) = p(x)g(x) where p(x) is a polynomial of degree less than n-r. This ensures that c(x) is a multiple of g(x), which guarantees the cyclic property.

Theorem 3

If C is an (n, k) cyclic code, and if C(x) is a codeword in C, then for any polynomial P(x), [P(x)C(x)]n is also a codeword in C.

Proof: suppose $P(x) = \sum_{i=0}^{k} P_i x^i$. *Then*

$[P(x)C(x)]_n = \left[\left(\sum_{i=0}^{k} P_i x^i\right)C(x)\right]_n$

$= \sum_{i=0}^{k} P_i \left[x^i C(x)\right]_n$

by lemma 1 (iii).But by remarks proceeding the statement of this theorem. $\left[x^i C(x)\right]_n$ is a codeword for each i, and so, since the code is linear, the linear combination $\sum P_i \left[x^i C(x)\right]_n$ is also a codeword.

Lemma 4

Suppose that C is a cyclic code with generator polynomial g(x). (a) If g' (x) is another generator polynomial, then g' (x) = λg(x), for some nonzero element λ ∈ F. (b) If P(x) is a polynomial such that [P(x)]n is a codeword, then g(x) divides P(x).

(c) Generator Function

Theorem 5

(a) If C is an (n, k) cyclic code over F, then its generator polynomialis a divisor of $(x^n - 1)$. Furthermore, the vector If C = $(C_0, C_1, \ldots, C_{n-1})$ is in the code if and only if the corresponding generating function C(x) = $C_0 + C_1 x + \ldots + C_{n-1} x^{n-1}$ is divisible by g(x). If k denotes the dimension of C, then k = n − deg g(x).

(b) Conversely, if g(x) is a divisor of $(x^n - 1)$, then there is an (n, k) cyclic code with g(x) as its generator polynomial and k = n − deg g(x), namely, the set of all vectors $(C_0, C_1, \ldots, C_{n-1})$ whose generating functions are divisible by g(x).

Proof of (a)

First, let P(x) = $(x^n - 1)$ in Lemma 4 (b) ; $[P(X)]_n$ = 0, which is of course a codeword, and so g(x) divides $(x^n - 1)$. Next, by Theorem 3, any vector of length n whose generating function is a multiple of g(x) is a codeword. Conversely, if C(x) = $C_0 + C_1 x + \ldots + C_{n-1} x^{n-1}$ is a codeword, then $[C(X)]_n$ = C(x) and so Lemma 4 implies that g(x) divides C(x). Finally, the assertion about the degree of g(x) follows since C(x) =$C_0 + C_1 x + \ldots + C_{n-1} x^{n-1}$ is a multiple of g(x) if and only if C(x) = g(x)I(x), where

deg I ≤ n − 1 − deg g.

Proof of (b)

Suppose that g(x) is a divisor of $(x^n - 1)$. Then C(x)=$(C_0, C_1, \ldots, C_{n-1})$ is a multiple of g(x) if and only if C(x) = g(x)I(x), where deg g + deg I ≤ n − 1. Thus, the set of all such words is an (n, k) linear code, where k = n − deg g. To show that this code is cyclic, we must show that the right cyclic shift of any codeword is also a codeword. Thus let I(x)g(x) be any codeword; by Theorem 2 its right cyclic shift is $[xI(x)g(x)]_n$. But since g(x) divides $(x^n - 1)$, we have $[xI(x)g(x)]_n$ mod g(x)=[xI(x)g(x)] mod g(x) (by Lemma 1 (v)) = 0 (by Lemma 1(ii)), which proves that $[xI(x)g(x)]_n$. is a multiple of g(x), so the code is indeed cyclic.

Theorem 5 shows the importance of the generator polynomial of cyclic code. A closely-related and equally important is the parity- check polynomial for cyclic code, which is denoted by h(x), and defined by h(x) = $\frac{x^n-1}{g(x)}$

d) Parity- check polynomial

Parity-check polynomial: cyclic codes also have a corresponding parity-check polynomial, denoted as h(x). The –parity check polynomial of a cyclic code is derived from its generator polynomial. The parity-check polynomial is defined such that it divides evenly into $(x^n - 1)$ without any remainder, which means that $(x^n - 1)$ = g(x)h(x) for some generator polynomial g(x).

(e) **Generator and Parity-Check Matrices**

Generator and parity-check matrices are derived from generator and parity-check polynomial respectively.

Corollary 6

If C is an (n, k) cyclic code with generator polynomial g(x) = $g_0 + g_1x + \ldots + g_rx^r$ (with r = n − k), and parity-check polynomial h(x) = $h_0 + h_1x + \ldots h_kx^k$, then the following matrices are generator and parity-check matrices for C:

$$G1 = \begin{bmatrix} g_0 & g_1 & \cdots & \cdots & g_r & \cdots & \cdots & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & \cdots & h_o & o & \cdots & 0 \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & 0 & g_0 & g_1 & \cdots & \cdots & g_r \end{bmatrix} = \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}$$

$$H1 = \begin{bmatrix} h_k & h_{k-1} & \cdots & \cdots & h_0 & \cdots & \cdots & \cdots & 0 \\ 0 & h_k & h_{k-1} & \cdots & \cdots & h_0 & o & \cdots & 0 \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & 0 & h_k & h_{k-1} & \cdots & \cdots & h_0 \end{bmatrix} = \begin{bmatrix} h(x) \\ xh(x) \\ \vdots \\ x^{r-1}h(x) \end{bmatrix}$$

4.GENERAL STRUCTURE AND PROPERTIES OF CYCLIC CODES OVER GF (2)

a) Error-Correction Capability

The ability of cyclic codes to detect and correct errors is a key property. This property is influenced by factors such as the minimum distance of the code and the design of the generator polynomial**.** A code is said to correct e errors if a decoder and is capable of correcting any pattern of e or fewer errorsintroduced by the channel. The significant of the Hamming distance for a code becomes apparent with the following theorem.

Theorem

Let C be an (n, k)-code having distance d = 2e+1.Then C can correct e errors. If used for error detection only, C can detect 2e errors.

b) Cyclic Redundancy Check (CRC)

Cyclic codes are often used in CRC schemes for error detection. CRC polynomials are derived from cyclic codes and they possess properties such as error-detecting capability in and simplicity in implementations. Cyclic codes often exhibit low redundancy, meaning that they require relatively few additional bits to archive a given level error of correction capabilities. This efficiency makes cyclic codes attractive for bandwidth limited resources-constrained applications.

A transmitted bit can be received in error, due to "noise" on the transmission channel. If we are dealing with voice or video data, the occurrence of errors in a small percentage of bits is quite tolerable, but in many other cases it is crucial that all bits be received intact. There are many methods which have been developed to detect errors, applied at different levels of the seven layer model. Of course, no method can detect all errors, but a number of methods in use today are amazingly effective. The Cyclic Redundancy Checking (CRC) appends a few (typically 16 or 32) bits to the end of the bit string for a message and sends out the extended string. The receiver then performs a computation which would yield 0 if no bits of the message had been in error; if the result is not 0, then the receiver knows that there has been an error in one or more bits. (But if the result is 0, this does not necessarily mean there was no error) (Matloff, 2001)

(c) Modular Arithmetic implementation.

Cyclic codes can be implemented using modular arithmetic operation, particularly in binary fields GF (2). This implementation simplifies hardware design and reduces computational complexity, making cyclic codes suitable for hardware-based applications.The core of modular arithmetic lies in the concept of modulo operations. Modulo, often denoted as a mod m or a (mod m), is the remainder when a is divided by m. This operation plays a pivotal role in defining congruence and exploring the cyclic nature of modular arithmetic. In its essence, modulo arithmetic is about working with remainders. The expressions a ≡ b (mod m) signifies that a and b have the same remainders when divided by m. this relation, called congruence, forms the foundation of modular arithmetic. One of the fascinating aspects of modulo arithmetic is its ability to create cyclic patterns. For instance, consider n^2 mod 4 for various values of n. the results exhibit a cyclic pattern 0, 1, 0, 1, …, revealing the periodic nature of modulo operations.

(d) Encoding and decoding

Encoding in cyclic codes is performed by dividing the message polynomial by generator polynomial. The message polynomial m(x) is multiplied by x^k where k is the number of redundancy bits) to make room for the code polynomial. The remainder of the division, denoted as r(x). represents the redundancy bits of the codeword. The encoded codeword is obtained by appending r(x) to x^km(x).

Decoding in cyclic codes involves finding the error locator polynomial and error evaluator polynomial. The syndromes of the received polynomial are computed and used to find the error locator polynomial using the Berlekamp-Massey algorithm or

other methods. Once the error locator polynomial is found, the error locations can be determined, and the error values can be corrected accordingly.

(e) Closure under addition and multiplication

A Galois field, denoted as $GF(q)$, is a finite field with q elements, where q is a prime power, i.e., $q = pm$ for some prime number p and positive integer m. The field $GF(q)$ is often referred to as a Galois field of order q or a finite field of order q. In a Galois field $GF(q)$, addition and multiplication are performed modulo q. This means that the result of any addition or multiplication operation is reduced modulo q to ensure that the result remains within the field. Addition and multiplication operations in $GF(q)$ satisfy closure, associativity, and commutativity properties. There exists an additive identity (zero) and a multiplicative identity (one) in $GF(q)$. Every non-zero element in $GF(q)$ has an additive inverse and a multiplicative inverse. The characteristic of a Galois field $GF(q)$ is the smallest positive integer p such that p multiplied by any element of the field yields the additive identity.

(f) Burst Error Correction

On many channels of practical importance, errors, when they occur, tend to occur in bursts. Physically, a burst of errors occurs when for some reason the channel noise severity increases for a brief time, and then returns to normal. Cyclic codes are particularly effective at correcting burst errors, where consecutive bits in a codeword are corrupted. The cyclic shifting property allows the code to detect and correct burst errors efficiently, making cyclic codes suitable for applications with bursty noise. If a codeword C is transmitted, and is received as **R = C + E,** then the error vector **E** is called a burst of length **b** if the nonzero components of **E** are confined to **b** consecutive components.

(g) Minimum Distance:

The minimum distance of a cyclic code is the smallest Hamming distance between any pair of distinct codewords in the code. It determines the error-detecting and error-correcting capabilities of the code. Cyclic codes with larger minimum distances can correct a greater number of errors. A code with minimum distance d can detect up to $d-1$ errors in a codeword. This is because if fewer than d errors are introduced into a codeword, the resulting word is still closer to the original codeword than to any other codeword in the code.

A code with minimum distance d can correct up to $(d-1)/2$ errors. This capability arises because, with $(d-1)/2$ errors, the erroneous codeword is closer to the original codeword than to any other codeword. Therefore, a decoding process can correctly infer the original codeword by identifying the nearest valid codeword.A binary Hamming code of length $2^m - 1$ is any linear code whose parity-check matrix has as columns the $2^m - 1$ nonzero binary vectors of length m, arranged in any order. For example, the following parity-check matrix defines a (7, 4) Hamming code:

$$H = \begin{bmatrix} 0 & 1 & 11 & 1 & 00 \\ 1 & 0 & 11 & 0 & 10 \\ 1 & 1 & 01 & 0 & 01 \end{bmatrix}$$

There are of course $(2^m - 1)!$ ways to order these columns, and although any one of these orderings produces a perfect single-error correcting code, some orderings are better than others from an implementational standpoint.

(h) Parallelization

Cyclic codes can be parallel efficiently allowing for concurrent processing of multiple codewords. This property enables high-throughput communication systems and can improve overall system performance.

5. DISCUSSION

The structure and properties of cyclic codes in GF(2) play a vital role in cryptography, offering several advantages and capabilities that are essential for secure communication and data protection. The generator polynomial g(x) is the cornerstone of cyclic codes. It helped in defining the of validity of codewords and their structure. In encryption we used g(x) to ensure that each codeword are uniquely represented,providing a structured way to encode messages.The g(x)l ensured that the codewords were consistently generated which is critical for reliable encryption.

Having a well defined g(x) allowed for predictable transformations,which was essential for designing secure encryption algorithms. Cyclic invariance ensured error resilience and simplified the hardware implementation of coding systems, allowing for efficient encryption and decryption process using shift registers. The parity check polynomial ensured that the received codeword has not been altered,providing a means to verify the integrity of the encrypted message.The use of modular arithmetic ensured that the encoded data fits within a specified range,preventing overflow and maintaining the integrity of encrypted messages.parity check assist in detection of errors by checking if the received codeword is divisible by generator polynomial. The length of the cyclic code determined the size of the codewords and affects the overall efficiency and performance of the encryption and decryption processes. Longer codes provide stronger error detection and correction capabilities.

The closure properties of cyclic codes under addition and multiplication has ensured that encoded messages remain within the code space, preserving the security and integrity of the communication channel. CRC is a type of cyclic code used for error detection in data transmission. It generates a checksum based on the data content, which is appended to the message for error detection at the receiver's end.

6. CONCLUSION

We presented the unique structures and properties of cyclic codes over gf(2). They play a crucial role in cryptography, providing essential mechanisms for secure communication and data protection. Several key properties stand out as particularly important in cryptographic applications. However, these structures and properties can be applied in data encryption and decryption of data in further studies.

REFERENCES

Calkavur, S., & GÜZELTEPE, M. (2022). Secure encryption from cyclic codes. *Sigma Journal of Engineering and Natural Sciences*, *40*(2), 380-389.

Calkavar, S.(2013) .Binary codes and Minimal codewordes. Computer Technology and Application 4, 486-489.

Matloff, N. (2001). Cyclic redundancy checking. *Department of Computer Science, University of California*.

McEliece, R.J. (1978) A public-key cryptosystem based on algebraic coding theory, DSN Progress Report, 1978, pp. 114-116

Prange, E. (1957). Error detection and multiple-error correction. IRE Transactions on Electronic Computers, EC-6(1), 83-89

Roth, R. M. (2006). Introduction to coding theory. *IET Communications*, *47*(18-19), 4.

Huffman, W. C., Kim, J. L., & Solé, P. (2021). *Concise encyclopedia of coding theory*. Chapman and Hall/CRC.

Petrenko, V., Ryabtsev, S., Ryabtsev, S., Pavlov, A. S., &Apurin, A. A. (2019). Development of an encryption method based on cyclic codes. In A. Jones & B. Johnson (Eds.), 21. Proceedings of the 21st International Workshop on Computer Science and Information Technologies (CSIT 2019) Atlantis Press, (Vol. 3, pp. 196-201).