APPLICATION OF CYCLIC CODES OVER GF_2 ON ENCRYPTION AND DECRYPTION OF DATA

ABSTRACT

The constant increase in the number of new types of cyber threats actualizes the issues of their information transfer. This paper presents a secure encryption and decryption method using cyclic codes, inspired by the One-Time Pad cryptosystem, for smart grid communications. We convert plaintext into binary, chunk it into segments, and pad these to align with a generator polynomial. These segments are then transformed into polynomials, encrypted, and secured with a One-Time Pad. The decryption process reverses these steps, recovering the original plaintext.Our findings show that cyclic codes effectively maintain data integrity and security, demonstrating robustness. In a practical application, we securely transmitted the message "shed load" within a smart grid system. Cyclic codes provided a reliable and efficient means of securing data, accurately reversing the encryption steps and ensuring data fidelity. These results underscore the potential of cyclic codes to enhance smart grid communication security, offering a balance of security, efficiency, and robustness.

Key words: cyclic codes; encryption; decryption.

1 INTRODUCTION

The development of information and telecommunication technologies in the modern world has reached the level that they have penetrated into all areas of activity. In this regard,the requirements for for information security are constantly growing.

While cyclic codes have been studied extensively in coding theory, their application in practical cryptography systems, especially in smart grids, is less explored. Traditional methods of securing smart grid communications may not adequately address the increasing threat in the digital world.

The encrypted data (ciphertext) is received by the intended recipient or decryption system and the recipient obtains the appropriate decryption key required to decrypt the ciphertext. The decryption key must match the encryption key used during the encryption process.

The recipient applies the decryption algorithm to the ciphertext using the decryption key. The decryption algorithm reverses the transformation applied during encryption, recovering the original plaintext.

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

In 1978, McEliece proposed the first code-based cryptosystem. Original McEliece cryptosystem was low in encryption rate and had large key size. Baldi et al, (2008) improved the McEliece cryptosystem.

In a study by Calkavar. S. (2013) he investigated 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.in this cryptosystem cyclic codes were uesd for detection and correction of errors.

Efficient method of constructing code-based cryptosystems was developed by Calkavur and Guzeltepe (2022). This approach is based on the One Time Pad cryptosystem. This approach is very fast and the keys are short.The method can be applied by different organizations to ensure data is securely transmitted.

2. PRELIMINARIES

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 $a_{n-1} a_0\, a_1 \cdots a_{n-2} \in$ C.

Definition:

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:

Encryption is the process of converting plaintext or any other type of data into ciphertext, which is unintelligible and unreadable to unauthorized users or entities.

Definition:

Decryption process is done using decryption algorithm that is converting ciphertext, which is encrypted or encoded data, back into its original plaintext form, making it readable and intelligible to authorized users.

Theorem

Let C ≠ $\{\mathbf{0}\}$ be a cyclic code of length n over F.

(1) Let g(x) be a monic code polynomial of minimal degree in C. Then g(x) is uniquely determined in C, and

$$C = \{q(x)g(x) | q(x) \in F[x]_{n-r}\},$$

Where r = deg (g(x)),in particular, C has dimension n-r.

(2) The polynomial g(x) divides $x^n - 1$ in F$[x]$.

PROOF. As C≠ $\{\mathbf{0}\}$, it contains nonzero code polynomial, each of which has a unique monic scalar multiple.Thus there is a monic polynomial g(x) in C of minimal degree. Let this degree be r, unique even if g(x) is not. By remarks preceding the theorem, the set of polynomials

$$C_0 = \{q(x)g(x) | q(x) \in F[x]_{n-r}\}$$

Is certainly contained in C, since it is composed of those multiples of the code polynomial g(x) with the additional property of having degree less than n. Under addition and scalar multiplication C_0 is an F-vector space of dimension n-r. The polynomial g(x) is unique monic polynomial of degree r in C_0.

To prove (1), we must show that every code polynomial $c(x)$ is an $F[x]$- multiple of $g(x)$ and so is in the set C_0. By the Division Algorithm we have

$$C(x) = q(x)g(x) + r(x),$$

for some q(x), r(x)$\in F[x]$ with deg (r(x)) $<$ r = deg (g(x)),therefore

$$r(x) = c(x) - q(x)g(x)$$

By defination c(x) $\in$ C and q(x)g(x) is in C_0 (as c(x) has degree less than n).Thus by linearity, the right hand side of this equation is in C, hence the remainder term r(x) is in C. If r(x) was nonzero, then it would have a monic scalar multiple belonging to C and of smaller degree than r. But this would contradict the original choice of g(x). Therefore r(x) = 0 and c(x) = q(x)g(x), as required.

Next let

$$x^n - 1 = h(x)g(x) + s(x)$$

for some s(x) of degree less than deg(g(x)). Then, as before,

$$s(x) = (-h(x)g(x) \pmod{x^n - 1}$$

belongs to C. Again, if s(x) is not zero, then it has a monic scalar multiple belonging to C and smaller degree than that of g(x), a contradiction. Thus s(x) = 0 and g(x)h(x) = $x^n - 1$, as in (2).

The polynomial g(x) is called the generator polynomial for the code C.

The polynomial h(x) $\in F[x]$ determined by

$$g(x)h(x) = x^n - 1$$

is the check polynomial of C.

Among the first codes used practically were the cyclic codes which were generated using shift registers. Prange (1957) noticed that this class of cyclic codes has a rich algebraic structure.

The linear code C of length n is a cyclic code if it is invariant under a cyclic code shift

$$C = (c_0, c_1, c_2 \ldots, c_{n-2}, c_{n-1}) \in \mathbf{C}$$

If and only if

$$\tilde{C} = (c_{n-1}, c_0, c_1 \ldots, c_{n-3}, c_{n-2}) \in \mathbf{C}.$$

As C is invariant under this single right cyclic shift, by iteration it is invariant under any number of right cyclic shifts. As a single left cyclic shift is the same as n - 1 right cyclic shifts, C is also invariant under a single left cyclic precisely when it is invariant under all cyclic shift. Therefore the linear code **C** is cyclic precisely when it is invariant under all cyclic shifts.

It is convenient to take Cyclic codes consisting of polynomials as well as codewords. With every word

$$a = (a_0, a_1, \ldots a_i, \ldots, a_{n-2}, a_{n-1}) \in F^n$$

We associate the polynomial of degree less than n

$$a(x) = (a_0 + a_1 x. + \ldots a_i x^i + \ldots + a_{n-1} x^{n-1}) \in F[x]_n.$$

If **c** is a codeword of code **C**, then we call **c**(x) the associated code polynomial. And the shifted codeword $\widetilde{\boldsymbol{C}}$ has associated code polynomial

$$\widetilde{\boldsymbol{C}}(\boldsymbol{x}) = c_{n-1} + c_0 x + c_1 x^2 + \ldots + c_i x^{i+1} + \ldots + c_{n-2} x^{n-1}$$

PROPOSITION

If C is the cyclic code of length n with check polynomial h(x), then

$$C = \{c(x) \in F[x]_n \mid c(x)h(x) = 0 \ (mod x^n - 1)\}.$$

PROOF. Indeed if $c(x) \in C$,then by theorem1 there is a q(x) with c(x) = q(x)g(x). But then

$$c(x)h(x) = q(x)g(x) = q(x)(x^n - 1) = 0(\ mod x^n - 1).$$

Now consider an arbitrary polynomial $c(x) \in F[x]_n$ with

$$c(x)h(x) = p(x)(x^n - 1)$$

Then

$$c(x)h(x) = p(x)(x^n - 1)$$

$$= p(x)g(x)h(x),$$

Hence

$$(c(x) - p(x)g(x))h(x) = 0$$

As $g(x)h(x) = x^n - 1$,we do not have h(x) = 0 .Therefore

$$c(x) - p(x)g(x) = 0$$

$$\text{And } c(x) = p(x)g(x),$$

as required.

If we are in possession of generator polynomial $g(x) = \sum_{j=0}^{r} g_j x^j$ for the cyclic code C, then we can easily construct a generator matrix for C consider

$$G = \begin{bmatrix} g_0 g_1 \cdots \cdots \cdots g_{r-1} & g_r & 0 & 0 & \cdots & 0 \\ 0 & g_1 g_1 \cdots \cdots \cdots & \cdots & g_{r-1} g_r & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & g_0 g_1 & \cdots & \cdots & \ldots \ldots & g_{r-1} g_r \end{bmatrix}$$

The matrix G has n columns and k = n-r rows; so the first row, row g_0, finishes with a string of 0's of length k-1. Each successive row is cyclic shift of the previous row: $g_i = \bar{g}$ for

I = 1,…,k-1. As g(x)h(x) = $x^n - 1$,We have

$$g_0 h_0 = g(0)h(0) = 0^n - 1 \neq 0$$

In particular $g_0 \neq 0$ (and $h_0 \neq 0$), therefore G is echelon form (although likely not reduced). In particular the k = dim(C) rows of G are linearly indipendent. Clearly the rows of G belong to C, so G is indeed a generator matrix for C, sometimes called the cyclic generator matrix of C.

Secret key cryptosystem

A cryptosystem is called secret key cryptosystem if some secret piece of information (the key) has been agreed first between any two parties that want to communicate through the cryptosystem. There are some basic types of secret key cryptosystem:

Substitution based cryptosystems:- they substitute the characters of plaintext for another character.

Monoalphabetic cryptosystems: - they use a fixed substitution, one character is always replaced with the same group of symbols.

Polyalphabetic cryptosystems:- the substitution keeps changing during the encryption.

Transposition based cryptosystem:- they only traspose the characters of plaintext,for example permission/impression.

Stream cryptosystems:- each block of plaintext of plaintext is encrypted using a differeent key. Stream cryptosystem are more appropriate in some applications (telecommunication), usually are simpler to implement, faster and have no error propagation.

Block cryptosystem:- the same key is used to encrypt arbitrary long plaintext block by block.

One time pad cryptosystem:- a cryptosystem for encoding data using a key of same length as the data. If m is the plaintext , s is the key and c is the cryptotext, then the encryption algorithm e_s *is* $c = e_s$(m) = m+s and the decryption algorithm d_s

is $m = d_s(c)$= d+s

3. APPLICATION OF CYCLIC CODES OVER GF_2 TO ENCRYPTION OF DATA

Anencryption using One Time Pad cryptosystem constructed by Calkavur and Guzeltepe (2022) will be used here. The encryption scheme consists of the following parameters.

- ✓ Set up
- ✓ Key Generation
- ✓ Encryption
- ✓ Decryption

Key Generation Procedure:

i. Choose a codeword of a cyclic code of length n with generation matrix g(x) of degree r is called m.

ii. Compute a cyclic shift of the codeword is called s.

iii. Calculate c = m + s.

iv. The plaintext is m and the private key is s

Encryption

Plaintext ;$m_i = a_1(x)g(x)$, where $0 \leq i \leq p^{n-r}$

Key: $s_i = x^t a_i(x)\, g(x)$, where t is the number of shift and s = $s_1 s_2 \ldots s_n$.

Ciphertext: $c_i = m_i + s_i$.

We assume that $a_i(x)g(x) \neq a_j(x)g(x)$ for $i \neq j, 0 \leq i, j \leq p^{n-r}$

Decryption:

Ciphertext: c_i

Plaintext:$m_i = c_i + (p-1)s_i$

Correctness: The correctness of the encryption scheme relies on the structure of a cyclic code. It is known that any cyclic shift of a cyclic code is also a codeword. Every cyclic shift of a codeword consist of a key and this key has the same length with the plaintext. Furthermore the key is used only once.

In a smart grid system, the controller communicates various types of messages to different components to ensure efficient, reliable, and secure grid operations. Here are some examples of communication messages sent by the controller to the smart grid; Load Control Commands like, load shedding-this is a Command to reduce or disconnect certain loads to prevent overloading the grid. We take the example of shed load and communicate the message from controller to smart grid.
The first step is to convert plain text shed load to binary, followed by putting it to one message string, chunk the message to 7 bits to able to use the proposed generator polynomial, encrypt the codewords add OTP then decrypt it back to original plaintext.

CHARACTER	ASCII	BINARY
s	115	01110011
h	104	01101000
e	101	01100101
d	100	01100100
space	32	00100000
l	108	01101100
o	111	01101111

a	97	01100001
d	100	01100100

01110011 01101000 01100101 01100100 00100000 01101100 01101111 01100001 01100100

Remove the spaces to make it one string.

0111001101101000011001010110010000100000011011000110111101100001011001 00100

Divide the message to 7 bits string and pad the last cordword to have 7 bits

0111001, 1011010, 0001100, 1010110, 0100001, 0000001, 1011000, 1101111, 0110000, 1011001, 0000000 converted to polynomials they will be $x^5 + x^4 + x^3 + 1, x^6 + x^4 + x^3 + x, x^3 + x^2, x^6 + x^4 + x^2 + x, x^5 + 1, 1, x^6 + x^4 + x^3, x^6 + x^5 + x^3 + x^2 + x + 1, x^5 + x^4, x^6 + x^4 + x^3 + 1, 0$

Encryption scheme used based on these codewords is given in the following

$m_i = a_i(x)g(x), s_i = x^t a_i(x)g(x), (let\ t = 1), c_i = m_i + s_i, 1 \leq \leq 11$

Now we use this encryption scheme by using the generator polynomial g(x) = 1 + x + x^3

$m_1 = a_1(x)g(x) = (1 + x + x^3)(x^5 + x^4 + x^3 + 1) = 0 = 0000000$

$s_1 = xa_1(x)g(x) = x(0) = 0 = 0000000$

$c_1 = m_1 + s_1 = 0+0 = 0 = 0000000$

$m_2 = a_2(x)g(x) = (x^6 + x^4 + x^3 + x)\ (1 + x + x^3) = x^5 + x^4 + x^3 + x = 0111010$

$s_2 = xa_2(x)g(x) = x(x^5 + x^4 + x^3 + x) = x^6 + x^5 + x^4 + x^2 = 1110100$

$c_2 = m_2 + s_2 = (x^5 + x^4 + x^3 + x) + (x^6 + x^5 + x^4 + x^2) = x^6 + x^3 + x^2 + x = 1001110$

$m_3 = a_3(x)g(x) = (x^3 + x^2)(x^3 + x + 1) = x^6 + x^5 + x^4 + x^2 = 1110100$

$s_3 = xa_3(x)g(x) = x(x^6 + x^5 + x^4 + x^2) = x^6 + x^5 + x^3 + 1 = 1101001$

$c_3 = m_3 + s_3 = (x^6 + x^5 + x^4 + x^2) + (x^6 + x^5 + x^3 + 1) = x^4 + x^3 + x^2 + 1 = 0011101$

$m_4 = a_4(x)g(x) = (x^6 + x^4 + x^2 + x)(x^3 + x + 1) = x^6 + x^3 + x^2 + x = 1001110$

$s_4 = xa_4(x)g(x) = x(x^6 + x^3 + x^2 + x) = (x^4 + x^3 + x^2 + 1) = 0011101$

$c_4 = m_4 + c_4 = 1010011$

$m_5 = a_5(x)g(x) = (x^5 + 1)(x^3 + x + 1) = x^6 + x^5 + x^3 + 1 = 1101001$

$s_5 = xa_5(x)g(x) = x(x^6 + x^5 + x^3 + 1) = x^6 + x^4 + x + 1 = 1010011$

$c_5 = m_5 + s_5 = x^5 + x^4 + x^3 + x = 0111010$

$m_6 = a_6(x)g(x) = 1(x^3 + x + 1) = x^3 + x + 1) = 0001011$

$s_6 = xa_6(x)g(x) = x(x^3 + x + 1) = x^4 + x^2 + x = 0010110$

$c_6 = m_6 + s_6 = (x^3 + x + 1) + (x^4 + x^2 + x) = x^4 + x^3 + x^2 + 1 = 0011101$

$m_7 = a_7(x)g(x) = (x^6 + x^4 + x^3)(x^3 + x + 1) = x^5 + x^3 + x^2 = 0101100$

$s_7 = xa_7(x)g(x) = x(x^5 + x^3 + x^2) = x^6 + x^4 + x^3 = 1011000$

$c_7 = (x^5 + x^3 + x^2) + (x^6 + x^4 + x^3) = x^6 + x^5 + x^4 + x^2 = 1110100$

$m_8 = a_8(x)g(x) = (x^6 + x^5 + x^3 + x^2 + x + 1)(x^3 + x + 1) = x^6 + x^3 + x^2 + x = 1001110$

$s_8 = xa_8(x)g(x) = x(x^6 + x^3 + x^2 + x = x^4 + x^3 + x^2 + 1 = 0011101$

$c_8 = m_8 + s_8 = x^6 + x^4 + x + 1 = 1010011$

$m_9 = a_9(x)g(x) = (x^5 + x^4)(x^3 + x + 1) = x^6 + x^4 + x + 1 = 1010011$

$s_9 = xa_9(x)g(x) = x(x^6 + x^4 + x + 1) = x^5 + x^2 + x + 1 = 0101011$

$c_9 = m_9 + s_9 = x^6 + x^5 + x^4 + x^2 = 1110100$

$m_{10} = a_{10}(x)g(x) = (x^6 + x^4 + x^3 + 1)(x^3 + x + 1) = x^5 + x^3 + x^2 + 1 = 0101101$

$s_{10} = xa_{10}(x)g(x) = x(x^5 + x^2 + x + 1) = x^6 + x^3 + x^2 + x = 1011010$

$c_{10} = m_{10} + s_{10} = x^6 + x^5 + x^4 + x^2 + x + 1 = 1110111$

$m_{11} = a_{11}(x)g(x) = 0\ (x^3 + x + 1) = 0 = 0000000$

$s_{11} = x\ a_{11}(x)g(x) = x(0) = 0 = 0000000$

$c_{11} = m_{11} + s_{11} = 0 + 0 = 0 = 0000000$

This ciphertext is sent to the smart grid which should perform the decryption process by applying decryption key which reverses the ciphertext to plaintext.

Decryption process

$m_i = c_i + (p - 1)s_i$

$m_1 = s_1 + c_1 = 0000000$

$m_2 = s_2 + c_2 = 0111010$

$m_3 = s_3 + c_3 = 1110100$

$m_4 = s_4 + c_4 = 1001110$

$m_5 = s_5 + c_5 = 1101001$

$m_6 = s_6 + c_6 = 0001011$

$m_7 = s_7 + c_7 = 0101100$

$m_8 = s_8 + c_8 = 0001010$

$m_9 = s_9 + c_9 = 1010011$

$m_{10} = s_{10} + c_{10} = 0101101$

$m_{11} = s_{11} + c_{11} = 0000000$

We must perform polynomial division to extract our original polynomial.

m_1 divided by generator it results to this original polynomial $x^5 + x^4 + x^3 + 1 = 0111001$. It is performed as follows:

$$x^5 + x^4 + x^3 + 1$$

$$x^3 + x + 1 \overline{)\, x^8 + x^7 + x + 1}$$

$$x^8 + x^6 + x^5$$

$$x^7 - x^6 - x^5$$

$$x^7 + x^5 + x^4$$

$$x^6 + x^4 + x$$

$$x^6 + x^4 + x^3$$

$$x^3 + x + 1$$

$$x^3 + x + 1$$

$$0 \quad 0 \quad 0$$

$m_2 = x^6 + x^4 + x^3 + x = 1011010$

$m_3 = x^3 + x^2 = 0001100$

$m_4 = x^6 + x^4 + x^2 + x = 1010110$

$m_5 = x^5 + 1 = 0100000$

$m_6 = 1 = 0000001$

$m_7 = x^6 + x^4 + x^3 = 1011000$

$m_8 = x^6 + x^5 + x^3 + x^2 + x + 1 = 1101111$

$m_9 = x^5 + x^4 = 0110000$

$m_{10} = x^6 + x^4 + x^3 + 1 = 1011001$

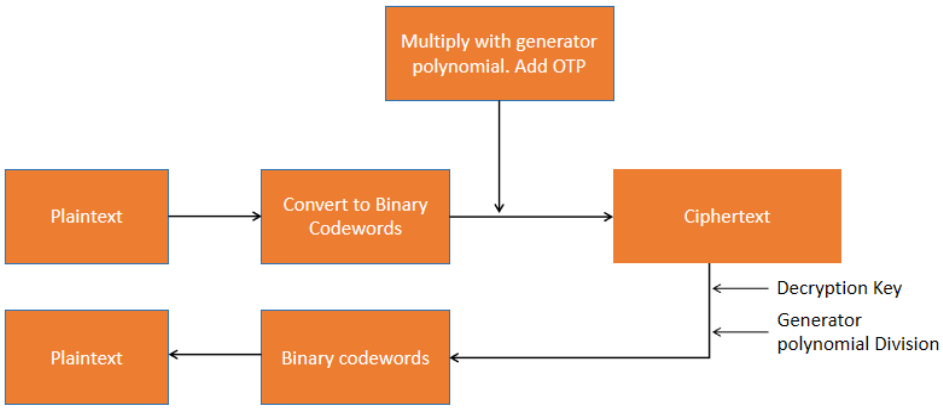
$m_{11} = 0 = 0000000$

We convert the 7 bits decrypted codewords to one string , then back to 8 bits for interpretation by computer.

01110011011010000110010101100100001000000110110001101111011000010110010000000

The last one was padded by 5 zeros to make 7-bits,we remove the zeros.these are the 8 bits ,

01110011, 01101000, 01100101, 01100100, 00100000, 01101100, 01101111, 01100001, 01100100, which is interpreted as 115, 104, 101, 100, 32, 108, 111, 97, 100, which represents the plain text ‘shed load’.



a diagram showing the encryption and encryption process

4.DISCUSSION

On application of cyclic codes in encryption and decryption of data. The example showcased how the plaintext message "shed load " was converted into its binary representation using ASCII encoding, chunked into manageable segments, and then padded to fit the generator polynomial requirements. These chunks were then transformed into polynomial representations to facilitate encryption. By multiplying the polynomial codes by a generator polynomial and adding a One-Time Pad (OTP), the plaintext was successfully encrypted into a ciphertext. This process underscores the effectiveness of cyclic codes in creating secure data streams that can resist unauthorized access and ensure data integrity. The decryption involved reversing the encryption steps by using the generator polynomial to decode the ciphertext back into its original polynomial form. This step-by-step reversal highlighted the robustness of cyclic codes in maintaining data fidelity through the entire encryption-decryption cycle. Cyclic codes over GF_2 offer a powerful tool for data encryption and decryption, providing a balance of security, reliability, and efficiency. Their application in smart grid communications, as demonstrated, highlights their potential to enhance critical infrastructure operations, ensuring that data integrity and security are maintained in the face of growing digital threats. However, both the sender and receiver must maintain perfect sychronization with the OTP, which can be difficult to achieve and maintain in dynamic network conditions. Further refinement can be done on robustsychronization mechanisms to ensure the seamless operation of OTP-based encryption on smart grid.

5.CONCLUSION

We presented an application of code-based cryptosystem to smart grid. The cryptosystem is based on One Time Pad. The One Time Pad is a proven unbreakable

encryption method. The One Time Pad cryptosystem method is an addictive stream cipher, where truly random keys is generated generated and then combined with the plaintext for encryption or with ciphertext for decryption by an “exclusive OR” (XOR) addition.

The exploration of cyclic codes and their properties, along with their application in data encryption and decryption within smart grids, presents numerous opportunities for further study. These can be done to improve on smart grid.

Investigating the integration of cyclic codes with new technologies such as IoT devices in smart grids to improve overall system resilience can also be tried.

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.

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).