Asian Research Journal of Mathematics

Volume 20, Issue 8, Page 142-151, 2024; Article no.ARJOM.120392
ISSN: 2456-477X

An Application of Cyclic Codes over GF_2 for Data Encryption and Decryption in Smart Grid Communications

Beatrice Gacheri Munjuri [a*], Loyford Njagi [a] and Josphine Mutembei [a]

[a] Department of Mathematics, Meru University of Science and Technology, Kenya.

Authors' contributions

This work was carried out in collaboration among all authors. All authors read and approved the final manuscript.

Article Information

DOI: https://doi.org/10.9734/arjom/2024/v20i8822

Open Peer Review History:
This journal follows the Advanced Open Peer Review policy. Identity of the Reviewers, Editor(s) and additional Reviewers, peer review comments, different versions of the manuscript, comments of the editors, etc are available here: https://www.sdiarticle5.com/review-history/120392

Original Research Article

Received: 25/05/2024
Accepted: 25/07/2024
Published: 13/08/2024

Abstract

There is increase in the number of new types of cyber threats which 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. AES and RSA are more complex to

**Corresponding author: Email: munjurib@gmail.com;*

Cite as: *Munjuri, Beatrice Gacheri, Loyford Njagi, and Josphine Mutembei. 2024. "An Application of Cyclic Codes over GF2 for Data Encryption and Decryption in Smart Grid Communications". Asian Research Journal of Mathematics 20 (8):142-51. https://doi.org/10.9734/arjom/2024/v20i8822.*

implement compared to the cyclic code encryption scheme. They require more computational resources for encryption and decryption. The cyclic code scheme is conceptually straightforward with polynomial operations. These results underscore the potential of cyclic codes to enhance smart grid communication security, offering a balance of security, efficiency, and robustness.

Keywords: Cyclic codes; encryption; decryption.

1 Introduction

The advancement of information and telecommunication technologies has become so extensive that they now impact every aspect of our lives. Consequently, the demand for information security is continually increasing.

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 intended recipient or decryption system receives the encrypted data (ciphertext) and acquires the necessary decryption key to decode the ciphertext. A decryption key must match the encryption key used during the encryption process [1].

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.

According to Abdullah [2] AES is providing much more security compared to DES, 3DES and ECC. However the implementation of AES algorithms is facing complexities as result of lengthy of the keys.

In 1957 Prange introduced Binary cyclic codes have been the topic of hundreds of papers since. Cyclic codes are under going a lot of developments.

In 1978, Mc Eliece proposed the first code-based cryptosystem. Original Mc Eliece cryptosystem was low in encryption rate and had large key size. Baldi et al, [3] improved the Mc Eliece cryptosystem by replacing the permutation matrix with dense transformation matrix.

In a study by Calkavar. S. [4] he investigated the minimal codewords in the binary cyclic codes and obtained that:

Let C be an $[n, k]$-cyclic code over F_2 with generator polynomial g(x) = $g_0 + g_1x + \cdots g_{n-k}x^{n-k}$ 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.

An encryption method based on cyclic BCH codes was developed by Petrenko et al [5]. They used RSA encryption algorithm and error correcting codes. In this cryptosystem cyclic codes were used for detection and correction of errors.

Efficient method of constructing code-based cryptosystems was developed by Calkavur and Guzeltepe [6]. 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. According to Bellovin.S.M. [7], Gilbert S.Vernam and Joseph O. Mauborgne are credited to invention of One Time Pad.

2. Preliminaries

Definition: A code C is considered cyclic if it is a a linear code and any cyclic shift of a codeword is also a codeword. In other words, if $a_0\, a_1 \cdots a_{n-1} \in$ C, then also $a_{n-1} a_0\, a_1 \cdots a_{n-2} \in$ C.

Definition: A k×n generator matrix G is formed by arranging the base vectors of the code C as rows of G. this matrix is referred to as generator matrix of the linear $[n, k]$-code C.

Definition: Encryption involves transforming plaintext into ciphertext using an encryption algorithm. Ciphertext 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 $\neq \{\mathbf{0}\}$ 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$\neq \{\mathbf{0}\}$, 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 belongs to 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) ∈ 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) ∈ F$[x]$ determined by

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

is the check polynomial of C.

One of the earliest types of codes applied in practice were cyclic codes, which were created using shift registers. In 1957, Prange observed that these cyclic codes possess a complex and 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

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

Since C is invariant under this single right cyclic shift, it remains invariant under any number of right cyclic shifts through iteration. Since a single left cyclic shift is the same as n - 1 right cyclic shifts, C is also invariant under a single left cyclic if and only if it is invariant under all cyclic shift. Consequently, the linear code **C** is cyclic precisely when it is invariant under all cyclic shifts.

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) ∈ 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) ∈ 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 .Hence

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

And c(x) = p(x)g(x),as required.

With generator polynomial g(x) = $\sum_{j=0}^{r} g_j x^j$ for the cyclic code C, then construction of a generator matrix for C is simple, consider

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

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

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

Particularly $g_0 \neq 0$ (and $h_0 \neq 0$), therefore G is echelon form. Specifically, the k = dim(C) rows of G are linearly indipendent. Obviously, the rows of G belong to C, thus G serves as generator matrix for C, often referred to as the cyclic generator matrix of C.

Secret key cryptosystem:

A cryptosystem is referred to as a secret key cryptosystem if a shared piece of confidential information (the key) is agreed upon beforehand by the parties wishing to communicate securely. There are several fundamental types of secret key cryptosystems:

1. Substitution-based cryptosystems: These systems replace the characters of the plaintext with different characters.
2. Monoalphabetic cryptosystems: These use a fixed substitution where each character is always replaced by the same symbol or group of symbols.
3. Polyalphabetic cryptosystems: In these systems, the substitution changes continually throughout the encryption process.
4. Transposition-based cryptosystems: These systems rearrange the characters of the plaintext, such as transforming "permission" to "impression."
5. Stream cryptosystems: Each block of plaintext is encrypted using a different key. Stream cryptosystems are often more suitable for certain applications, such as telecommunications, because they are typically simpler to implement, faster, and do not propagate errors.
6. Block cryptosystems: The same key is used to encrypt arbitrarily long plaintext, processing it in blocks.

One time pad cryptosystem:- a cryptosystem for encoding data using a key of the 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

An encryption using One Time Pad cryptosystem constructed by Calkavur and Guzeltepe [6] will be used here. The encryption scheme consists of the following parameters.

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

Key Generation Procedure:

1. Select a codeword m from a cyclic code of length n with generation matrix g(x) of degree r.
2. Compute a cyclic shift of the codeword is denotet s.
3. Calculate c = m + s.
4. The plaintext is m and the private key is s

Encryption:

- ,Plaintext ; $m_i = a_1(x)g(x)$, where $0 \leq \mathrm{i} \leq p^{n-r}$
- .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: $c_i = m_i + \mathrm{s_i}$.

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

Decryption:

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

Correctness: The correctness of the encryption scheme depends on the structure of a cyclic code. It is known that any cyclic shift of a cyclic code remains a codeword. Each cyclic shift of a codeword serves as a key, and this key has the same length as the plaintext. Additionally, 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 operation. The smart grid is an upgraded version of the 20th-century electrical grid, incorporating two-way communications and distributed intelligent devices [8]. These two-way flows of electricity and information can enhance the delivery network.

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 [9].

The first step is to convert a plaintext '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 the original plaintext [10].

List 1. Examples of communication messages sent by the controller to the smart grid

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.

011100110110100001100101011001000010000001101100011011110110000101100100

Divide the message to 7 bits string and pad the last codeword 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

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

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

$m_1 = a_1(x)g(x) = (1+x+x^3)(x^5+x^4+x^3+1) = 0 = 0000000$
$s_1 = xa_1(x)g(x) = x(0) = 0 = 0000000$
$c_1 = m_1 + s_1 = 0+0 = 0 = 0000000$
$m_2 = a_2(x)g(x) = (x^6+x^4+x^3+x)(1+x+x^3) = x^5+x^4+x^3+x = 0111010$
$s_2 = xa_2(x)g(x) = x(x^5+x^4+x^3+x) = x^6+x^5+x^4+x^2 = 1110100$
$c_2 = m_2 + s_2 = (x^5+x^4+x^3+x)+(x^6+x^5+x^4+x^2) = x^6+x^3+x^2+x = 1001110$
$m_3 = a_3(x)g(x) = (x^3+x^2)(x^3+x+1) = x^6+x^5+x^4+x^2 = 1110100$
$s_3 = xa_3(x)g(x) = x(x^6+x^5+x^4+x^2) = x^6+x^5+x^3+1 = 1101001$
$c_3 = m_3 + s_3 = (x^6+x^5+x^4+x^2)+(x^6+x^5+x^3+1) = x^4+x^3+x^2+1 = 0011101$
$m_4 = a_4(x)g(x) = (x^6+x^4+x^2+x)(x^3+x+1) = x^6+x^3+x^2+x = 1001110$
$s_4 = xa_4(x)g(x) = x(x^6+x^3+x^2+x) = (x^4+x^3+x^2+1) = 0011101$
$c_4 = m_4 + c_4 = 1010011$
$m_5 = a_5(x)g(x) = (x^5+1)(x^3+x+1) = x^6+x^5+x^3+1 = 1101001$
$s_5 = xa_5(x)g(x) = x(x^6+x^5+x^3+1) = x^6+x^4+x+1 = 1010011$
$c_5 = m_5 + s_5 = x^5+x^4+x^3+x = 0111010$
$m_6 = a_6(x)g(x) = 1(x^3+x+1) = x^3+x+1) = 0001011$
$s_6 = xa_6(x)g(x) = x(x^3+x+1) = x^4+x^2+x = 0010110$
$c_6 = m_6 + s_6 = (x^3+x+1)+(x^4+x^2+x) = x^4+x^3+x^2+1 = 0011101$
$m_7 = a_7(x)g(x) = (x^6+x^4+x^3)(x^3+x+1) = x^5+x^3+x^2 = 0101100$
$s_7 = xa_7(x)g(x) = x(x^5+x^3+x^2) = x^6+x^4+x^3 = 1011000$
$c_7 = (x^5+x^3+x^2)+(x^6+x^4+x^3) = x^6+x^5+x^4+x^2 = 1110100$
$m_8 = a_8(x)g(x) = (x^6+x^5+x^3+x^2+x+1)(x^3+x+1) = x^6+x^3+x^2+x = 1001110$
$s_8 = xa_8(x)g(x) = x(x^6+x^3+x^2+x = x^4+x^3+x^2+1 = 0011101$
$c_8 = m_8 + s_8 = x^6+x^4+x+1 = 1010011$
$m_9 = a_9(x)g(x) = (x^5+x^4)(x^3+x+1) = x^6+x^4+x+1 = 1010011$
$s_9 = xa_9(x)g(x) = x(x^6+x^4+x+1) = x^5+x^2+x+1 = 0101011$
$c_9 = m_9 + s_9 = x^6+x^5+x^4+x^2 = 1110100$
$m_{10} = a_{10}(x)g(x) = (x^6+x^4+x^3+1)(x^3+x+1) = x^5+x^3+x^2+1 = 0101101$
$s_{10} = xa_{10}(x)g(x) = x(x^5+x^2+x+1) = x^6+x^3+x^2+x = 1011010$
$c_{10} = m_{10} + s_{10} = x^6+x^5+x^4+x^2+x+1 = 1110111$
$m_{11} = a_{11}(x)g(x) = 0(x^3+x+1) = 0 = 0000000$
$s_{11} = x\,a_{11}(x)g(x) = x(0) = 0 = 0000000$
$c_{11} = m_{11}+s_{11} = 0+0 = 0 = 0000000$

The output of encryption is the ciphertext $c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, and\ c_{11}$.

This ciphertext is sent to the smart grid for decrypting process by applying the 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:

$$x^5 + x^4 + x^3 + 1$$
$$x^3 + x + 1 \,\overline{)\, x^8 + x^7 + x + 1}$$
$$\underline{x^8 + x^6 + x^5}$$
$$x^7 - x^6 - x^5$$
$$\underline{x^7 + x^5 + x^4}$$
$$x^6 + x^4 + x$$
$$\underline{x^6 + x^4 + x^3}$$
$$x^3 + x + 1$$
$$\underline{x^3 + x + 1}$$
$$0 \quad 0 \quad 0$$

$m_2 = x^6 + x^4 + x^3 + x = 1011010$
$m_3 = x^3 + x^2 = 0001100$
$m_4 = x^6 + x^4 + x^2 + x = 1010110$
$m_5 = x^5 + 1 = 0100000$
$m_6 = 1 = 0000001$
$m_7 = x^6 + x^4 + x^3 = 1011000$
$m_8 = x^6 + x^5 + x^3 + x^2 + x + 1 = 1101111$
$m_9 = x^5 + x^4 = 0110000$
$m_{10} = x^6 + x^4 + x^3 + 1 = 1011001$
$m_{11} = 0 = 0000000$

The output for decryption is the message we get after performing polynomial division i.e $m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, and\ m_{11}$

We convert the 7 bits (all equivalent for m_i) decrypted codewords to one string.

01110011011010000110010101100100001000000110110001101111011000010110010000000

The last one was padded by 5 zeros to make 7-bits, we remove the zeros. Then chunk the above string to 8 bits as follows;

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’.

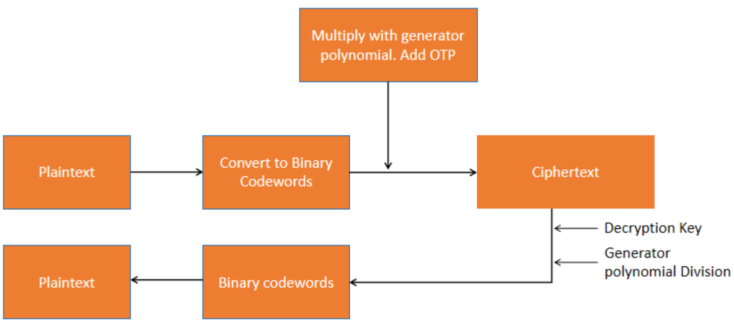

Fig. 1. Diagram showing the encryption and encryption process

4 Discussion

Cyclic codes can be applied 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 robust sychronization mechanisms to ensure the seamless operation of OTP-based encryption on the 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. One Time Pad cryptosystem method is an addictive stream cipher, where truly random keys are generated and then combined with the plaintext for encryption or with ciphertext for decryption by an “exclusive OR” (XOR) addition. The cyclic code encryption scheme used in smart grids offers a unique blend of error correction and encryption capabilities, making it suitable for secure communication in smart grid environments. However, it faces challenges related to key management and scalability. In contrast, AES is highly secure, efficient, and well-standardized, making it a popular choice for many smart grid applications. RSA and ECC provide strong security for key exchange and resource-constrained environments, respectively, but come with their own implementation complexities and performance trade-offs. Future work could explore hybrid approaches that combine the strengths of these different encryption methods to enhance the overall security and efficiency of smart grid communications. Investigating the integration of cyclic codes with new technologies such as Internet of Things devices in smart grids to improve overall system resilience can also be tried.

Disclaimer (Artificial intelligence)

Author(s) hereby declare that generative AI technologies such as Large Language Models, etc have been used during writing or editing of manuscripts. This explanation will include the name, version, model, and source of the generative AI technology and as well as all input prompts provided to the generative AI technology

Details of the AI usage are given below:

1. gpt-3.5-turbo 0125

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Asif M, Asamoah JK, Hazzazi MM, Alharbi AR, Ashraf MU, Alghamdi AM. A novel image encryption technique based on cyclic codes over Galois field. Computational Intelligence and Neuroscience. 2022;2022(1):1912603.

[2] Abdullah A. Advanced Encryption Standard (AES) algorithm to encrypt and decrypt data; 2017.

[3] Baldi M, Bodrato M, Chiaraluce F. A new analysis of the McEliece cryptosystem based on QC-LDPC codes, in: Security and Cryptography for Networks, LNCS, Springer. 2008;5229:246-262.

[4] Calkavar S. Binary codes and minimal codewords. Computer Technology and Application. 2013;4:486-489.

[5] Petrenko V, Ryabtsev S, Ryabtsev S, Pavlov AS, Apurin AA. 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. 2019;3:196-201.

[6] Calkavur S, Güzeltepe M. Secure encryption from cyclic codes. Sigma Journal of Engineering and Natural Sciences. 2022;40(2):380-389.

[7] Bellovin SM. Frank Miller: Inventor of the one-time pad. Cryptologia. 2011;35(3):203-222.

[8] Hu J, Lanzon A. Distributed finite-time consensus control for heterogeneous battery energy storage systems in droop-controlled microgrids. IEEE Transactions on Smart Grid. 2019;10(5):4751-4761.

[9] McEliece RJ. A public-key cryptosystem based on algebraic coding theory, DSN Progress Report. 1978;114-116.

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

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
https://www.sdiarticle5.com/review-history/120392