Project Report

On

IMPLEMENTATION OF CRYPTOGRAPHIC ALGORITHM FOR SECURED COMMUNICATION IN MALAYALAM LANGUAGE USING PYTHON PROGRAMMING

Submitted

in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE

In

MATHEMATICS

By

FEBINA ANN MARIA P.S

(AB22BMAT029)

Under the supervision of

Dr. SUSAN MATHEW PANAKKAL

DEPARTMENT OF MATHEMATICS

ST. TERESA'S COLLEGE (AUTONOMOUS)

ERNAKULAM, KOCHI – 682011

APRIL 2025

ST. TERESA'S COLLEGE (AUTONOMOUS) , ERNAKULAM

THE FEAR OF THE LORD IS THE BEGINNING OF WISDOM
ST. TERESA'S COLLEGE (AUTONOMOUS) ESTD 1925

CERTIFICATE

This is to certify that the dissertation entitled, **IMPLEMENTATION OF CRYPTOGRAPHIC ALGORITHM FOR SECURED COMMUNICATION IN MALAYALAM LANGUAGE USING PYTHON PROGRAMMING** is a bonafide record of the work done by **FEBINA ANN MARIA P.S** under my guidance as partial fulfillment of the award of the degree of **Bachelor of Science in Mathematics** at St. Teresa's College (Autonomous), Ernakulam affiliated to Mahatma Gandhi University, Kottayam. No part of this work has been submitted for any other degree elsewhere.

Date:10-01-2025

Place: Ernakulam

[illegible]

Dr. Susan Mathew Panakkal

Assistant professor,

Department of Mathematics,

St.Teresa's College(Autonomous),

Ernakulam

ST. TERESA'S COLLEGE (AUTONOMOUS) DEPARTMENT OF MATHEMATICS AND STATISTICS ERNAKULAM

[illegible]

Dr. Elizabeth Reshma M T

Assistant professors and Head ,

Department of Mathematics,

St.Teresa's College(Autonomous),Ernakulam.

External Examiners

1: [illegible] 30/4/25

2: ....

Dr. SREEJA K.U. (PEN:716628)
Assistant Professor
Department of Mathematics
Maharaja's College
Ernakulam - 682 011

DECLARATION

I hereby declare that the work presented in this project is based on the original work done under the guidance of Dr. Susan Mathew Panakkal , Assistant Professor , Department of Mathematics , St. Teresa's College (Autonomous) , Ernakulam and has not been included in any other project submitted previously for the award of any degree.

Febina

FEBINA ANN MARIA P.S

(AB22BMAT029)

Place :Ernakulam.

Date: 10-01-2025

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude towards Dr. Susan Mathew Panakkal of the Department of Mathematics of St. Teresa's College who encouraged to carry this work. Her continuous invaluable knowledgeable guidance throughout this study helped me to complete the work upto this stage. I will always be thankful to you in this regard. I also express thanks to all those who have indirectly guided and helped in the completion of this project.

[illegible]

FEBINA ANN MARIA P.S

(AB22BMAT029)

Place: Ernakulam

Date: 10-01-2025

Contents

Chapter 1

INTRODUCTION TO CRYPTOGRAPHY

1.1 Introduction

“An original message is known as the plaintext, while the coded message is called the ciphertext. The process of converting from plaintext to ciphertext is known as enciphering or encryption; restoring the plaintext from the ciphertext is deciphering or decryption. The many schemes used for encryption constitute the area of study known as cryptography.”[1]

The need to conceal message has always been essential since humans moved out of caves and started living in groups. To comprehend the evolution of cryptography over time, we need to understand the role of “cipher”. A cipher is a set of specific rules defined for the “encryption” and “decryption”[2]. In ancient times ciphers were of the basic pattern which was very easy for anyone to crack the message manually. In order to avoid insecurity, the modern encryption techniques are being adopted instead of classical techniques. As a fruitful result of modern encryption techniques, there exists complex algorithms which may require decades of research to crack the encoded message in English language. There are different kinds of classical encryption techniques. One such basic and widely known algorithm is the “Caesar cipher” technique. In this technique, each character of a given text is replaced by a character on the basis of a fixed ‘’key”. But it is a weak method of cryptography which can be easily decrypted by the prediction of frequency of letters among the 26 English characters. The main objective of this article is to implement the cryptographic technique for Malayalam language. Without much complications we have sticked on to the classical method itself as the Malayalam language possesses a large set of alphabets and hence it has much larger vocabulary. So, it is highly impossible for anyone to predict or rather find the letter that is most commonly used. Thus, the concept of frequency analysis doesn’t hold for Indian regional languages. If that is the case, it would be sufficient for us to make necessary changes and implement the classical method itself as it would be simple and helpful even for a non- technical person to effectively use it.

1.2 Literature Review

Cryptography is the practice and study of securing information and communication through the use of codes so that only intended recipients can read and process it. While cryptographic methods have been well-developed in global languages, studies and implementations in regional languages like Malayalam are still in an evolving stage. This review aims to analyze the progress made in this area, using references from research on other Dravidian languages, such as Tamil, and a specialized study on Moolabhadra language.

1.Cryptography for Tamil Language

The paper “Cryptography for Tamil Language” by M. Vivek Prabu, N. Keerthana, and R. Karthika discusses cryptographic techniques tailored to the Tamil script and linguistic structure. Tamil, being a Dravidian language, has its own script complexities like vowel markers, ligatures, and consonant-vowel clusters that present challenges in encryption and decryption. The authors analyze the intricacies of Tamil phonetics and writing systems and explore how existing cryptographic techniques, such as substitution and transposition ciphers, can be adapted. They propose that an effective cryptographic system for Tamil must take into account the following:

Tamil words often contain agglutination, meaning a single word might represent an entire phrase or sentence. Thus, the encryption system needs to handle morphemes as basic units.

Tamil uses a unique Unicode range, and encoding standards must be strictly adhered to in cryptographic algorithms to prevent misinterpretations or errors in transmission.

2. Cryptography in Moolabhadra Language

The paper “An Analytical Approach to the Cryptography of Moolabhadra Language” by Chanchal Seraphine, Susan Mathew Panakkal, Sangeetha Chandran, and Aparna Sebastian offers insights into cryptographic techniques for the Moolabhadra language, which shares structural similarities with Malayalam, as both are derived from the Dravidian family of languages.

The authors suggest using homophonic substitution to manage the variety of phonetic symbols, Ensuring that similar sounds can be encrypted into multiple cipher representations. This approach adds complexity to the decryption process for attackers.

Cryptographic systems for Malayalam, inspired by research in Tamil and Moolabhadra languages, must account for the script's complexity, phonetic variety, and morphological features. More work is required to develop language-specific algorithms, and future research should focus on enhancing efficiency, security, and accessibility in cryptography for Malayalam speakers

1.3 History of cryptography in Malayalam language

The history of cryptography in the Malayalam language is a blend of traditional secrecy methods and modern encryption practices. In ancient Kerala, rulers and local kingdoms like Travancore and Cochin used coded messages and symbolic language to communicate sensitive information, especially in warfare and politics. Palm leaf manuscripts (Olakkuda), the primary medium for writing, sometimes contained stylized scripts or disguised texts to protect confidential information. Traditional art forms, folklore, and temple rituals also embedded cryptic messages through riddles and metaphors, reflecting early forms of hidden communication.

During the colonial period, the arrival of the Portuguese, Dutch, and British introduced Western cryptographic practices to Kerala. European powers used secret codes for military and administrative purposes, indirectly influencing local methods of discreet communication. Christian missionaries also employed encrypted language to safely translate and distribute religious texts in Malayalam.

In the modern era, cryptography in Malayalam has advanced with digital technology. The adoption of Unicode allowed the standard representation of Malayalam characters in digital systems, enabling secure data handling. Messaging platforms like WhatsApp and Signal provide end-to-end encryption for Malayalam text, ensuring privacy in communication. Kerala's e-governance initiatives also implement encryption protocols to protect citizen data.

Today, research is growing in developing cryptographic algorithms tailored for Malayalam, addressing challenges in cybersecurity and data privacy. With the integration of artificial Intelligence and natural language processing, the future of Malayalam cryptography holds potential for innovative, language-specific security solutions, blending cultural heritage with modern technology.

1.4 Objectives

- To understand and implement cryptographic techniques using Python for the Malayalam Language.
- To analyze the challenges of applying encryption and decryption to a non-Latin script like Malayalam.
- To create and test Python-based cryptographic algorithms for Malayalam text.

1.5 Keywords

Plaintext: This is the original intelligible message or data that is fed into the algorithm as input.

Secret key: The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.

Ciphertext: This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.

Encryption: The process of encoding a message using an algorithm and a key

Decryption: The process of decoding ciphertext back into plaintext using a key

Decryption algorithm: This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

Chapter 2

CRYPTOGRAPHIC ALGORITHM FOR MALAYALAM LANGUAGE

2.1 Mechanism

Malayalam, the official language of Kerala in India, uses a unique script derived from the Brahmi script, and it consists of 56 letters. Here’s a quick breakdown of the Malayalam alphabet:

1.Vowels (Swaraksharangal): There are 15 vowels in Malayalam, including the basic ones and some complex forms.

അ(a)	ആ(ā)	ഇ(i)	ഈ(ī)	ഉ(u)
ക	കാ	കി	കീ	കു
ഊ(ū)	ഋ(r)	എ(e)	ഏ(ē)	ഐ(ai)
കൂ	കൃ	കെ	കേ	കൈ
ഒ(o)	ഓ(ō)	ഔ(au)	അം(am)	അഃ(ah)
കൊ	കോ	കൗ	കം	കഃ

2.Consonants (Vyanjanaksharangal): Malayalam has 36 consonants, divided into five main groups based on articulation, such as velars, palatals, retroflexes, dentals, and labials.

ക**(ka)**	ഖ**(kha)**	ഗ**(ga)**	ഘ**(gha)**	ങ**(ṅa)**
ച**(ca)**	ഛ**(cha)**	ജ**(ja)**	ഝ**(jha)**	ഞ**(ña)**
ട**(ṭa)**	ഠ**(tha)**	ഡ**(da)**	ഢ**(dha)**	ണ**(ṇa)**
ത**(ta)**	ഥ**(tha)**	ദ**(da)**	ധ**(dha)**	ന**(na)**
പ**(pa)**	ഫ**(pha)**	ബ**(ba)**	ഭ**(bha)**	മ**(ma)**

യ**(ya)**	ര**(ra)**	ല**(la)**	വ**(va)**
ശ**(śa)**	ഷ**(ṣa)**	സ**(sa)**	ഹ**(ha)**
ള**(ḷa)**	ഴ**(za)**	റ**(ṟa)**	

3. Chillu (Pure Consonants): These are standalone consonants without the inherent vowel sound, specific to Malayalam script:

ൺ(ṇ) ൻ(n) ർ(r̲) ൽ(l) ൾ(l)

4.Sounds- chihnams: Here, the symbols corresponding to each vowels are given

◌്	◌ാ	◌ി	◌ീ	◌ു
അ	ആ	ഇ	ഈ	ഉ
◌ൂ	◌ൃ	െ◌	േ◌	ൈ◌
ഊ	ഋ	എ	ഏ	ഐ
െ◌ാ	േ◌ാ	◌ൗ	◌ം	◌ഃ
ഒ	ഓ	ഔ	അം	അഃ

5. Special Consonant Signs: Malayalam also has conjunct consonants, which combine two consonants and sometimes use a Virama (◌്) To indicate the absence of a vowel. Malayalam script has a unique way of representing each sound, which contributes to its distinct phonetic structure and fluid appearance.

2.2 Encryption Process

In the Malayalam language, there are 15 vowels (Swaraksharangal), 36 consonants (Vyanjana aksharangal), and 5 pure consonants (Chillu). For encryption, we can assign distinct key ranges to each category: vowels receive values from 1 to 15, consonants from 1 to 36, and pure consonants from 1 to 5. This structured key mapping enables tailored encryption within the Caesar cipher framework, enhancing both security and consistency for Malayalam script. To encrypt, a vowel key of 3 shifts each vowel character three positions forward to its corresponding cipher character. Similarly, a consonant key of 4 shifts each consonant character four positions forward, generating its respective cipher text. For pure consonants, a key of 2 shifts each character by two. positions. This systematic shifting forms the foundation for encrypting Malayalam letters through the Caesar cipher.

The following table provides the cipher characters corresponding to the plain characters for vowels (Swaraksharangal) using a vowel key of 3.

Plain text	**Cipher text**	**Plain text**	**Cipher text**	**Plain text**	**Cipher text**
അ	ഈ	ഊ	ഏ	ഒ	അ
ആ	ഉ	ഋ	ഐ	ഓ	ആ
ഇ	ഊ	എ	ഒ	ഔ	ഇ
ഈ	ഋ	ഏ	ഓ	അം	ഈം
ഉ	എ	ഐ	ഔ	അഃ	ഈഃ

The following table provides the cipher characters corresponding to the plain characters for consonants (Vyanjana aksharangal) using a consonant key of 4.

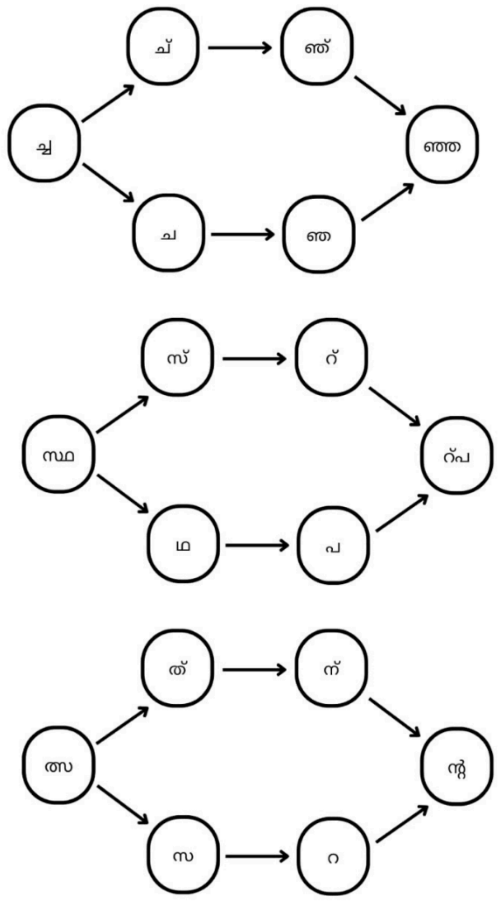
Plain text	**Cipher text**	**Plain text**	**Cipher text**
ക	ങ	ഞ	ഢ
ഖ	ച	ട	ണ
ഗ	ഛ	ഠ	ത
ഘ	ജ	ഡ	ഥ
ങ	ഝ	ഢ	ദ
ച	ഞ	ണ	ധ
ഛ	ട	ത	ന
ജ	ഠ	ഥ	പ
ഝ	ഡ	ദ	ഫ

Plain text	Cipher text	Plain text	Cipher text
ധ	ബ	ല	സ
ന	ഭ	വ	ഹ
പ	മ	ശ	ള
ഫ	യ	ഷ	ഴ
ബ	ര	സ	റ
ഭ	ല	ഹ	ക
മ	വ	ള	ഖ
യ	ശ	ഴ	ഗ
ര	ഷ	റ	ഘ

The following table provides the cipher characters corresponding to the plain characters for pure consonants (Chillu) using a consonant key of 2

Plain text	Cipher text
ർ	ൽ
ൻ	ൺ
ൽ	ൾ
ൺ	ർ
ൾ	ൻ

In Malayalam, we have compound characters (kootaksharangal) like ച്ച, ഞ്ച, ന്ത, മ്പ, and ത്സ. Before beginning the encryption process, these compound characters should be broken down into their respective vowel (swaraksharam) and consonant (vyanjan aksharam) components. Next, we can identify the corresponding cipher characters for each decomposed swaraksharam and vyanjan aksharam using the above tables. Finally, the two cipher characters obtained should be combined to form the cipher for the compound characters (kootaksharangal).



Now, we write down the cipher characters for each plain characters in the plain text using the above-mentioned tables and diagrammatic representations

അ	:	ഈ
വ	:	ഹ
രെ	:	ഷെ
ല്ലാ	:	സ്സാ
വ	:	ഹ
രും	:	ഷും
ന	:	ഭ
ല്ല	:	സ്സ
സ	:	റ
ന്തോ	:	ഭ്നോ
ഷ	:	ഴ
ത്തി	:	ന്നി
ലാ	:	സാ
ണ്	:	ധ്

**PLAIN TEXT : ** അവരെല്ലാവരും നല്ല സന്തോഷത്തിലാണ്

**CIPHER TEXT : ** ഈഹഷെസ്സാഹഷും ഭസ്സ റഭ്നോഴന്നിസാധ്

2.3 Decryption Process

Decryption is the process of converting encrypted data into original form. It's the opposite of encryption. It is a key concept, that helps to keep data safe and secure

**CIPHER TEXT : ** ഈഹഷെസ്സാഹഷും ഭസ്സ റഭ്നോഴന്നിസാധ്

To decode the ciphertext using the same key, apply a leftward shift to each character rather than a rightward shift. This reverse shift will gradually guide each encrypted character back to its original, plain form. As you proceed through the decryption process, if any compound characters appear in the ciphertext, they should be carefully divided into their separate components: vowel (Swaraksharam) and consonant (Vyanjan Aksharam). By isolating these elements, it becomes easier to accurately trace each part back to its plain text equivalent.

Once you have identified the individual plain characters by consulting the decryption tables, you can begin reconstructing the compound characters. Take each decoded vowel and consonant pair obtained from the separated cipher characters, and recombine them to form the corresponding compound character in the plain text. This method not only ensures a faithful translation of each encoded compound character but also preserves the intended meaning and structure of the original text. By following this systematic approach, you achieve a clear, accurate decryption that closely mirrors the content and form of the plaintext.

We get,

സ്സ→(സ് + സ)→ (ല് + ല) → ല്ല (സ്സ→ ല്ല) which is the corresponding plain character for the cipher character. Similarly, we obtain the respective plain texts for the corresponding cipher texts which is given below.

സ്സ→(സ്+ സ)→ (ല്+ ല) → ല്ല (സ്സ→ ല്ല)

മ്മ→ (മ്+ മ) → (പ്+ പ) → പ്പ (മ്മ →പ്പ)

ന്ന→ (ന്+ ന) → (ത്+ ത)→ത്ത(ന്ന→ത്ത)

CIPHER TEXT : ഈഹഷെസ്സാഹഷും ഭസ്സ റഭ്നോഴന്നിസാധ്

PLAIN TEXT : അവരെല്ലാവരും നല്ല സന്തോഷത്തിലാണ്

Chapter 3

CRYPTOGRAPHIC ALGORITHM FOR MALAYALAM LANGUAGE USING PYTHON PROGRAMMING

3.1 Algorithm

Step 1: Start the Program

Step 2: Initialise the program by defining the Section 1, Section 2, and Section 3 which categorise Malayalam characters into vowels (Swaraksharangal), consonants (vyanjan aksharam), and other consonantal sounds (Chillu).

Step 3: Define Special Compound Characters

(like "ക്ഷ", "ച്ച", "ട്ട", etc.) that are combinations of base characters and diacritical marks. These need to be split into separate characters before ciphering.

Step 4: Define shift values for each section of characters. The shifting logic works on a Caesar cipher, where each character is shifted by a fixed number depending on its section.In this case, Section 1 has a shift of 3, Section 2 has a shift of 4, and Section 3 has a shift of 2.

Step 5: Define a function get_section that checks in which section the character belongs to. This will help assign the correct cipher shift based on the character's section.

Step 6: Define a function shift_char that applies the Caesar cipher shift to a given character. This function determines the section for the character. Shifts the character accordingly within the defined section. Handles wrapping around the list if the shift goes beyond the bounds.

Step 7: Define a function handle_special_characters that splits the compound characters into individual base characters. This ensures that special characters like "ക്ഷ" are properly decoded and encoded.

Step 8: Define the encrypt_malayalam_text function that processes the input text, splits compound characters, and shifts each character accordingly using the cipher logic.

Step 9: Define the decrypt_malayalam_text function that decrypts the input text. This function reverses the encryption process by applying the negative shifts.(i.e., shift_section_1 = -3, shift_section_2 = -4, shift_section_3 = -2)

Step 10: Test the encryption and decryption process by providing an example input and printing the results.

3.2. Coding for Encryption

Some parts of the Python programming were completed using reference materials.[3][4]

```
# Define Malayalam characters for each section
section_1 = ["അ", "ആ", "ഇ", "ഈ", "ഉ", "ഊ", "ഋ", "എ", "ഏ", "ഐ", "ഒ", "ഓ", "ഔ"]
section_2 = ["ക", "ഖ", "ഗ", "ഘ", "ങ", "ച", "ഛ", "ജ", "ഝ", "ഞ", "ട", "ഠ", "ഡ", "ഢ", "ണ",
        "ത", "ഥ", "ദ", "ധ", "ന", "പ", "ഫ", "ബ", "ഭ", "മ", "യ", "ര", "ല", "വ",
        "ശ", "ഷ", "സ", "ഹ","ള", "ഴ", "റ"]
section_3 = ["ർ", "ൻ", "ൽ", "ൺ", "ൾ"]

# Create a dictionary for sections for easy lookup
all_characters = section_1 + section_2 + section_3

# Special compound characters
special_characters = {
  "ക്ഷ": ["ക", "്", "ഷ"],
  "ച്ച": ["ച", "്", "ച"],
  "ട്ട": ["ട", "്", "ട"],
  "ത്ത": ["ത", "്", "ത"],
  "ദ്ധ": ["ദ", "്", "ധ"]
}

# Define shift keys for each section
```

```
shift_section_1 = 3
shift_section_2 = 4
shift_section_3 = 2

# Function to get the section for a given character
def get_section(char):
    if char in section_1:
        return 1
    elif char in section_2:
        return 2
    elif char in section_3:
        return 3
    return None

# Function to perform Caesar cipher shift for a given character
def shift_char(char, shift):
    section = get_section(char)
    if section is None:
        return char  # Return as-is if no match found

    # Apply appropriate shift based on section
    if section == 1:
        shift = shift_section_1
    elif section == 2:
        shift = shift_section_2
    elif section == 3:
        shift = shift_section_3
    # Get the index of the character in the section and shift it
    section_list = section_1 if section == 1 else section_2 if section == 2 else section_3
    if char in section_list:
        idx = section_list.index(char)
        new_idx = (idx + shift) % len(section_list)
        return section_list[new_idx]
    return char


# Function to handle special characters and split compound letters
def handle_special_characters(text):
    for special, split_chars in special_characters.items():
        if special in text:
            text = text.replace(special, ''.join(split_chars))
    return text

# Main encryption function
def encrypt_malayalam_text(text):
    text = handle_special_characters(text)  # Split special compound letters first
    encrypted_text = []

    # Encrypt each character
    for char in text:
        encrypted_char = shift_char(char, 0)  # Default shift of 0 since shift is handled by section
        encrypted_text.append(encrypted_char)

    return ''.join(encrypted_text)
```

```
# Example Usage
input_text = "അവരെല്ലാവരും നല്ല സന്തോഷത്തിലാണ്"
encrypted_text = encrypt_malayalam_text(input_text)
print(f"Encrypted Text: {encrypted_text}")
```

Encrypted Text: ഈഹഷെസ്സാഹഷും ഭസ്സ റഭ്നോഴന്നിസാഡ്

3.3 Coding for Decryption

```
# Define Malayalam characters for each section
section_1 = ["അ", "ആ", "ഇ", "ഈ", "ഉ", "ഊ", "ഋ", "എ", "ഏ", "ഐ", "ഒ", "ഓ", "ഔ"]
section_2 = ["ക", "ഖ", "ഗ", "ഘ", "ങ", "ച", "ഛ", "ജ", "ഝ", "ഞ", "ട", "ഠ", "ഡ", "ഢ", "ണ",
          "ത", "ഥ", "ദ", "ധ", "ന", "പ", "ഫ", "ബ", "ഭ", "മ", "യ", "ര", "ല", "വ",
          "ശ", "ഷ", "സ", "ഹ","ള", "ഴ", "റ"]
section_3 = ["ർ", "ൻ", "ൽ", "ൺ", "ൾ"]

# Create a dictionary for sections for easy lookup
all_characters = section_1 + section_2 + section_3

# Special compound characters
special_characters = {
   "ക്ഷ": ["ക", "്", "ഷ"],
```

```
   "ച്ച": ["ച", "്", "ച"],
   "ട്ട": ["ട", "്", "ട"],
   "ത്ത": ["ത", "്", "ത"],
   "ദ്ധ": ["ദ", "്", "ധ"]
}
```

```
# Define shift keys for each section
shift_section_1 = -3
shift_section_2 = -4
shift_section_3 = -2

# Function to get the section for a given character
def get_section(char):
    if char in section_1:
        return 1
    elif char in section_2:
        return 2
    elif char in section_3:
        return 3
    return None

# Function to perform Caesar cipher shift for a given character
def shift_char(char, shift):
    section = get_section(char)
    if section is None:
        return char  # Return as-is if no match found

    # Apply appropriate shift based on section
    if section == 1:
        shift = shift_section_1
    elif section == 2:
        shift = shift_section_2
    elif section == 3:
        shift = shift_section_3
    # Get the index of the character in the section and shift it
    section_list = section_1 if section == 1 else section_2 if section == 2 else section_3
    if char in section_list:
        idx = section_list.index(char)
        new_idx = (idx + shift) % len(section_list)
        return section_list[new_idx]
    return char

# Function to handle special characters and split compound letters
def handle_special_characters(text):
    for special, split_chars in special_characters.items():
        if special in text:
            text = text.replace(special, ''.join(split_chars))
    return text

# Main Decryption function
def decrypt_malayalam_text(text):
    text = handle_special_characters(text)  # Split special compound letters first
    Decrypted_text = []

    # Decrypt each character
    for char in text:
        decrypted_char = shift_char(char, 0)  # Default shift of 0 since shift is handled by section
        decrypted_text.append(decrypted_char)

    return ''.join(decrypted_text)



# Example Usage
```

```
input_text = "ഈൗഹഷെസ്സാഹഷ്യം ഭസ്സ റഭ്നോഴന്നിസാധ്"
decrypted_text = encrypt_malayalam_text(input_text)
print(f"Decrypted Text: {decrypted_text}")
```

Decrypted Text: അവരെല്ലവരും നല്ല സന്തോഷത്തിലാണ്

3.4 CONCLUSION

In today's technological world, ensuring the security of data shared across global platforms is essential. To address this need, we have extended existing cryptographic algorithms to support the vast Malayalam language, making them more accessible through Python programming. This approach offers a practical and efficient alternative to manual processes, enabling safer communication across recognized languages through a user-friendly program that saves both time and effort.

When technology enables communication in one's preferred language, safeguarding information secrecy becomes crucial. Instead of relying on machine-level language, using the same natural language for encryption can be more reliable and authentic. This algorithm will help Malayalam speakers worldwide communicate securely whenever needed. Unlike English, Indian regional languages currently lack well-established frequency analysis. For languages with a larger alphabet set, the likelihood of a single letter being frequently repeated is not consistent, limiting the applicability of frequency analysis.

Additionally, the unique structure of Malayalam, which includes the decomposition of compound letters into their base elements, enhances encryption strength. Thus, traditional cryptographic methods may be highly effective for such languages, reducing the need to rely on advanced mechanisms for secure communication.

•

Chapter 4

Applications and limitations of Cryptography

4.1 Applications

Cryptography plays a vital role in securing digital communication and data for the Malayalam language across various sectors. It ensures secure communication by enabling encrypted messaging and email services in Malayalam, protecting personal and sensitive conversations. In document protection, cryptography safeguards legal, government, and official documents by preventing unauthorized access and verifying data integrity. Digital signatures allow for authenticity verification in Malayalam documents, ensuring that the sender's identity is confirmed and the content remains untampered. In data storage, encryption protects sensitive Malayalam data in databases and cloud platforms, while e-governance services rely on cryptography to secure online transactions and confidential citizen services. The healthcare sector benefits from encrypted electronic health records (HER) and secure doctor-patient communication in Malayalam. Additionally, cryptography secures financial transactions and digital payments, preventing fraud and identity theft. In education, it protects exam papers and academic records, and in web security, it encrypts Malayalam websites and applications, guarding against cyber threats.

4.2 limitations

Implementing cryptography for the Malayalam language faces several limitations. The complex script structure, with ligatures and compound characters, makes accurate encryption and decryption challenging. Text normalizationissues arise due to different Unicode representations, causing inconsistencies. The absence of tandardized cryptographic algorithms for Malayalam leads to compatibility issues and inefficiencies. Furthermore, processing Malayalam text increases computational overhead and storage requirements due to Unicode encoding. Limited user awareness about encryption tools and practices in the Malayalam-speaking community restricts its widespread adoption. Integration challenges also exist, as many encryption tools are not optimized for Malayalam, leading to software compatibility and cross-platform issues. Additionally, key management poses a challenge, with risks of key loss resulting in permanent data inaccessibility. Addressing these challenges requires the development of customized cryptographic solutions

tailored to the unique features of the Malayalam script to ensure secure and efficient digital communication.

4.3 Future works

The current implementation of cryptographic algorithms for the Malayalam language can be further enhanced in several ways. Future work could focus on developing more advanced encryption techniques tailored specifically to handle the complexity of Malayalam script, such as homomorphic encryption or post-quantum cryptography. Integrating machine learning techniques could improve anomaly detection and enhance security measures. Additionally, expanding the system to support secure multimedia communication, such as encrypting Malayalam audio and video content, would broaden its applications. Another area of improvement is optimizing the algorithm for performance, ensuring scalability for large datasets and high-speed communication systems. Developing user-friendly applications, such as secure messaging apps or file encryption tools with Malayalam language support, would also make this technology more accessible to non-technical users.

4.4 Real World Impacts

Implementing cryptographic solutions for the Malayalam language has significant real-world implications. It ensures that digital communication in Malayalam is secure, protecting sensitive data from cyber threats and unauthorized access. This is particularly valuable for government agencies, businesses, and educational institutions that operate primarily in Malayalam, as it safeguards confidential information. Moreover, it promotes digital inclusivity by allowing native Malayalam speakers to engage in secure online communication without relying on English-based systems. In the era of increasing cybercrimes, providing robust security for regional languages strengthens data privacy and boosts user trust in digital platforms. It can also encourage the development of secure e-governance portals, e-commerce platforms, and educational tools, ultimately contributing to the digital empowerment of Malayalam-speaking communities.

.

REFERENCES

[1]Stallings, W. (2017). Cryptography and network security: Principles and practice (7th ed., Global ed.). Pearson.86

[2]Prabua,M.V., Keerthana,N. and Karthika,R. (2023), Cryptography for tamil language.1

[3]Prabu, M. V., Keerthana, N., & Karthika, R. (2023). Implementation of cryptographic algorithm for secured communication in Tamil language using Python program. In AIP Conference Proceedings.6-9

[4]Seraphine, C., Panakkal, S. M., Chandran, S., & Sebastian, A. (2024). An analytical approach to the cryptography of Moolabhadra language. Malayalam Research Journal,17(1)13-15