# **Cryptography and Encryption Techniques**

# **A PROJECT REPORT**

# ***Submitted by***

T. Thanus Thapasvi - 192211121

G. Bhavesh Kumar - 192211169

*Under the guidance of*

*Dr. K. Vijaya Bhaskar*

***in partial fulfillment for the completion of the course***

***CSA1343 – Theory of Computation with Pushdown Automata***



**SIMATS ENGINEERING**

**THANDALAM**

**MARCH 2024**

# **Table Of Contents:**

| **S.No** | **Contents** | **Page No** |
| --- | --- | --- |
| 1 | Abstract | 3 |
| 2 | Introduction | 3 |
| 3 | Materials & Methods | 4 |
| 4 | Working Code | 5 - 6 |
| 5 | Results | 7 |
| 6 | Conclusion | 8 |

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# **ABSTRACT:**

Cryptography is the practice and study of techniques for secure communication in the presence of third parties. It involves various methods for encoding information so that only authorized parties can read it.

Text encryption techniques abstract the original message into a scrambled format using algorithms and keys, making it unintelligible to anyone without the decryption key. This ensures confidentiality and security in digital communication.

Our Project involves converting a text message into Cipher text and using RSA cryptography algorithm to encrypt text generate a key for the encryption and hide the text in an image for hiding the message.

# **Introduction:**

Cryptography is the cornerstone of secure communication in the digital age, providing a means to safeguard sensitive information from unauthorized access. At its core, cryptography involves the transformation of plaintext into ciphertext through various encryption techniques. These techniques ensure that even if intercepted, the encrypted data remains indecipherable without the appropriate decryption key. From ancient civilizations to modern digital networks, cryptography has played a pivotal role in protecting secrets, facilitating secure transactions, and upholding privacy. This introduction explores the abstract principles of cryptography and its fundamental role in securing information exchange.

Encryption techniques can be categorized into two main types: symmetric encryption and asymmetric encryption. Symmetric encryption uses a single shared key for both encryption and decryption, making it efficient for large volumes of data. Examples include AES (Advanced Encryption Standard) and DES (Data Encryption Standard). Asymmetric encryption, on the other hand, utilizes a pair of keys—a public key for encryption and a private key for decryption. RSA (Rivest-Shamir-Adleman) is a widely used asymmetric encryption algorithm.

One of the most renowned encryption algorithms in modern cryptography is RSA (Rivest-Shamir-Adleman). Named after its inventors, RSA is an asymmetric encryption algorithm that utilizes a pair of keys—a public key for encryption and a private key for decryption. RSA's security is based on the computational difficulty of factoring large prime numbers, making it robust against brute-force attacks.

In this project, we even used Steganography, for the covert practice of hiding secret information within seemingly innocuous data or media, which has a rich history dating back to ancient times. In contrast to cryptography, which focuses on making messages unreadable to unauthorized users, steganography aims to conceal the existence of the message itself. This clandestine technique has evolved alongside advances in technology, finding new applications in digital communication, watermarking, and data hiding.

**Materials & Methods:**

* The materials consist primarily of encryption algorithms, key generation mechanisms, digital signatures, hash functions, secure communication protocols, and cryptographic libraries and tools.
* Encryption algorithms such as AES and RSA form the foundation of secure communication by transforming plaintext into ciphertext.
* Key generation methods ensure the creation of robust cryptographic keys, while key management protocols oversee their secure distribution and revocation.
* The process typically involves encoding the secret message into the carrier medium in such a way that it is imperceptible to casual observers. This can be achieved by subtly altering certain aspects of the carrier, such as modifying the least significant bits of pixels in an image or the least significant bits of audio samples.
* Developers leverage cryptographic libraries and tools to implement encryption techniques seamlessly within software applications, ensuring robust security measures are in place.

**1. Encryption Algorithms:** Various encryption algorithms are employed to transform plaintext into ciphertext. Common examples include symmetric encryption algorithms like AES (Advanced Encryption Standard) and asymmetric encryption algorithms like RSA (Rivest-Shamir-Adleman).

**2. Key Generation:** Encryption relies on cryptographic keys to encrypt and decrypt data. Key generation methods involve algorithms that create random or pseudo-random keys of sufficient length to ensure security.

**3. Key Management:** Proper management of encryption keys is crucial for maintaining security. This includes securely storing keys, distributing them to authorized parties, and revoking compromised keys.

**4. Digital Signatures:** Digital signatures are cryptographic techniques used to verify the authenticity and integrity of digital messages or documents. They involve using asymmetric encryption to sign and verify messages.

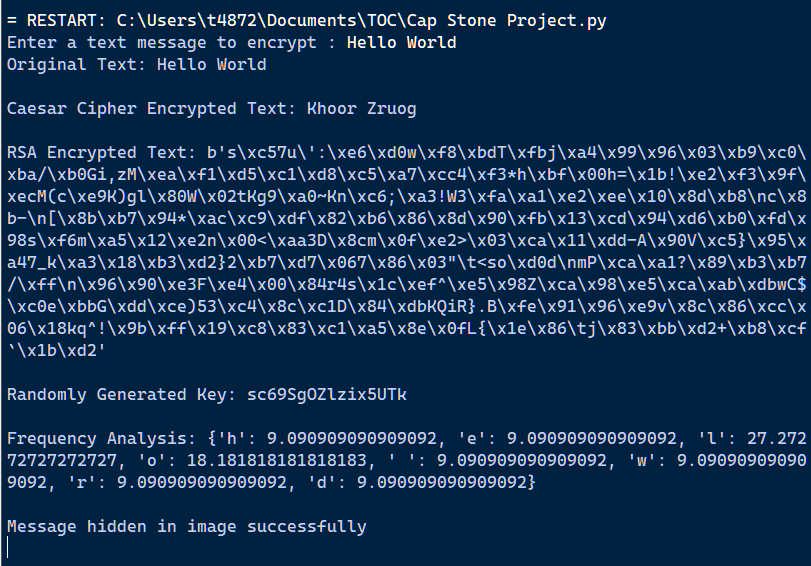
**5. Image Steganography:** Steganography is the art and science of concealing information within other non-secret data or media to ensure its confidentiality. Unlike cryptography, which focuses on making the content of a message unreadable to unauthorized users, steganography aims to hide the existence of the message itself. This is achieved by embedding the secret information within seemingly innocuous carriers such as images, audio files, video files, or even text.

**Working Code:**

| import random import string import collections from cryptography.hazmat.backends import default\_backend from cryptography.hazmat.primitives import hashes from cryptography.hazmat.primitives.asymmetric import rsa from cryptography.hazmat.primitives import serialization from cryptography.hazmat.primitives.asymmetric import padding from PIL import Image  def caesar\_cipher(text, shift):  encrypted\_text = ''  for char in text:  if char.isalpha():  shifted = chr((ord(char) - ord('a' if char.islower() else 'A') + shift) % 26 + ord('a' if char.islower() else 'A'))  encrypted\_text += shifted  else:  encrypted\_text += char  return encrypted\_text  def rsa\_encrypt(text):  private\_key = rsa.generate\_private\_key(  public\_exponent=65537,  key\_size=2048,  backend=default\_backend()  )  public\_key = private\_key.public\_key()  ciphertext = public\_key.encrypt(  text.encode(),  padding.OAEP(  mgf=padding.MGF1(algorithm=hashes.SHA256()),  algorithm=hashes.SHA256(),  label=None  )  )  return ciphertext  def generate\_random\_key(length):  return ''.join(random.choices(string.ascii\_letters + string.digits, k=length))  def calculate\_frequency(text):  frequency = collections.Counter(text.lower())  total\_characters = sum(frequency.values())  frequency\_percentage = {char: count / total\_characters \* 100 for char, count in frequency.items()}  return frequency\_percentage  def hide\_message\_in\_image(image\_path, message):  image = Image.open(image\_path)  width, height = image.size  message\_length = len(message)  if message\_length > (width \* height):  raise ValueError("Message too long to hide in image")  binary\_message = ''.join(format(ord(char), '08b') for char in message)  pixel\_index = 0  for char in binary\_message:  x = pixel\_index % width  y = pixel\_index // width  pixel = list(image.getpixel((x, y)))  pixel[-1] = (pixel[-1] & 254) | int(char)  image.putpixel((x, y), tuple(pixel))  pixel\_index += 1  image.save('hidden\_message\_image.png')  text = input("Enter a text message to encrypt : ") print("Original Text:", text)  caesar\_encrypted = caesar\_cipher(text, 3) print("\nCaesar Cipher Encrypted Text:", caesar\_encrypted)  rsa\_encrypted = rsa\_encrypt(text) print("\nRSA Encrypted Text:", rsa\_encrypted)  random\_key = generate\_random\_key(16) print("\nRandomly Generated Key:", random\_key)  frequency = calculate\_frequency(text) print("\nFrequency Analysis:", frequency)  hide\_message\_in\_image("C:/Users/t4872/Pictures/Screenshots/Screenshot (10).png", "Secret Message") print("\nMessage hidden in image successfully") |
| --- |

**Results:**

Successfully converted the message into Caesar cipher text and also RSA encrypted text. We even generated a random key for the encrypted text, Using frequency analysis and Steganography encrypted message is hidden in a copy of the selected Photo.



**Conclusion:**

In conclusion, steganography and the RSA algorithm stand as indispensable pillars in the realm of information security and cryptography. Steganography, with its covert communication techniques, provides a subtle and powerful means of hiding sensitive information within seemingly innocuous data, ensuring confidentiality and privacy in digital communication.

Meanwhile, the RSA algorithm's asymmetric encryption methodology enables secure transmission of data and authentication in digital environments. Its robust encryption and decryption processes, coupled with digital signature capabilities, ensure the integrity and authenticity of messages exchanged over public networks.

Together, these techniques play pivotal roles in safeguarding sensitive information, mitigating cybersecurity threats, and upholding privacy rights in the digital age. As technology advances and security challenges evolve, continued research and innovation in steganography, RSA, and other cryptographic techniques will be essential to staying ahead of emerging threats and preserving the security of digital ecosystems worldwide.