Image Cryptography using RSA Algorithm

Gudiwada Raghava (212is009)

Department of Computer Science and Engineering
National Institute of Technology Karnataka
Surathkal, Mangaluru, India

Abstract. In today's world, it is a crucial concern that proper encryption-decryption to send data from one location to another over the internet to prevent illegal access. Cryptography is a method of delivering data securely. The purpose of this method is to ensure one-of-a-kind information security. Image cryptography is a type of encryption that uses data hidden in an image to encrypt and decrypt the original message using a key value. Computational hardness is provided by a few algorithms, making it difficult to break a key and find the original message. To improve the security in the communication area for data transmission, the RSA technique is utilized to encrypt the image files. To transport data from one location to another, an image file is chosen for encryption and decryption utilizing a key generation process.

Keywords: Cryptography,RSA Algorithm,Key Generation,Prime Numbers,key

1 Introduction

Information security has become a major concern in the storing and transfer of data. It typically needs data security to prevent unauthorized access. Cryptography is a study of secure communications and data to prevent their contents from being revealed by eavesdropping by utilizing codes and ciphers, allowing only a small number of people to access the actual message.
web dangers become more serious, image security has become a primary focus. Medical imaging, military communication, multimedia systems, telemedicine, and other applications benefit from image encryption and decryption.

RSA is a cryptographic method that is used to provide encryption and authentication. [1]RSA was proposed in 1977. The most widely used encryption and authentication algorithm is this one. One of the first public-key cryptosystems, the RSA algorithm is frequently used to protect the data transfer. The encryption key in such a cryptosystem is made public, while the decryption key is kept private. This asymmetry is predicated in RSA on the factoring issue, which is the product of two big prime numbers. The RSA encrypted key is used to encrypt the image so that it can be converted to a ciphertext and saved as a text file. The opposite process of encryption, the reverse process, is computed using another RSA algorithm decryption key and decrypts the image from the

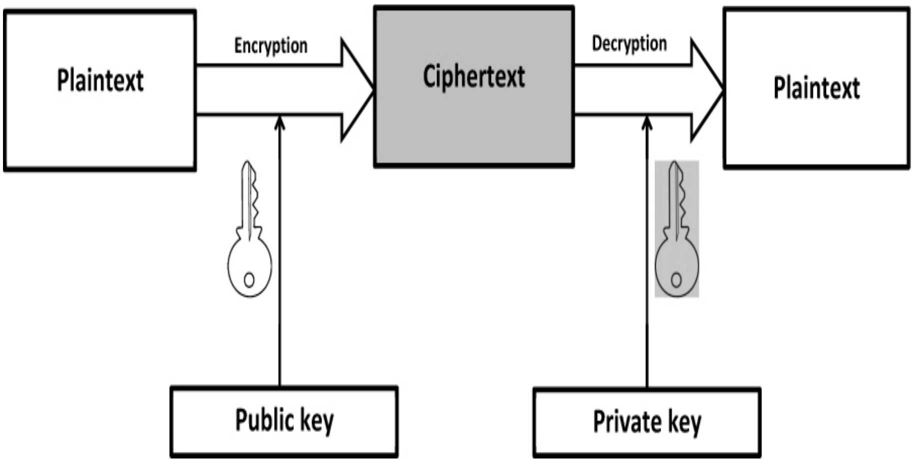


Fig. 1. Process of Encryption and Decryption

cipher text. Finally, using decryption algorithms, it will discover the generated image.

As shown in Fig 1, Encryption is the process of making a piece of information, known as the plaintext, unintelligible to everyone except those with special knowledge, commonly referred to as a key, by altering it using an algorithm, known as the cipher. The result is referred to as the ciphertext. The process of transforming ciphertext to plaintext is called decryption.

1.1 Problem Description

The RSA encrypted key is used to encrypt the image so that it can be converted to cyphertext and saved as a text file. The opposite form of encryption, the reverse process, is computed using another RSA algorithm decryption key and decrypts the image from the ciphertext.

1.2 Motivation

As the rate of communication has expanded, image security has become more vital in today's environment. All of the solutions were only able to achieve a low level of security in real-time image encryption. The RSA technique is updated for color image encryption in this study. The image is shown here. Approaches to encryption and decryption are highly secure and take less time to compute. The simulation results show that this approach is an effective image encryption cryptosystem. It's also appropriate for secure image transfer over the Internet.

1.3 Scope

When compared to the modern period, technology was used relatively less in the past. They previously employed the data decryption method, but it is now possible to use the picture encryption process, in which secret information can be stored in the image itself and decoded using a key.

1.4 Objectives

- The main objective of image encryption technique is provide privacy and security.
- Any image encryption approach aims to produce a high-quality hidden image in order to keep information private.
- Finally to get the accurate image after process of decryption using the key.

1.5 Organization of the Report

Later sections of the paper will be discussed. Here, we talk about the work related to image cryptography and next section can describing the objectives of cryptography. Later we provide a solution to the above problem and cross-checking the results. Finally to achieve the outcome should be same as input.

2 Related Work

The approach for securing data is known as encryption, according to Gunasekaran G. and Bimal Kumar Ray, et.al. [2]. The encrypted data is sent over the internet. Decryption is the process of decrypting the encrypted data using the specified algorithm. The secret information is encoded as a picture and transferred using a secret key.

[3]The propose image encryption techniques convert the original image to another image that is difficult to comprehend and keep the image confidential between users. It is critical that no one can access the information without a decryption key.

One of the most prominent and secure public-key encryption methods is the RSA) algorithm [4]. In [5] described image cryptography that can encrypt images using existing cryptosystems. However, it has two flaws. The first issue is that image sizes are always significantly larger than text sizes. As a result, the cryptosystems take a long time to encrypt the image. The second issue is that the decrypted data must match the original data.

3 The Purpose of Cryptography

The goal of cryptography is to protect data sent in the face of a potential attacker. The process of encrypting plaintext data to produce ciphertext is known as cryptographic transformation of data. A selected recipient can reverse-transform the ciphertext, allowing the original plaintext to be recovered.
Cryptography plays an important role in

1. **Authentication :**It is the process of confirming a user's or information's identity. When a user login into a computer system, user authentication is the process of verifying that user's identity.

2. **Data confidentiality :**Only authorised individuals/systems have access to sensitive or classified information, according to confidentiality. Unauthorized individuals should not have access to data exchanged across the network.

3. **Data integrity :**It refers to the general completeness, accuracy, and consistency of data in the context of networking. When delivering data via a network, data integrity must be maintained. Error checking and rectification techniques can help achieve this.

4. **Nonrepudiation :**The certainty that someone cannot deny the legitimacy of anything is known as non-repudiation. It is a legal concept that is extensively employed in the field of information security and refers to a service that gives confirmation of data's origin and integrity.

4 The Proposed Approach for IMAGE CRYPTOGRAPHY METHODOLOGY BY RSA

The RSA algorithm used for encrypting and decrypting data. Because there is a significant need for image transmission security, this work extends the RSA method to perform image encryption and decryption. It is also known as the public-key cryptosystems. RSA is a cryptographic algorithm that is commonly used for secure data transmission.

The RSA technique encrypts the original image and uses different keys to decrypt it. As shown in Fig2

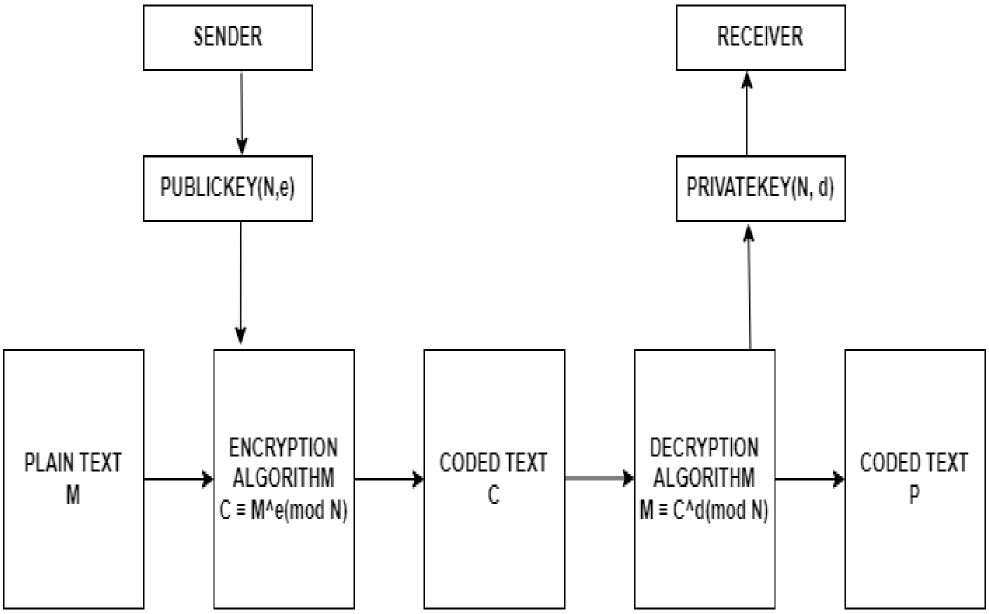


Fig. 2. RSA Diagram

RSA algorithm involves three steps

1. **Key Generation :**The first phase in the RSA algorithm is key generation. A public key and a private key are used in the RSA algorithm. The public key on those keys can be seen by anybody and is used to encrypt messages. The private key can decrypt messages and encrypted with the public key [6]. The following procedures are used to generate the keys for the RSA algorithm.
   - Choose two distinct prime numbers p and q.
   - These prime numbers p and q should be chosen at random for security [7] reasons and must have similar bit lengths. Primality testing is an efficient way to find prime integers.
   - The Miller–Rabin [8] primality test is an algorithm that determines whether a given number is likely to be prime.
   - Compute n = pq. Both the public and private keys utilise n as the modulus. Its length is measured in bits, which is the same as the key length.
   - Compute φ(n) = φ(p)φ(q) = (p 1)(q 1) , where is Euler's totient function.
   - Select an integer e such that 1 $< e <$ φ(n) and gcd(e, φ(n)) = 1; e and φ(n) must be co-prime. where e indicates the public key.
   - Determine d as $d \equiv e^{-1}$(modφ(n)) i.e.,where d indicates the multiplicative inverse of e(moduloφ(n)). simplify the above equation to get $de \equiv$ 1(modφ(n)).

2. **Encryption** [9] **:** Alice gives to Bob her public key (n, e) but keeps her private key d hidden. Bob then wishes to send Alice the message M.

$$c \equiv m^e (modn) \tag{1}$$

3. **Decryption :** Alice can retrieve m from c by computing with her private key exponent d. By reversing the padding scheme, she can recover the original message M given m.

$$m \equiv c^d (modn) \tag{2}$$

5 Experimental Results and Analysis

With the implementation of the RSA algorithm, we have come to the result that for better security of any text or image, the RSA algorithm should be used. The cryptography technique in this paper uses the RSA algorithm with public key encryption to improve the security levels of the encrypted data. One key is required to encrypt the image, while another is required to decrypt it. Finally, the image cryptography experiment demonstrates the possibility of image security in an network application scenario.

6 Code

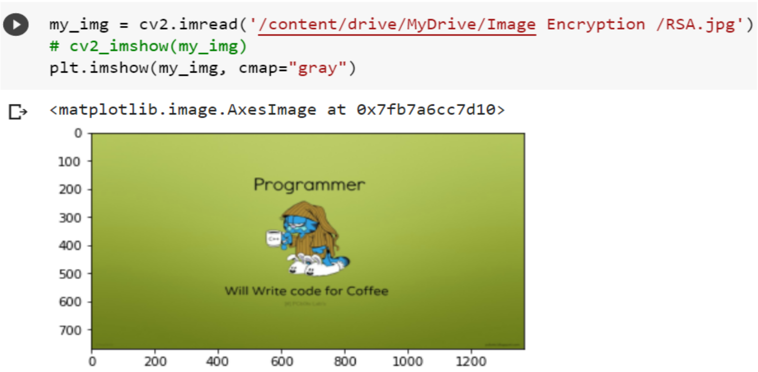
```python
# -*- coding: utf-8 -*-
"""Image Encryption RSA.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1CZOm16KQlhme1k73jZg83irbQSr_IkLk
"""

from google.colab import drive
drive.mount('/content/drive')

# Commented out IPython magic to ensure Python compatibility.
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt
# %matplotlib inline

my_img = cv2.imread('/content/drive/MyDrive/Image Encryption /RSA.jpg')
# cv2_imshow(my_img)
plt.imshow(my_img, cmap="gray")

```

```
#RSA

# STEP 1: Generate Two Large Prime Numbers (p,q) randomly
from random import randrange, getrandbits


def power(a,d,n):
  ans=1;
  while d!=0:
    if d%2==1:
      ans=((ans%n)*(a%n))%n
    a=((a%n)*(a%n))%n
    d>>=1
  return ans;


def MillerRabin(N,d):
  a = randrange(2, N - 1)
  x=power(a,d,N);
  if x==1 or x==N-1:
    return True;
  else:
    while(d!=N-1):
      x=((x%N)*(x%N))%N;
      if x==1:
        return False;
      if x==N-1:
        return True;
      d<<=1;
  return False;


def is_prime(N,K):
  if N==3 or N==2:
    return True;
  if N<=1 or N%2==0:
    return False;

  #Find d such that d*(2^r)=X-1
  d=N-1
  while d%2!=0:
    d/=2;

  for _ in range(K):
    if not MillerRabin(N,d):
      return False;
  return True;




def generate_prime_candidate(length):
  # generate random bits
  p = getrandbits(length)
  # apply a mask to set MSB and LSB to 1
  # Set MSB to 1 to make sure we have a Number of 1024 bits.
  # Set LSB to 1 to make sure we get a Odd Number.
  p |= (1 << length - 1) | 1
  return p



def generatePrimeNumber(length):
  A=4
  while not is_prime(A, 128):
        A = generate_prime_candidate(length)
  return A



length=5
P=generatePrimeNumber(length)
Q=generatePrimeNumber(length)

print(P)
print(Q)

#Step 2: Calculate N=P*Q and Euler Totient Function = (P-1)*(Q-1)
N=P*Q
eulerTotient=(P-1)*(Q-1)
print(N)
print(eulerTotient)

#Step 3: Find E such that GCD(E,eulerTotient)=1(i.e., e should be co-prime)
# such that it satisfies this condition:-  1<E<eulerTotient

def GCD(a,b):
  if a==0:
    return b;
  return GCD(b%a,a)
```

```

E=generatePrimeNumber(4)
while GCD(E,eulerTotient)!=1:
  E=generatePrimeNumber(4)
print(E)

# Step 4: Find D.
#For Finding D: It must satisfies this property:-  (D*E)Mod(eulerTotient)=1;
#Now we have two Choices
# 1.we randomly choose D &check which condition is satisfying above condition.
# 2.For Finding D we can Use Extended Euclidean Algorithm: ax+by=1
#i.e., eulerTotient(x)+E(y)=GCD(eulerTotient,e)
#Here, Best approach is to go for option 2.( Extended Euclidean Algorithm.)

def gcdExtended(E,eulerTotient):
  a1,a2,b1,b2,d1,d2=1,0,0,1,eulerTotient,E

  while d2!=1:

    # k
    k=(d1//d2)

    #a
    temp=a2
    a2=a1-(a2*k)
    a1=temp

    #b
    temp=b2
    b2=b1-(b2*k)
    b1=temp

    #d
    temp=d2
    d2=d1-(d2*k)
    d1=temp

    D=b2

  if D>eulerTotient:
    D=D%eulerTotient
  elif D<0:
    D=D+eulerTotient

  return D
```

```


D=gcdExtended(E,eulerTotient)
print(D)

row,col=my_img.shape[0],my_img.shape[1]
enc = [[0 for x in range(3000)] for y in range(3000)]

#Step 5: Encryption

for i in range(100,700):
  for j in range(100,1200):
    r,g,b=my_img[i,j]
    C1=power(r,E,N)
    C2=power(g,E,N)
    C3=power(b,E,N)
    enc[i][j]=[C1,C2,C3]
    C1=C1%256
    C2=C2%256
    C3=C3%256
    my_img[i,j]=[C1,C2,C3]


# plt.imshow(my_img, cmap="gray")
cv2_imshow(my_img)

#Step 6: Decryption
for i in range(100,700):
  for j in range(100,1200):
    r,g,b=enc[i][j]
    M1=power(r,D,N)
    M2=power(g,D,N)
    M3=power(b,D,N)
    my_img[i,j]=[M1,M2,M3]

cv2_imshow(my_img)
#plt.imshow(my_img, cmap="gray")
```

6.1 Output



```
my_img = cv2.imread('/content/drive/MyDrive/Image Encryption /RSA.jpg')
# cv2_imshow(my_img)
plt.imshow(my_img, cmap="gray")
```

```
<matplotlib.image.AxesImage at 0x7fb7a6cc7d10>
```

Fig. 3. Input Image

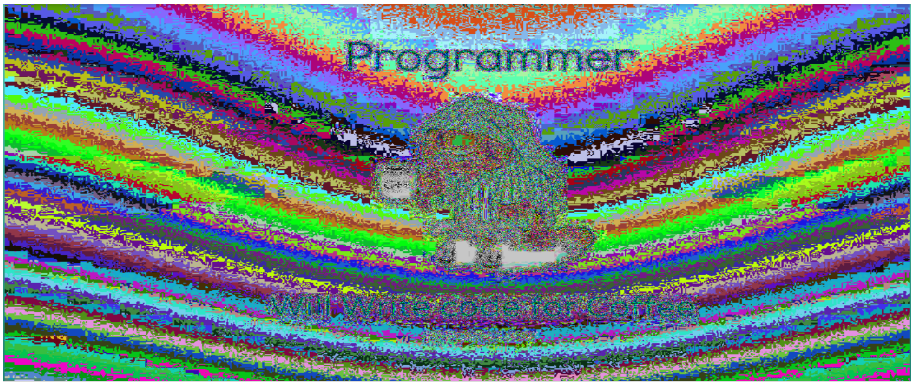


Fig. 4. Encrypted Image

Fig. 5. FinalOuput

References

1. Rivest, R.L., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, Vol 21, No. 2, February 1978, p. 120-26.
2. Gunasekaran G. and Bimal Kumar Ray, Encrypting And Decrypting Image Using Computer Visualization Techniques, Journal of Engineering and Applied Sciences VOL. 9, NO. 5, ISSN 1819-6608, MAY 2014.
3. Komal D Patel , Sonal Belani, Image Encryption Using Different Techniques: A Review International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 1, Issue 1, November 2011.
4. Rivest, R.L., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, Vol 21, No. 2, February 1978, p. 120-26.
5. Hiral Rathod, Mahendra Singh Sisodia, Sanjay Kumar Sharma, Design and Implementation of Image Encryption Algorithm by using Block Based Symmetric Transformation Algorithm (Hyper Image Encryption Algorithm), ISSN 2249-6343 International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 1, Issue 3.
6. Vishwagupta, Gajendra Singh ,Ravindra Gupta,“ Advance cryptography algorithm for improving data security", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 1, January 2012 ISSN: 2277 128X.
7. K.Sony , Desowja Shaik, B.Divya Sri , G.Anitha," Improvised Asymmetric Key Encryption Algorithm Using MATLAB", IOSR Journal of Electronics and Communication Engineering (IOSR-JECE),Volume 10,Issue 2, (Mar - Apr.2015), e-ISSN: 2278-2834,p- ISSN: 2278-8735
8. Rabin, Michael O. (1980), "Probabilistic algorithm for testing primality", Journal of Number Theory, 12 (1): 128–138, doi:10.1016/0022-314X(80)90084-0
9. K.Sony , Desowja Shaik, B.Divya Sri , G.Anitha," Improvised Asymmetric Key Encryption Algorithm Using MATLAB", IOSR Journal of Electronics and Communication Engineering (IOSR-JECE),Volume 10,Issue 2, (Mar - Apr.2015), e-ISSN: 2278-2834,p- ISSN: 2278-8735.