RESEARCH ARTICLE OPEN ACCESS

Data Hiding and Retrival Using Advanced Encryption and Decryption Algorithms

Mamtha Shetty, Shreedhar. A. Joshi (P.Hd)
PG Scholar Dept E & CE S.D.M. College of Engineering Engineering & Technology, Dharwad-02
Assistant Professor Dept E & CE S.D.M. College of & Technology, Dharwad-02

Abstract
In this era of digital world, with the evolution of technology, there is an essential need for optimization of online digital data and information. Nowadays, Security and Authenticity of digital data has become a big challenge. This paper proposes an innovative method to authenticate the digital documents. A new method is introduced here, which allows multiple encryption and decryption of digital data.

I. INTRODUCTION

Quick Response (QR) code is a type of 2 dimensional matrix barcode, which gained popularity because of its large capacity to hold digital data and it can be integrated in any mobile devices. It can be applied to encrypt data in defense system, banking sector, mobile network.QR code was invented by DENSO CORPORATION in 1994. Approved as AIMI Standard in 1997 and ISO/IEC Standard in 2000. Adapted as an industry-wide standard code by AIAG, JAMA and JTA. High readability by a reader is pursued.QR code is a barcode which is readable by any camera-enabled smart phone. They are typically seen as a white square with black geometric shapes.Users point their phones at the QR code, scan it, and are then taken to the end data. This could be text, a website, a YouTube video, a podcast…

II. Methods

We use TTJSA [1] encryption algorithm, which was designed by Nath et al. and is an amalgamation of three different cryptographic modules: generalized modified Vernam cipher [1], MSA [2] and NJJSA [3], for the encryption purpose of data in the QR Code. After encrypting the data, we embed the data in the QR Code using a set of different protocols and ultimately generate the encrypted QR Code.

Till now, only few articles in the concerned area are published. The proposed analysis will choose two of the algorithms namely DJSA and NJJSAA [1],[2]. By going through this work [1],[2], they propose the key generation and almost the same process for the encryption as well as a contrary multiple encryption using bit exchange, right shift and XOR operation makes the system of NJJSAA differ from DJSA. Both [1],[2] results in large mathematical calculations and CPU processing. This leads to unnecessary encryption time consumption.[3],[4] also gives the insight about different proportions of power consumption. TTJSA [1] is a combined symmetric key cryptographic method, which is formed of generalized modified Vernam cipher, MSA and NJJSA symmetric key cryptographic methods.

a) Modified Vernam Cipher

In this step, we break the whole file into different small blocks (like in Block Cipher system []), where each block size should be less than or equal to 256 bytes. Then we follow these steps:

Step1: Perform normal Vernam Cipher method with the block of randomized key i.e. each byte of blocks of the file + each byte of the blocks of randomized key.

Step 2: If the pointer reaches the end of each block then after performing Vernam Cipher method, pass the remainder of the addition of the last byte of the file block with the last byte of the key to the next file block and add the remainder with the first byte of the that file block. (This mechanism is called feedback mechanism)

Step 3: Perform Step 1 and Step 2 until the whole file is encrypted and repeat this step for random number of times. After performing the aforementioned steps, we again merge the blocks of the encrypted file and thus we get the final encrypted result of this modified Vernam Cipher method.

b) NJJSAA Algorithm

The encryption number (=secure) and randomization number (=times) is calculated according to the method mentioned in MSA algorithm [2].
Step 1: Read 32 bytes at a time from the input file.

Step 2: Convert 32 bytes into 256 bits and store in some 1- dimensional array.

Step 3: Choose the first bit from the bit stream and also the corresponding number(n) from the key matrix. Interchange the 1st bit and the n-th bit of the bit stream.

Step 4: Repeat step-3 for 2nd bit, 3rd bit...256-th bit of the bit stream

Step 5: Perform right shift by one bit.

Step 6: Perform bit(1) XOR bit(2), bit(3) XOR bit(4),...,bit(255) XOR bit(256)

Step 7: Repeat Step 5 with 2 bit right, 3 bit right,...,n bit right shift followed by Step 6 after each completion of right bit shift.

III. Generation of QR Code

To create a QR code [9][10][11] is we first create a string of data bits. This string includes the characters of the original message (encrypted message in this case) that you are encoding, as well as some information bits that will tell a QR decoder what type of QR Code it is.

After generating the aforementioned string of bits, we use it to generate the error correction code words for the QR Code. QR Codes use Reed-Solomon Error Correction technique [10][12].

IV. Algorithms

I) Algorithm of TTJSA (Encryption):

```
Step 1: Start
Step 2: Initialize the matrix mat[16][16] with numbers 0 to 255 in row major wise.
Step 3: call keygen() to calculate randomization number (=times), encryption number (=secure).
Step 4: call randomization() function to randomize the contents of mat[16][16].
Step 5: times2=times
Step 6: copy file f1 into file2
Step 7: k=1
Step 8: if k>secure go to Step 15
Step 9: p=k%6
Step 10: if p=0 then
 Callvernamenc(file2,outf1) times=times2
callnjjsaa(outf1,outf2)
callmsa_encryption(outf2,file1)
elseifp=1then
call vernamenc(file2,outf1)
times=times2
callmsa_encryption(outf1,file1)
call file_rev(file1,outf1)
callnjjsaa(outf1,file2)
callmsa_encryption(file2,outf1)
call vernamenc(outf1,file1)
times=times2
else if p=2 then
callmsa_encryption(file2,outf1)
call vernamenc(outf1,outf2)
set
times=times2
callnjjsaa(outf2,file1)
else if p=3 then
callmsa_encryption(file2,outf1)
call njjsaa(outf1,outf2)
call vernamenc(outf2,file1)
times=times2
 else if p=4 then
call njjsaa(file2,outf1)
call vernamenc(outf1,outf2)
 times=times2
call msa_encryption(outf2,file1)
 else if p=5 then
 callnjjsaa(file2,outf1)
callmsa_encryption(outf1,outf2)
call vernamenc(outf2,file1)
 times=times2
Step 11: call function file_rev(file1,outf1)
Step 12: copy file outf1 into file2
Step 13: k=k+1
Step 14: goto Step 8
Step 15: End
```

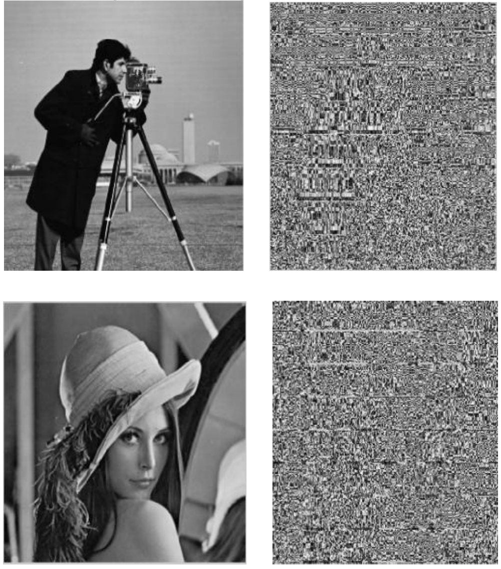
*II)*Algorithm of vernamenc(f1,f2):

```
Step 1: Start vernamenc() function
Step 2: The matrix mat[16][16] is initialized with numbers 0-255 in row major wise order.
Step 3: call function randomization() to randomize the contents of mat[16][16].
Step 4: Copy the elements of random matrix mat[16][16] into key[256] (row major wise)
Step 5: pass=1, times3=1, ch1=0
Step 6: Read a block from the input file f1 where number of characters in the block 256 characters
Step 7: If block size < 256 then goto Step 15
Step 8: copy all the characters of the block into an array str[256]
 Step 9: call function encryption where str[] is passed as parameter along with the size of the current block
Step 10: if pass=1 then
 times=(times+times3*11)%64
pass=pass+1
else if pass=2 then
times=(times+times3*3)%64
pass=pass+1
else if pass=3 then
times=(times+times3*7)%64
pass=pass+1
else if pass=4 then
times=(times+times3*13)%64
pass=pass+1
else if pass=5 then
```

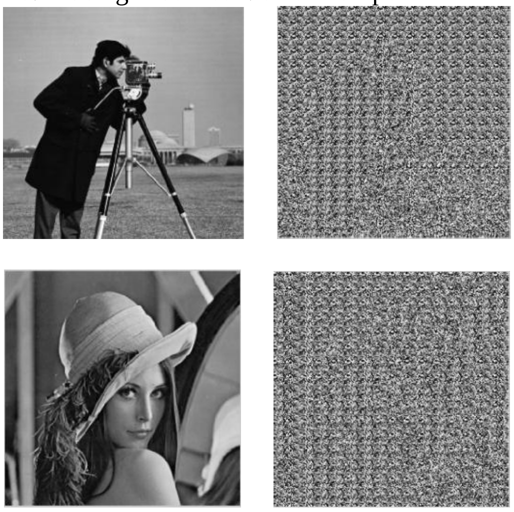
```
times=(times+times3*times3)%64
pass=pass+1
elseifpass=6then
times=(times+times3*times3*times3)%64
pass=1
Step 11: call function randomization() with current value of times
Step 12: copy the elements of mat[16][16] into key[256]
Step 13: read the next block
Step 14: goto Step 7
Step 15: copy the last block (residual character if any) into str[]
Step 16: call function encryption() using str[] and the no. of residual characters
Step 17: Return
```

V. Results

a. Using NJJSA Algorithm



b. Using Modified Vernam Chiper



VI. Tables

Formats	Mean Square Error	PSNR
.tif	9496.56	8.36
.jpg	2262.03	14.62
.png	1083.96	17.81

Table for NJJSA method

Formats	Mean Square Error	PSNR
.tif	9442.49	8.41
.jpg	2250.22	14.64
.png	1079.32	17.83

Table for Vernam Chiper method

VII. Applications and Advantages

1. Advantages of all 2D symbols are integrated in the QR code.
2. Large data capacity
3. High density
4. High-speed reading
5. 360-degree reading
6. Error correction capability
7. Special characters *and alphanumeric are supported.*

VIII. Conclusion and Future Scope

In the present work, it is mainly focused on confidential encrypted data hiding in QR code.

A smart phone running on Android or iOS or any other new generation of mobile OS, can be used to extract the encrypted data from embedded QR-code and finally that data to be decrypted using the TTJSA decryption algorithm.

References

[1] Symmetric key cryptosystem using combined cryptographic algorithms - Generalized modified Vernam Cipher method, MSA method and NJJSAA method: TTJSA algorithm " Proceedings of Information and Communication Technologies (WICT), 2011 " held at Mumbai, 11th – 14th Dec, 2011, Pages:1175-1180.

[2] Symmetric Key Cryptography using Random Key generator: Asoke Nath, Saima Ghosh, Meheboob Alam Mallik: "Proceedings of International conference on security and management(SAM'10" held at Las Vegas, USA Jull 12-15, 2010), P-Vol-2, 239-244(2010).

[3] New Symmetric key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJJSAA symmetric key algorithm: Neeraj

Khanna,Joel James,Joyshree Nath, Sayantan Chakraborty, Amlan Chakrabarti and Asoke Nath : Proceedings of IEEE CSNT-2011 held at SMVDU(Jammu) 03-06 June 2011, Page 125-130(2011).

[4] Somdip Dey, Joyshree Nath, Asoke Nath, "An Integrated Symmetric Key Cryptographic Method – Amalgamation of TTJSA Algorithm, Advanced Caesar Cipher Algorithm, Bit Rotation and Reversal Method: SJA Algorithm", IJMECS, vol.4, no.5, pp.1-9, 2012.

[5] Somdip Dey, Joyshree Nath and Asoke Nath. Article: An Advanced Combined Symmetric Key Cryptographic Method using Bit Manipulation, Bit Reversal, Modified Caesar Cipher (SD-REE), DJSA method, TTJSA method: SJA-I Algorithm. *International Journal of Computer Applications46(20):* 46-53, May 2012. Published by Foundation of Computer Science, New York, USA.

[6] Somdip Dey, "SD-EQR: A New Technique To Use QR CodesTM in Cryptography", Proceedings of "1st International Conference on Emerging Trends in Computer and Information Technology (ICETCIT 2012)", Coimbatore, India, pp. 11-21.

[7] Cryptography and Network Security, William Stallings, Prentice Hall of India.

[8] Cryptography & Network Security, Behrouz A. Forouzan, Tata McGraw Hill Book Company.

[9] "QR Code, Wikipedia", http://en.wikipedia.org/wiki/QR_code [Online] [Retrieved 2012-02-09]