SBD Encryption Algorithm

Kanaad Deshpande
Department of Computer Engineering
D.J Sanghavi College of Engineering
Mumbai, India
26kanaad@gmail.com

Junaid Girkar
Department of Computer Engineering
D.J Sanghavi College of Engineering
Mumbai, India
junaidgirkar@gmail.com

Dr. Ramchandra Mangrulkar
Department of Computer Engineering
D.J Sanghvi College of Engineering
Mumbai, India
ramchandra.mangrulkar@djsce.ac.in

***Abstract*—This document is a research paper on a new type of data encryption algorithm which makes use of a Sudoku, an everyday object, as its encryption key. It can encryption multiple types of data be it a file, an image or simple plaintext. The encryption process involves multiple rounds of block cipher encryption and transposition cipher encryption based on a randomly generated Sudoku resulting in the data being encrypted better than some of the other commonly used encryption techniques.**

***Index Terms*—encryption, cipher, block cipher, transposition cipher, image encryption, Sudoku**

I. Introduction

With the rise in data being shared over the internet, data security has been gaining more importance day by day and new algorithms are being developed daily to ensure the data is being sent securely. This paper introduces a new way of encrypting multiple types of data files using an everyday objects [**?**] as the key.

II. Proposed Methodology

A. General Process

The encryption algorithm is based on a set of recurring patterns in everyday objects. One of the most popular such item is the 9 x 9 Sudoku puzzle that is available in almost all newspapers. It has a fixed pattern sequence where all the numbers from 0 - 9 must be there in each row, in each column and in each sub-blocks of 3 x 3. This data encryption algorithm is a combination of block cipher [**?**] and transposition ciphers [4] where the data goes through multiple rounds of encryption.

B. Image Encrytion

For image encryption [5], the algorithm takes a RGB color image [6] which it converts into a series of 9 x 9 matrices. For each matrix the algorithm then generates a random Sudoku using the Sudokugen python library. Using the rows and columns of the randomly generated Sudoku as the transposition key, the rows and columns of the image matrix are transpositioned multiple times to ensure a higher level of encryption.

C. Text Encryption

For text encryption, similar to the image encryption, the plain text is converted into matrices of 9 x 9. Then using the Sudokugen library, a random transposition key is generated. Looping through the key, row transposition and column transposition is applied multiple times to generate the encrypted cipher text.

D. Block Diagram

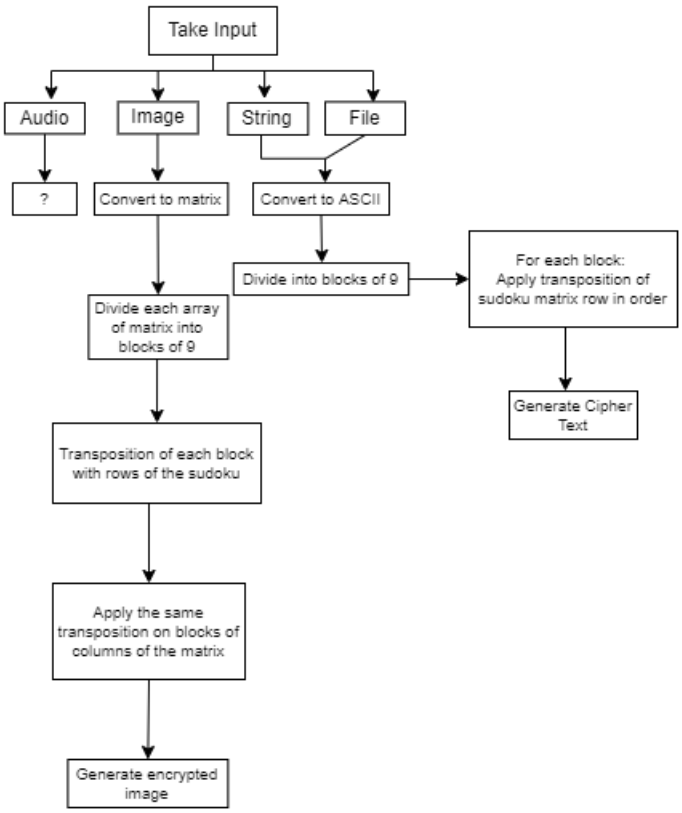

Figure 1. Block Diagram.

E. Algorithm

Algorithm 1 Add padding around image

procedure SBDPADDING($image, paddingValue$)
  if $height < 81$ **then**
    add padding 81 - height
  else if $width < 81$ **then**
    add padding 81 - width
  else if $heightmod81! = 0$ **then**
    add padding (height//81+1)*81 - height
  else if $widthmod81! = 0$ **then**
    add padding (width//81+1)*81 - width
=0

Algorithm 2 Encrypt image

procedure SBDPADDING($image, key$)
  Generate random Sudoku.
  Solve the Sudoku to generate the key.
  For each row in Sudoku:
  Generate random Sudoku.
  Solve the Sudoku to generate the key
for $eachrowinSudoku$ **do**
  for $eachblockof9bits$ **do**
    apply permutation to each block of 9 bits for each row
  end for
end for
for $eachcolumninSudoku$ **do**
  for $each9-bitblockofcolumnofimage$ **do**
    apply permutation to each column of image for each column of Sudoku
  end for
end for
=0

III. Experimentation

It has been observed that any traditional art form including but not limited to various objects, games or mathematical models that creates a pattern can be used for encipherment. Further study shows that a Sudoku based approach of encipherment, involving pixel scattering encrypted the original data beyond recognition in the image after several iterations of applying the encryption algorithm. The number of iterations can be decided based on how much scattering is required for each image. Having a flexible number of iterations makes the algorithm computationally heavy, yet secure. Experimentation with various Sudokus followed. This involved using different keys for each iteration based on the 6 quintillion options available.

IV. Comparison with other algorithms

A. Comparison with S-DES

The S-DES algorithm [7] doesn't work well with symmetric images. This is because the algorithm is applied uniformly to every single pixel. The output generated hence contains pixels uniformly scattered in the plane. Sudoku-based encryption offers a flexible number of iterations so we loop through until a completely unrecognizable image is formed.

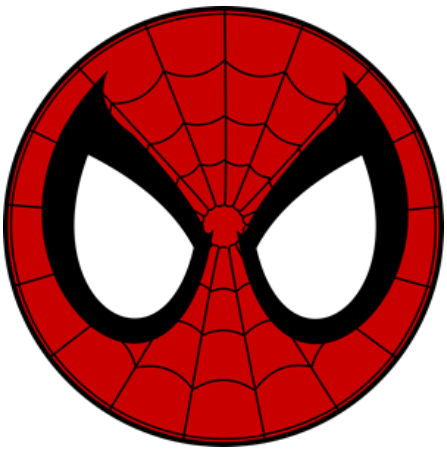

Figure 2. Original Image

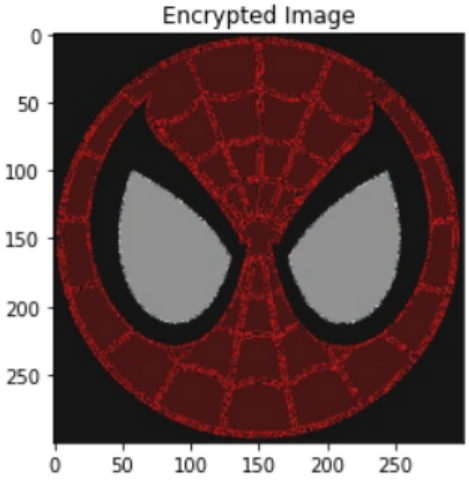

Figure 3. S-DES Encryption of Symmetric Image

$$a + b = \gamma \tag{1}$$

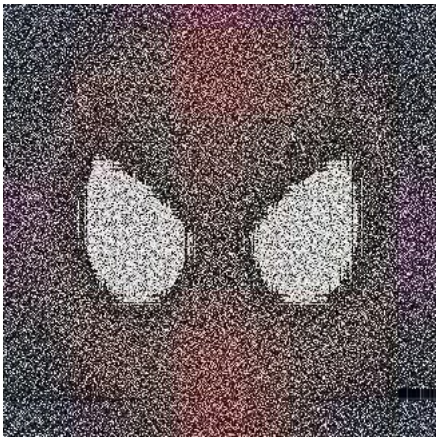

Figure 4. Sudoku-Based Encryption of Symmetric Image After 1000 Iterations

V. Attacking the Algorithm

A. Brute Force Attack

Considering a brute force attack on the algorithm, the attacker first has to find the randomly generated 9 x 9 Sudoku key. This has a possibility of 1 in 6,670,903,752,021,072,936,960 cases. Even with a powerful device, it would take an extremely long time to try all the possible combinations in order to find the correct one as they will have to run the algorithm through a high number of randomly generated layers of decryption whose count will need to be separately brute forced.

Even if we consider the one in sextillion chance that an attacker manages to guess the randomly generated key using the brute force algorithm, the attacker would gain access to the first 324 bits of data only. For the remaining data, the attacker has to again brute force for the other randomly generated 9 x 9 key.

So brute forcing this algorithm has an infinitesimally chance of success and the return per success is also very less.

B. Results

The algorithm was tested on several iterations - 10, 50, 100, 250, 400, 500, 750, 1000 and the following results were obtained:

Table I
Comparison of Iterations

Number of iterations	Time Required
10	2.658229500000001
50	11.760182800000003
100	25.154778399999998
250	62.587474000000014
400	105.29135510000003
500	131.0632736
750	217.96522460000006
1000	287.7479881999998

It can be observed that the algorithm takes nearly nearly 5 minutes to complete a thousand iterations, which shows its computational inefficiency, which is a trade-off when it comes to security.

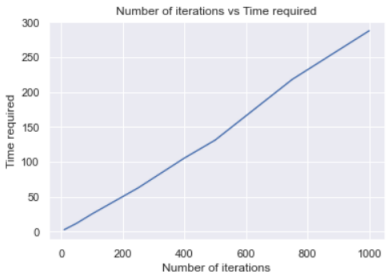

Figure 5. Time Analysis

Figure 6. Image with padding after first iteration

Figure 7. Image after 10 iterations

Figure 8. Image after 100 iterations

Figure 9. Image after 250 iterations

Figure 10. Image after 400 iterations

Figure 11. Image after 500 iterations

Figure 12. Image after 750 iterations

Figure 13. Image after 1000 iterations

ACKNOWLEDGMENT

We would like to express our gratitude to Department of Computer Engineering at Dwarkadas J Sanghvi College of Engineering who motivated us to dive into research and guided us when we faced any difficulty.

REFERENCES

[1] Mousavi, Maryam, and Babak Sadeghiyan. "A new image encryption scheme with Feistel like structure using chaotic S-box and Rubik cube based P-box." Multimedia Tools and Applications 80.9 (2021): 13157-13177.

[2] Wang, Xu, et al. "Multi-level reversible data hiding for crypto-imagery via a block-wise substitution-transposition cipher." Journal of Information Security and Applications 64 (2022): 103067.

[3] José A.P. Artiles, Daniel P.B. Chaves, and Cecilio Pimentel. 2019. Image encryption using block cipher and chaotic sequences. ¡i¿Image Commun.¡/i¿ 79, C (Nov 2019), 24–31. https://doi.org/10.1016/j.image.2019.08.014

[4] Djamalilleil, As'ad Muslim, Muslim Salim, Yulita Alwi, Erick Azis, Huzain Herman,. (2018). Modified Transposition Cipher Algorithm for Images Encryption. 1-4. 10.1109/EIConCIT.2018.8878326.

[5] Zia, U., McCartney, M., Scotney, B. et al. Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains. Int. J. Inf. Secur. (2022). https://doi.org/10.1007/s10207-022-00588-5

[6] Al-Roithy, Budoor Obid, and Adnan Gutub. "Remodeling randomness prioritization to boost-up security of RGB image encryption." Multimedia Tools and Applications 80.18 (2021): 28521-28581.

[7] Kumar, Sanjay, and Sandeep Srivastava. "Image encryption using simplified data encryption standard (S-DES)." International Journal of Computer Applications 104.2 (2014).