Binary Encryption based on a Rubik's cube

Tejeswini Sundaram
Dept. of Computer Science & Engineering,
Manipal Institute of Technology,
Manipal – 576104, India
tejeswinisundaram@live.com

Vyom Chabbra
Dept. of Computer Science & Engineering,
Manipal Institute of Technology,
Manipal – 576104, India.
vyomchabbra@live.com

***Abstract*— This Paper presents a chaos based binary encryption scheme based on the Rubik's cube principle, wherein an arbitrary length key provided by the user is used to encrypt the binary data. The challenge of not knowing the key length provides a stronger base for this algorithm, as the key specific attacks will involve more permutations. In order to provide a high order of scattering we have used half the encrypted bits of the previous cycle in the encryption of the next cycle. To achieve robustness the bit size to be used can also be arbitrary as we can base it on a different sized cube. This way we have multiple levels of scattering the data. In order to prevent the possibility of the reordering of cipher text to get plain text we have made use of XORing. To achieve pseudo randomness in the cipher text we have repeated the scattering process with multiple keys. The performance of this algorithm is discussed against common attacks like brute force, cipher text and plain text attacks and analysis shows that such a high degree of randomness positively slows the attack by a large time interval.**

Keywords*: *Rubik's Cube, Chaotic, Binary Encryption, Brute Force Attack, Pseudo Randomness

I. INTRODUCTION

Rubik's Cube is a 3-D combination puzzle invented in 1974[1] by Hungarian sculptor and professor of architecture Ernő Rubik. Originally called the Magic Cube, the cube is the world's top selling puzzle game and considered as the world's bestselling toy. Rubik's Cube is well-known to have a rich underlying mathematical structure (group theory) [2] and also has a rich underlying algorithmic structure [4]. The Rubik's Cube has been used as a shining example of group theory.

Mathematically, the Rubik's Cube is a permutation group. In a classic Rubik's Cube, each of the six faces is covered by nine stickers, among six solid colors (traditionally white, red, blue, orange, green, and yellow). Every face could be rotated clockwise or counter clockwise. The original (3×3×3) Rubik's Cube has eight vertexes and twelve edges. There are 8! (40,320) ways to arrange the corner cubies. Seven can be oriented independently, and the orientation of the eighth depends on the preceding seven, giving 37 (2,187) possibilities. There are 12! /2 (239,500,800) ways to arrange the edges, since an odd permutation of the corners implies an odd permutation of the edges as well. Eleven edges can be flipped independently, with the flip of the twelfth depending on the preceding ones, giving 211 (2,048) possibilities [6]. In total, there are approximately 4.3×1019 permutations. Thus, the large number of permutations leads to a high complexity of the positions on Rubik's Cube. For a 3 × 3 × 3 magic cube, if we suppose one step indicates a 90 degree rotation, then fifteen parts or faces can be rotated as the first step. Therefore 30 different rotating methods can be applied to rotate the cube clockwise or counterclockwise as one step.

In cryptography, encryption is the process of encoding messages or information in such a way that only authorized parties can read it. Encryption does not of itself prevent interception, but denies the message content to the interceptor In an encryption scheme, the message or information, referred to as plaintext, is encrypted using an encryption algorithm and a key, generating cipher text that can only be read if decrypted using the same key. For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. It is in principle possible to decrypt the message without possessing the key, but, for a well-designed encryption scheme, large computational resources and skill are required. An authorized recipient can easily decrypt the message with the key, provided by the originator to recipients but not to unauthorized interceptors.

In the technique described in this paper, we describe a chaos based symmetric encryption technique based on the principles of a Rubik's cube. In symmetric-key schemes, [8] the encryption and decryption keys are the same. Thus communicating parties must have the same key before they can achieve secret communication. Unlike the conventional cryptographic algorithm mainly based on discrete mathematics, chaos based cryptography is relied on the dynamics of nonlinear systems. The chaotic encryption methods are highly sensitive to initial conditions and highly deterministic and highly unpredictable and perceptual random behaviors. All these properties make chaos based encryption algorithm a good and attractive option for cryptography. By this encryption technique we aim to make it computational very hard and difficult for interceptors to decrypt the message being passed.

The property of scrambling the data based on different rotation of the Rubik's cube has inspired our study of the encryption of binary data using Rubik's Cube. In this paper we will be describing how a plain text can be converted to a cipher text using a key of any length, as required by the user. The

study is based on the scrambling nature of the cube which also allows for the re-arrangement of the scrambled data by using specific rotations and moves. The details of the methodology used has been discussed in Section 4 of this paper.

In this paper we have considered an 8x8x8 Rubik's cube for the study. The cube has 6 different faces and each face has 8x8 elements .The cube can be considered as an ordered list which has 384 elements in total as shown in Fig 1. By rotating the 6 faces of the cube we can define 6 basic operations or permutations which rearrange the ordered list in a certain way. Repeating and combining these permutations we can define new permutations, which rearrange the list in another way. In the methodology we have described below, the 6 faces of the cube are of more relevance than the color of each face of the cube. See [3] for how to compute the number of elements in the group generated by the basic Rubik's Cube moves (or any set of permutations) in polynomial time.

In this paper, we introduce a novel binary data encryption Method using the Rubik's Cube Scrambling Technique. It utilized the key and the various rotation techniques on the cube to generate a random-like binary output as the cipher text. This chaos-based symmetric encryption method, is tested and we demonstrate its robustness and effectiveness for generating uniform-like encrypted data. The presented cipher can be used to encrypt data in the form of text input.

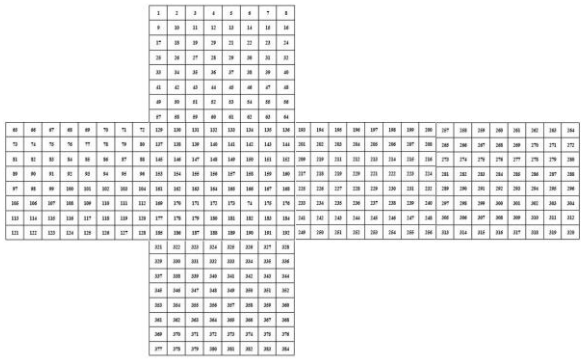

Fig 1. An 8X8X8 Rubik's Cube with 384 elements.

The rest of the paper is organized as follows: in Section II, we discuss the prior research that have been conducted on the Rubik's cube; in Section III, the novel methodology of the binary encryption based on Rubik's cube is discussed; in Section IV the experimental results and the security analysis have been made; and in Section V, the conclusion remarks are drawn.

II. RELATED WORK

The previous studies conducted on the usage of Rubik's cube techniques and principle for encryption purpose focuses on its application in the field of image cryptography.

In [7] paper improves an image scrambling algorithm based on Rubik's cube rotation and logistic sequence. The improved algorithm resizes original image and partitions a resized image into six blocks to generate cubes. Rotate these cubes in 25 steps by using 30 different rotating methods which are controlled by a chaotic system. After rotating, remap those cubes to a chaotic image. Then use Chebyshev chaotic system to generate the rotating methods and scramble the chaotic image again in order to get better result. As the rotation methods are totally generated by the chaotic system, the encrypted image cannot be retrieval unless the arguments of the chaotic system are exactly the same. Compared with an algorithm based on Arnold transform and the algorithm based on Rubik's cube rotation and logistic sequence, this improved algorithm has better scrambling degree and less correlation between adjacent pixels. This algorithm has a relatively large key space and very sensitive to the keys. Through analysis the statistical characteristics of the original image, the [2] algorithm and the improved algorithm, it could be observed that the improved algorithm has better scrambling degree. For adjacent pixels, the encrypted image has less correlation coefficients.

In [5] the paper proposes an efficient image cryptosystem based on permutation and diffusion operations in order to enhance the protection of iris-based systems against replay attacks. These attacks have been identified and located in different points by Ratha et al. [6]. First, the original image is partitioned into blocks, which are then permuted using a permutation key. After that, the Rubik's cube principle is applied to each block in order to obtain a scrambled image. Finally, the pixels values of rows and columns of the scrambled image are changed using XOR operator to generate the encrypted image. Experimental tests and security analysis have been carried out on iris images, chosen from CASIA database. The obtained results clearly show the robustness of the proposed image cryptosystem against common attacks, namely exhaustive, differential and statistical attacks and also reveal the high security level achieved by the proposed algorithm.

In [9] the authors proposed a block cipher based on the Sudoku Matrix. This technique is also chaos based and shows randomness. The cipher encrypts binary data to random like data, both the pixels' values and positions are changed according to the Sudoku matrix. The offered block cipher combines many advantages of chaos-based encryption and traditional transform-based encryption techniques. The technique is as follows. Initially the Sudoku matrix S is calculated as described in detail in [9]. Although the size of resulting Sudoku Matrix S might be different, one can always pad the Sudoku Matrix S to a desired size. The encryption and decryption process has been shown in the figure. The encryption process involves the plaintext block to be sent to two main components: the Sudoku Scrambling Component and the Sudoku XOR component. The cipher text block is obtained after the process. The decryption process is the exact mirror of the encryption process. Computer simulations show that a) the encrypted data have very random-like properties under many statistical metrics, b) unlike most chaos-based encryption methods generating unpredictable output, our new method is robust and effective for generating uniform-like encrypted

data; c) it has high sensitivity to the key. The paper says that the offered scheme can be applied to many different data types, such as audio, image and video.

III. METHODOLOGY

The proposed methodology is as described in the diagram shown below [Fig 2]. The encryption process is described in this figure. The method is a symmetric encryption technique. Hence, the decryption process is the reverse of the encryption process and it uses the same secret key to decrypt. Initially, the Plain Text is obtained from the user. The text obtained is then converted to binary digits, i.e. 0 or 1. The Binary Values are then populated on the data structure depicting the Rubik's Cube Model. Once this phase is completed, the values are passed into two components which are responsible for the encryption process and the generation of the cipher text. The two components are: (1) Shuffling and (2) Vignere Cipher operation. The Secret key used in this technique is used to determine the scrambling. The details of these components have been described in the sub-sections that follow. The resulting output obtained is a random incomprehensible value called the cipher text. The final key generated will be of the form ni1i2i3i4i5....ik, where 'n' is side length of the cube and 'k' is the length of the secret key. This cipher text is then passed to the recipient, who on receiving the cipher text performs the exact reverse of the above process with the secret key value to obtain the plain text.

A. Shuffling the Rubik's Cube

The Shuffling algorithm is used to scramble the data on the Rubik's cube by rotating the cube according to the secret key value imputed from the user.

Predefined Data:
n = side length of Rubik's Cube Model
m = no. of bits accepted in model = n^3
k = key length = no. of operations done on the Rubik's Cube Model
p = no. of bits left to work on in the Plain Text.

Step 1: Fill Rubik's Cube Model with m bits from p, If p<m then remaining bits are taken as 0.
Step 2: Apply k side rotation operations on the Rubik's Cube Model. The i^{th} value of Key defines the type of operation to be done on the model. The Total possible operations on a cube of size 'n' is 8n, and the i^{th} value of the key is converted to ASCII format and then K_i%8n defines the operation type to be performed.
Step 3: Convert Secret key into binary format and use it to apply Vignere cipher on the scrambled model values.
Step 4: Decrease p with m/2 from the front, if p>0 then go to step 1.

B. Vignere Cipher

The format we have used in this paper is that both the plain text and the secret key are converted into binary format initially, and then the Vignere cipher is applied by XORing the values of the plain text and the cipher text.
For example: Consider a plain text 'paper' and a secret key 'joke'. Binary value of 'paper' is 01110000 01100001 01110000 01100101 01110010 and the Binary value for 'joke' is 01101010 01101111 01101011 01100101. The Vignere cipher is as follows:

```
01110000 01100001 01110000 01100101 01110010
01101010 01101111 01101011 01100101 01101010
------------------------------------------------------------------
01111010 01101111 01111011 01100101 01111010
------------------------------------------------------------------
```

Which gives the final cipher text as: zo{ez

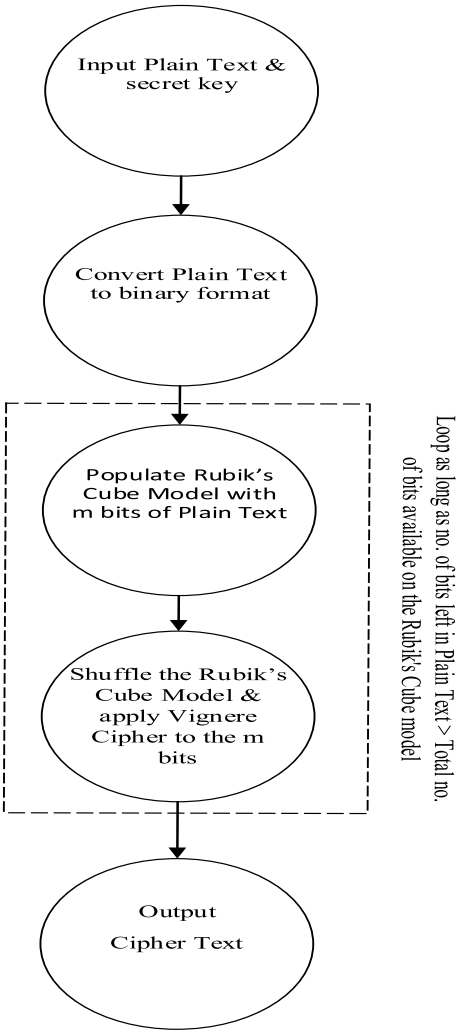

Fig 2. Proposed Method for Binary Encryption using Rubik's Cube Technique.

IV. EXPERIMENTAL RESULTS

The speed of the algorithm depends on the key length and the maximum possible bit changes at one time which is found to be n^2 + 4n where 'n' refers to the side length of the Rubik's Cube. For a key length 'k', the max bit operations per shuffle is k (n^2 + 4n) and n^3 for the vignere cipher. Therefore the total bit operations performed is given by k (n^2 + 4n) + n^3. This is then repeated j = 2p/n^3 – 1 times, where 'p' refers to the no. of bits in the plain text. Here, if p modulus n^3 is not equal to 0 then j = 2p/n^3. The total bit operations performed in the whole algorithm is found to be j*k (n^2 + 4n) + j*n^3.

The bit space required per shuffle is found to be n^3 + q, where 'q' is the bit size of the secret key.

The safety of the algorithm depends on Rubik's cube size specified by the user. Following are the permutations possible for a Rubik's cube of side length 'n'.

n=3 -> 4.3 * 10^19
n=4 -> 7.40 * 10^45
n=5 -> 2.83 * 10^74
n=6 -> 1.57 * 10^116
n=7 -> 1.95 * 10^160

Let 'x' be the total no. of permutations possible for shuffling a size 'n' cube. The total no. of permutations possible for shuffling the plaintext j times becomes x^j, if we do not consider the changes made by the vignere cipher. Therefore, we can safely assume that such a high value would give a safe encryption.

For n=7 the no. of permutations are in the degree 10^160, which means that if a brute force attack checks at the rate of 10^10 permutations per second, then it would take a total of 10^16 seconds to check all the permutations. Hence, Brute Force attack is not feasible on this encryption technique.

V. CONCLUSION

This algorithm is an attempt on creating a personal encryptor for users, wherein the access rights are based on the same principle as that of a home safe. The user is free to set the size of the Rubik's Cube, size of the secret key. The attacker to this algorithm is unaware from this information. This provides a high degree of safety as any attacker would have to check for all possible cube sizes and key sizes.

REFERENCES

[1] Ewing, John; Czes Kosniowski (1982). Puzzle it Out: Cubes, Groups and Puzzles. Cambridge: Press Syndicate of the University of Cambridge. p. 4. ISBN 0 521 28924 6. Retrieved 19 May 2014.

[2] Stephen A. Cook. Can computers routinely discover mathematical proofs? Proceedings of the American Philosophical Society, 128(1):40-43, 1984.

[3] Merrick Furst, John Hopcroft, and Eugene Luks, "Polynomial-time algorithms for permutation groups", *In Proceedings of the 21st Annual Symposium on Foundations of Computer Science*, pages 36-41, 1980.

[4] Erik D. Demaine1, Martin L. Demaine1, Sarah Eisenstat1, Anna Lubiw2, and Andrew Winslow3, "Algorithms for Solving Rubik's Cubes", arXiv:1106.5736v1 [cs.DS] 28 Jun 2011

Khaled Loukhaoukha, Makram Nabti and Khalil Zebbiche, "An efficient image encryption algorithm based on blocks permutation and Rubik's cube principle for iris images", in proceedings of *2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA).*

[5] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Systems Journal*, vol. 40, no. 3, pp. 614–634, Mar. 2001.

[6] Xiao Feng, Xiaolin Tian, Shaowei Xia, "An Improved Image Scrambling Algorithm Based On Magic Cube Rotation and Chaotic Sequences", in Proceedings of *the 2011 4th International Congress on Image and Signal Processing.*

[7] http://en.wikipedia.org/

[8] Vue Wu, Joseph P. Noonan and Sos Agaian, "Binary Data Encryption using the Sudoku Block Cipher"