**prospective titles:**

performance of encryption using proposed advanced rubiks cube algorithm

**Abstract**

**literature review**

**Introduction**

**Working Principle (include flowcharts and tree diagrams)**

**Structure of Code**

**Scope of Encryption Strength**

**Scope of technology on a deployment scale**

**plans of future development**

the proposed encryption system is inspired from the movements of a Rubik’s cube , the plane rotation of a Rubik’s cube results in a specific output state for a given unique set of Inputs (initial state and rotations performed).Thus can be used for effective Symmetrical Encryption . In prior attempts at using the mathematical model of a Rubik’s cube to perform encryption, the data was split into 6 main parts and each part was then arranged onto the 6 sides of the virtual Rubik’s cube, the plane turning operations were then carried out on theses surfaces to obtain the encrypted text.

To generate an algorithm on which a computer can efficiently act on the virtual Rubik’s cube was unfolded into a cross like shape to convert it into a 2-D dimensional system on which then linear / circular shift operations were performed on in accordance with the mathematical constraints of a Rubik cube. Storing of Data on the surface increases the complexity of the model (which in turn requires more computational overhead) as there are exposed two or three faces of each sub cube (commonly referred to as 'Cubie' in Literature) while the rest faces remain hidden inside the cube unexposed .This property of Rubik’s cube is what makes it convenient to build on the 2D abstraction of 3D system. Such an abstraction of a 3D System into a easier to program 2D system comes at the additional costs of increase in algorithmic complexity as illustrated in the figure where it can be observed that the a simple Rotation maneuver in 3D plane gets converted to a group of Linear and/or Circular shifts in the 2D plane. Each action while simple to intuitively understand in 3D system gets abstracted to a complex series of operations when done using the 2D analogy. Thus the 2D analogy method is computationally expensive, this drawback is the primary one overcomed in the proposed system.

The proposed system maintains the effectiveness of the diffusion obtained by a Rubik’s cube analogous system but is unique from other previously proposed systems on account of the fact that the data is stored in 3D matrix (at the center of each sub Cube) instead on the three or two exposed flat surfaces.

Here instead of using the surface of a Rubik’s cube on which the data is placed, the virtual Cube is imagined as a simple 3D matrix, which can be visualized as a collection of points along a cuboidal lattice as shown in the figure,

The mathematical Construct of such an arrangement is simply an array with 3D dimensions .the data is stored as member element of this array.

The array doesn't have any complex surfaces associated with each element hence can be converted into a simple algorithm (the actual program of which will also run faster hence giving an advantage).

The data is arranged directly into the 3D array whose side is large enough to accommodate the entire input data.

The rotation operation of the Virtual Rubik cube acts on planes of these member elements arranged in space as shown.it is observed that the complexity is much lesser

(Because the operation performed is a simple matrix slice rotation, instead of a series of 2D shift operations) which benefits by providing more encryption on less time.

The mathematical construct of the classic Rubik’s cube makes the faces on the surface have a relationship to each other, such relationships include the corner 3 faces to be always inseparable from each other as they are a part of one sub cube, and similarly for the 2 faces on the edge). Such relationships have been speculated to reduce the possible combinations of a Rubik’s cube's faces by a factor of 600.

By such relationships it is possible to reverse engineer the original cube with much lesser amount of Brute force as one element of a state to some extent predicts the positon of elements in that state. This is a potential downfall of using faces instead of points.

When the data is stored at point rather than faces then the rotation operation acts on the planes of elements that are inherently mathematically unconnected and their connection by virtue of the data is also broken down due the rotation.

Hence predictive and calculated forms of brute force based on the mathematical model will not be very different from the normal all-possibility brute force. Although it should be noted that there is not much crypto-analysis for previous Rubik’s cube encryption methods, due to the relative nascence of the technology.

Qualitative results being absent, quantitative results based on standard tests performed on a variety of text and image encryption algorithms have also been performed on an implementation of the proposed method, the results are summarized in the later sections of the paper.

**Working**

The data obtained initially from user is a linear stream, which is then used to populate the 3D array, the entire data should fit, and hence the length of the side of the cube is set appropriately.

For this the length of the data is cube-rooted and rounded upward (Ceil) to find the side of the virtual cube. Upward rounding is used to avoid loss of data, this results in memory locations of the array which remain unpopulated, these locations are populated with dummy data (like known text, spaces, common symbols, number digits).

Once the data is set into the cube we can use the plane rotation operations to scramble the text.

The standard cuboid Rubik’s cube has an orientation along 3 mutually perpendicular axes, for our convenience let us call them X, Y, Z.

It’s possible to rotate the cube in the YZ plane (about the X-Axis) at a particular plane number (which is the value of the X coordinate at that plane) for a certain fixed amounts of rotation in either the clockwise or counter-clockwise sense.

For simplicity we fix the direction of the rotation in one direction, when it is required to rotate in the opposite direction the value given for rotation will be negative.

Also it should be noted that since the number of sides are 4 , rotating the plane by 4 times leads to initial configuration , while rotating it by 3 times is equivalent to rotating it  -1 times.

If the virtual cube is taken and each plane of a specific axis is rotated for a fixed number of turns, and this is done for each axis, then what we obtain on the final cube is simply a change of orientation, as the all neighboring planes were scrambled together and their mutual relationship still has not been broken. Hence the original arrangement of the cube can be regained again with ease as there was no true scrambling.

Effective scrambling happens only when, about a particular axis, there is a pattern followed, where some planes are rotated while some are kept fixed. Also the pattern can incorporate spinning the plane in the opposite direction with the same magnitude as the normally rotating planes in the axis series. This will ensure that there is no correlation between each element any more. This jumbling is done in one plane (two dimensions) in simple cryptosystems which employ a permutation algorithm of one kind or the other .Here this jumbling is happening in a three dimensional arrangement hence the potential diffusion is much more.

This jumbling is simply followed in the reverse direction to obtain the original state of the cube, when performing decryption.

While reversing to decrypt, the sequence of axes are chosen in the reverse order to that chosen initially and each rotation operation is performed in the opposite sense of rotation than considered initially

Utilizing the proposed system we can convert a linear stream of information into a 3d matrix on which carefully controlled rotation operations can be applied along a desired axis at desired position(s). All these controlling parameters are what define the uniqueness of the encryption system, and a properly formatted parameter sequence code is the cryptographic 'key' for the system. This key dictates the encryption and needs to be sent to the receiving party for decryption.

A model encryption algorithm is created on the basis of the proposed technique. Here it is defined, and then explained in the following section.

For encryption/decryption a pass key is required. To demonstrate the algorithms basic feature a seven digit pass key will suffice.

The schema is outlined below using the tables.

Let the key be [ a b c d e f g ]

Where a,b,c,d,e,f,g are integers :

a,c,e,g van take values from 0 to 9

b,d,f can take values   1, 2, 3

'a' represents the shuffle sequence , the order of executing axes operations

For ex: a value of 4 represents 'ZXY' this implies: First execute spin actions about z- axis, followed by rotation in X axis, and then lastly spin about y axis.

for ex ; a value of 7 represents '-YXZ' this implies : First execute spin in opposite direction about y axis , followed by spin in X axis , lastly spin in Z axis.

'b' , 'd' , 'f' represent the amount of spinning that takes place in a fixed sense of rotation about the X , Y , Z axes respectively.

Theoretically it could take any integer value, but no response generated is unique for any integer other than the integers 1, 2, 3.

Integers which are a multiples of 4 should be avoided as there is no change in matrix arrangement at 4 rotations.

'c' , 'e' , 'g' Styles of turns:

The importance of not rotating each of the planes associated with a particular axis was explained earlier, hence there is a requirement that a particular fashion be followed in determining which planes to rotate and how to rotate them.

Refer to table 2, with the following legend:

'X' represents keep plane fixed

'/\' represents rotate plane (say clockwise)

'\/' represents rotate plane in the other direction (hence anti-clockwise)

To understand this let us consider an example:

style '1' : X-X-/\ : This represents keep two planes fixed and rotate the third one , after which repeat the pattern for the next set of planes ( like keep planes 4 and 5 fixed rotate 6 , and so on)

style '8' : X-/\-\/-/\-X  : This represents keep the first plane fixed, rotate the second plane, rotate the third plane in opposite direction, rotate the fourth in the normal direction, and keep the fifth plane fixed. Continue this pattern for every next set of five planes

This schema was adopted for the code developed to run the tests for text data and images for analysis.

A code [ 7 2 4 1 7 3 9 ]

Would imply following shuffle sequence '7' - (-YXZ) where the order of rotation is first Y axis spinning, followed by X axis spinning, lastly by Z axis spinning. Y axis rotates by default in the opposite direction.

For the Y axis there will be 1 rotation of style 7 (X-/\-\/-X) with the directions reversed

For the X axis there will be 2 rotations of style 4 (X-X-\/-\/)

For the Z axis there will be 3 rotations of style 9 (X-\/-/\-\/-X)

At the end of this sequence of rotations we will obtain the encrypted text.

**Thoughts:**

A lot more variations of the schema can be programmed into software to create powerfully unique ciphers.

The ability of an intruder to attack this system depends strongly on his ability to understand the different possibilities that are present, increase the number of possibilities and the difficulty for brute forcing increases exponentially.

so more schema can be added apart from the current 9 digit version with more complex shuffle sequences and the styles of turns can be added up to the length of the cube itself which is intended to be very long,

the more styles of turns there are there ,  more the predictability of the cipher is decreased and brute force attack on the text (not the method) then becomes the only way to solve the cipher which is practically impossible once the amount of data reaches more than 2^xxx .

Also the controlling parameters such as the shuffle sequence and style of turns can be totally redesigned arbitrarily and the newly generated schema converted to portable format which can be exchanged between two parties securely using techniques such as Diffie-Hellman Key exchange or RSA-KEY. After which the encryption becomes unique to a pair of people and brute force of data is the only theoretical option left, brute forcing the keys become useless as the operating algorithm now becomes potentially unique for each user pairs involved.

Hence the proposed system provides the ability to exchange two sorts of keys, a short encryption process key, and a long schema definition key. The long key can be sent every once in a while, whereas the short definition key can be exchanged more frequently to evade attempts at one-to-one function mapping if at all there is a successful brute force attempt at one transferred packet of the crypto-system.

In previously proposed crypto systems which are based around a Rubik’s cube, the side length was chosen to be 8, hence there were 8\*8\*8 cubies (sub cubes) and there were 6\*8\*8 = 384 Faces on which the data was placed. The new system proposed here does not have any fixed side length and can be scaled up to any requirement. Moreover the data is not stored on the surface, it is stored in the volume, and hence there is a more density of diffusion. This ensures that the data is scrambled thoroughly and not in small individual packets of fixed sizes which can be known to an intruder.

Additionally the strength of the cipher can be increased many folds by introducing one more parameter to the system. If instead of using one large 3D matrix we can give a choice to user to split the data into a number of equal sized streams which can be encrypted individually .The novelty being that the number of independent streams into which the main big data was split into will be unknown to an intruder unless he is also having excess to the schema (long key).

Hence Data can be partitioned, which at a small increase in computational costs renders exponential increase in the brute force range,

The advantage is especially high in case of very long input data streams.

The proposed algorithm creates complex diffusion, and is the most effective when the data size is moderately large .It is almost ineffective at small cube sizes, like most encryption algorithms due to susceptibility to ‘dumb-bruteforcing’, when the type of algorithm employed is known to a skilled intruder, then the mathematical model of the algorithm can be used to deconstruct and reconstruct the algorithm to create an intelligent brute forcing strategy. When the data is made sufficiently large to begin with then most good algorithms can make the data secure against dumb brute forcing , and when the algorithm is designed for maximum possible diffusion then the algorithm is also safe against intelligent/designed brute forcing .In case of the proposed algorithm diffusion is introduced using the various ways in which a rotation is performed (axis of rotation , position of rotation , magnitude of rotation , sequence of rotation of other axes , number of sub cubes generated from the main data.

The proposed algorithm is explained to be highly diffused, as there are two factors which are given importance Diffusion and Confusion. Diffusion relates to the Permutation (Transposition) of the data, while Confusion relates to the transmutation (transformation) of the data from its prior from.

Confusion modifies the data, while diffusion changes the sequence of it.

The algorithm so far has described a method for string diffusion, but the confusion aspect is not touched, yet.

For confusion a variety of different processes are available. Most commonly used encryption techniques have a diffusion technique in conjunction with a confusion technique to strengthen the cipher even more.

The most widespread technique used for confusion (transmutation) is the XOR technique. Owing to its widespread use its details will not be highlighted her, but a review can be found here [5756].Integrating the XOR will lend even more strength to the proposed cipher .The secret key used for XORing need not be very long, as it is conventionally required and can be a simple 4Bit word which is used recursively every 4 bits of the main input data. The word used can become an addition to the short key of the cipher. Such a level of confusion will be enough for high amounts security without too much increase in computational power required.

Similar to the rounds of successive confusion

and diffusion performed on a data to obtain extremely high amounts of possible values to immune the data from brute forcing , done as in the cases of commonly used algorithms like 3DES and AES , can also be implemented in the algorithm used , but it is only recommended if the security provided falls short , as the computational power is linearly proportional to the rounds of encryption done .The concept of rounds is a necessity in case of the ciphers mentioned above as they not only have a fixed block size , also the size of the block is smaller than the potential variable size of 3D matrices in the algorithm here. Also the dynamicity of the algorithm renders it difficult to be used as a stream cipher as the ideal requirements of a stream cipher a speed and quick response (i.e. Generation of output data from small input data without requiring too much context).

At its core the proposed algorithm is a transposition (permutation) cipher while techniques which use transmutation are more suited to the process as the data is more like a linear stream.

Analysis of the proposed algorithm is done without including the additional features of undeclared data compartmentalization, and XOR operation as it can be safely speculated that these features will only increase the security by increasing diffusion and confusion respectively.

Future prospects :

GPU based implementations

Full scale Mathematical analysis to determine if it is possible to create a intelligent brute force attack.

too much work to be done

shortcoming of the cipher effective for large files , also requires a lot of power.