Breaking Rotacrypt: Cryptanalysis Breakdown of Rotational Mechanics as a Cryptographic Primitive

Teo Honda Scully

Abstract

...TBD...

Contents

1 Introduction

Encryption, the process of encoding information in a way that only authorized parties can decipher it, lies at the heart of secure digital communication. It serves as a fundamental tool for safeguarding our messages, files, and personal data from unauthorized access and eavesdropping.

1.1 Outline

In the ever-evolving landscape of encryption, innovative approaches continue to emerge, each inspired by unique and unexpected sources. I propose the worst of them all, Rotacrypt, a new cryptosystem that employs the mechanics of the Rubik's Cube to create a fresh paradigm in secure communication.

Rotacrypt uses Rubik's Cube mechanics to construct a mathematically intricate and seemingly secure encryption scheme. Despite its promise, the system contains an inherent flaw related to the cube's cyclic rotational limitations. The latter part of this paper will perform a cryptanalysis of this flaw, showing its capacity to compromise Rotacrypt's security.

This paper first introduces Rotacrypt, a nearly flawless cryptosystem, and then proceeds to break it due to a singular flaw in the scheme's cryptographic primitive.

2 Introduction

Ah, the Rubik's Cube—the iconic toy that bedevils and delights in equal measure. Born from the ingenious mind of Ernő Rubik in 1974, it's more than just a puzzle. Forget mere child's play; this cube is chaos in the physical. Just kidding. To go from the state of chaos to order, one only needs to know a solving protocol of which many exist.

2.1 Review of Rubik's Cubes

Max Park, the current world record holder (11 June 2023) for the 2-handed solve, obliterated the cube in a mind-blowing 3.13 seconds. Lucky scramble? Hardly. Yiheng Wang, who holds the world record average-of-five, clocks in at a dizzying 4.48 second mean solve time throughout the five solves.[1] And no, I won't depress you by mentioning his tender age of nine years old. But let's not divert. The Rubik's Cube–a mathematical marvel and a plaintext[2] waiting to be encrypted.

Let's dive into the mechanics of the 3x3x3 puzzle. The cube boasts centers, edges, and corners. The single-colored center pieces serve as the invariant axis around which the peripheral smaller unit cubes rotate. The six unit colors are yellow, blue, red, green, orange, and white.

[1] An average-of-five is determined by taking the average of the three "middle" solves in a session of five scramble. In other words, the worst and best solve time are dropped from the calculation for the average.

[2] In cryptography, plaintext refers to the original readable message, while ciphertext is the scrambled message produced through encryption.

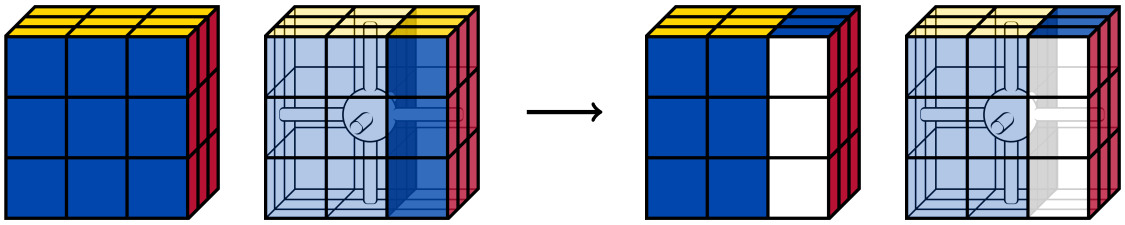
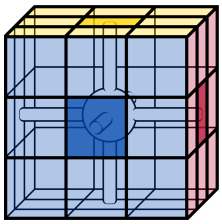


Figure 1: A visualization of the R operation (rotating the right layer clockwise).



A visualization of a 3x3x3 cube. The cube has 6 faces, each with 9 stickers. Notably, this means that there exists 54 different unit tiles on the cube. The cube has 43,252,003,274,489,856,000 possible states.[3]

For instance, if the cube is held with a yellow top and blue front, the red and orange faces will invariably be to the right and left, respectively. In fact, the red and orange center pieces will *always* be opposites, as will the blue-green and white-yellow pairs of center pieces.

2.2 Notation for Cube Operations

The notation used to describe the movements and algorithms for solving the Rubik's Cube is standardized to facilitate easy understanding and sharing of solutions. Each face of the cube is designated by an uppercase letter:

- **U** - Up (Top Layer)
- **D** - Down (Bottom Layer)
- **L** - Left (Left Layer)
- **R** - Right (Right Layer)
- **F** - Front (Front Layer)
- **B** - Back (Back Layer)

The following symbols are appended to these letters to indicate the direction of rotation:

- No symbol - 90-degree clockwise rotation
- **'** (apostrophe) - 90-degree counterclockwise rotation
- **2** - 180-degree rotation (either direction)

[3]This number is calculated by considering the 8 corners, each with 3 orientations, and the 12 edges, each with 2 orientations.

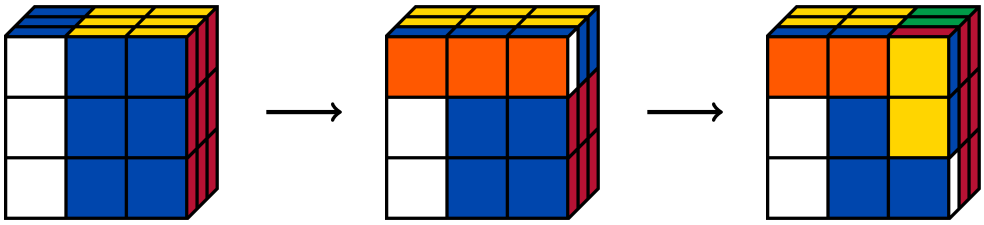


Figure 2: A visualization of the *L' U' R'* operation(s).

For example, the sequence *L' U' R'* would indicate a counterclockwise rotation of the left layer, followed by a counterclockwise rotation of the top layer, and finally, a counterclockwise rotation of the right layer.

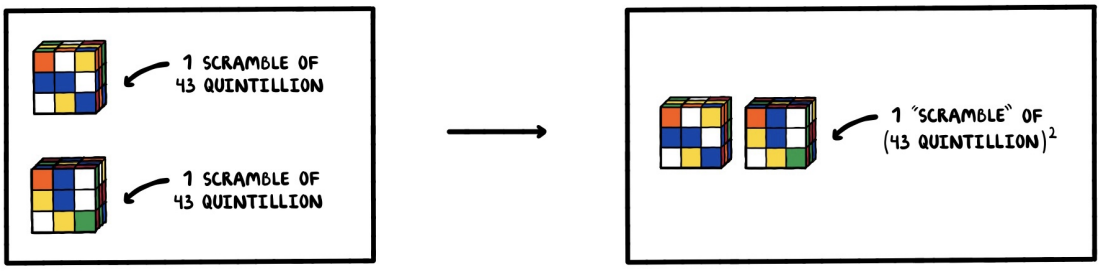
2.3 God's Number

In the realm of 3x3x3 Rubik's Cubes, *God's Number* is a term used to denote the maximum number of moves required to solve any scrambled cube. It has been proven that any cube can be solved in 20 moves or fewer (the citation can be found in the *References* section). This concept is an intriguing insight into the mathematical efficiency of the cube's design.

Furthermore, this means that any scramble can be reached with 20 moves or fewer. This is a key point to keep in mind when considering the security of the cube as a cryptographic primitive as well as for key size considerations.

2.4 Combinatorial Explosion with Multiple Cubes

Let us consider the number of possible states for a single 3x3x3 Rubik's Cube, which is $43,252,003,274,489,856,000$. This is not large enough for a secure cryptographic state space. However, when chaining[4] together the combinations of two different cubes, the number of combined states is $(43,252,003,274,489,856,000)^2$.



This squaring occurs because each state of the first cube can pair with every state of the second cube, yielding $43,252,003,274,489,856,000 \times 43,252,003,274,489,856,000$. Mathe-

[4]Linking the encryption of one block with the next, making the entire data set more secure and diffused by causing a ripple effect throughout the blocks.

matically, the set of possible states becomes the Cartesian product of the two sets of states, leading to an exponential increase in complexity.

The intriguing aspect of chaining multiple Rubik's Cubes is the notable exponential growth in the state space[5]. Let N represent the number of unique states for a single Rubik's Cube. For k chained Rubik's Cubes, the total number of unique states becomes N^k. This exponential increase serves a critical function: it substantially minimizes the likelihood of collisions whilst simultaneously increasing the difficulty of brute-force attacks.[6]

2.5 Edge and Corner Constraints

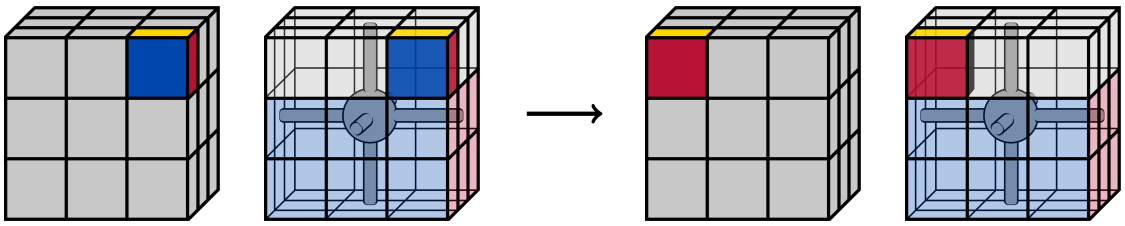


Figure 3: A visualization of categorical immutability. Corners will always map to corners.

One of the fundamental constraints of a Rubik's Cube is that edges and corners are immutable in their categories; edges cannot morph into corners and vice versa. Due to this constraint, it becomes inappropriate to assign all 48 movable units (54 total units minus 6 centers) to represent bits, as doing so would leave patterns in trivial plaintext-to-ciphertext cryptanalysis attacks.[7] If a bit mapped to a corner unit can never be encrypted into an edge piece no matter how many layer operations occur, this serves as a trivial pattern in which cryptanalysists can exploit.

While the utilization of multiple Rubik's Cubes in the encryption scheme introduces a combinatorial explosion in the state space, it is essential to consider the limitations imposed by the unique and contained mapping of plaintext bits to chosen cube pieces. Each plaintext bit (either a 0 or 1) is mapped to a specific piece on the cube, which can either be an edge or a corner for the entirety of the scheme.

2.6 Per-Cube 24-bit State Space

As a result of the aforementioned limitations, the encryption scheme opts for a more constrained assignment by sticking to either edges or corners to represent the bits. This decision narrows down the number of units used for bit representation to 24, thus limiting the state space to 2^{24} for each cube.

[5]The state space refers to the total number of possible configurations or states that a system can be in; for a cube, it's a very large number.

[6]In cryptography, a brute-force attack involves trying all possible combinations to decrypt a message, which is computationally expensive.

[7]Cryptanalysis attacks involve mathematical and computational techniques to analyze and possibly break an encryption scheme.

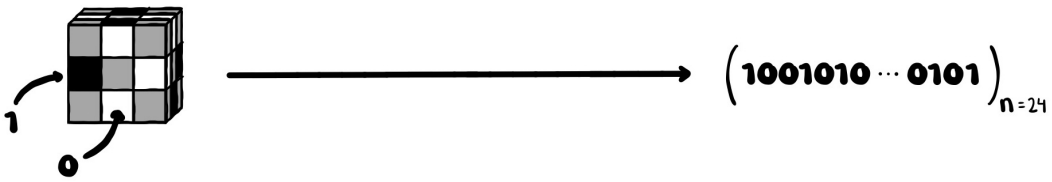


Figure 4: The serialization of a plaintext-mapped cube state. Black units represent 1s, while white units represent 0s.

A serialized 24-bit bitstring yields a state space of 2^{24} *because there are* 2^{24} *different ways to pick and choose 0s and 1s across 24 elements. For the first bit, there are 2 possible choices for the value. For each of those choices, the second bit can also have 2 possible values. This pattern continues for all 24 bits, leading to* $2 \times 2 \times 2 \times \ldots$ *(24 times)* $\rightarrow 2^{24}$.

Given the inherent limitations in the state space of a single Rubik's Cube, capped at 2^{24} due to geometric constraints, a novel approach to enlarging this space involves chaining multiple cubes together. Specifically, by employing a tuple of six Rubik's Cubes[8] for each block, the scheme effectively increases the state space to $2^{(24\times6)}$.

In this enhanced scheme, each block of plaintext is split into six portions, each of which is mapped onto a separate cube. The combined state of all six cubes serves as a unique representation of the original block. This approach leverages the geometric diversity across multiple cubes to create a more complicated and less predictable state space.

With a state space of $2^{(24\times6)}$, the algorithm becomes computationally prohibitive for brute-force attacks, even when accounting for potential parallelization.[9]

While the increase in state space significantly boosts the algorithm's resilience against attacks, it also introduces additional computational complexity. However, the impact on efficiency is deemed acceptable given the substantial increase in cryptographic security.[10]

2.7 Summary

The *Introduction* can be summarized with the following points:

- **3x3x3 Mechanics**: The cube boasts centers, edges, and corners. The six unique single-colored center pieces serve as the invariant axis around which the peripheral cubies rotate.
- **State Space**: The cube has 6 faces, each with 9 stickers. Notably, this means that there exists 54 different unit tiles on the cube. The cube has 43,252,003,274,489,856,000 possible states.

[8]In this scheme, a tuple refers to an ordered set of six individual Rubik's Cubes, each contributing to the encryption process.

[9]Parallelization refers to the process of dividing a task into sub-tasks that are solved concurrently, often used to speed up computational tasks.

[10]The term refers to the resilience of a cryptographic system against unauthorized access or data breaches.

- **Increasing State Space**: Let N represent the number of unique states for a single Rubik's Cube. For k chained Rubik's Cubes, the total number of unique states is N^k.
- **No More Colors**: In this scheme, colors are not mapped to units on the cube. Instead, each plaintext bit (either a 0 or 1) is mapped to a specific piece on the cube. As centers are immutable, they are not used to represent bits (only 48 movable units).
- **Mapping Constraints**: One should note that edges cannot morph into corners and vice versa. Therefore, bits are only assigned exclusively to either edges or corners (24 potential bit mappings per cube), thus limiting the state space to 2^{24} for each cube.
- **State Space Again**: By employing a tuple of six Rubik's Cubes for each block, the scheme effectively increases the state space to $2^{(24\times 6)}$ (yielding 2^{144} possible states).

3 Overview

The proposed encryption scheme is a block cipher[11] that leverages the mathematical complexities and state space of Rubik's Cubes. Each block in the scheme utilizes a tuple of six 3x3x3 Rubik's Cubes, effectively rendering a block size of 144 bits (24 bits per cube; 6 cubes).

3.1 Initial Round

At the start of the encryption process, each cube is initialized to the solved state and then a corresponding-plaintext-bit-to-3x3x3-layer-operation-mapping is applied. The details of this progress can be found in the *Encryption / Cube Mapping Procedure* section. This will yield six unique cube states.

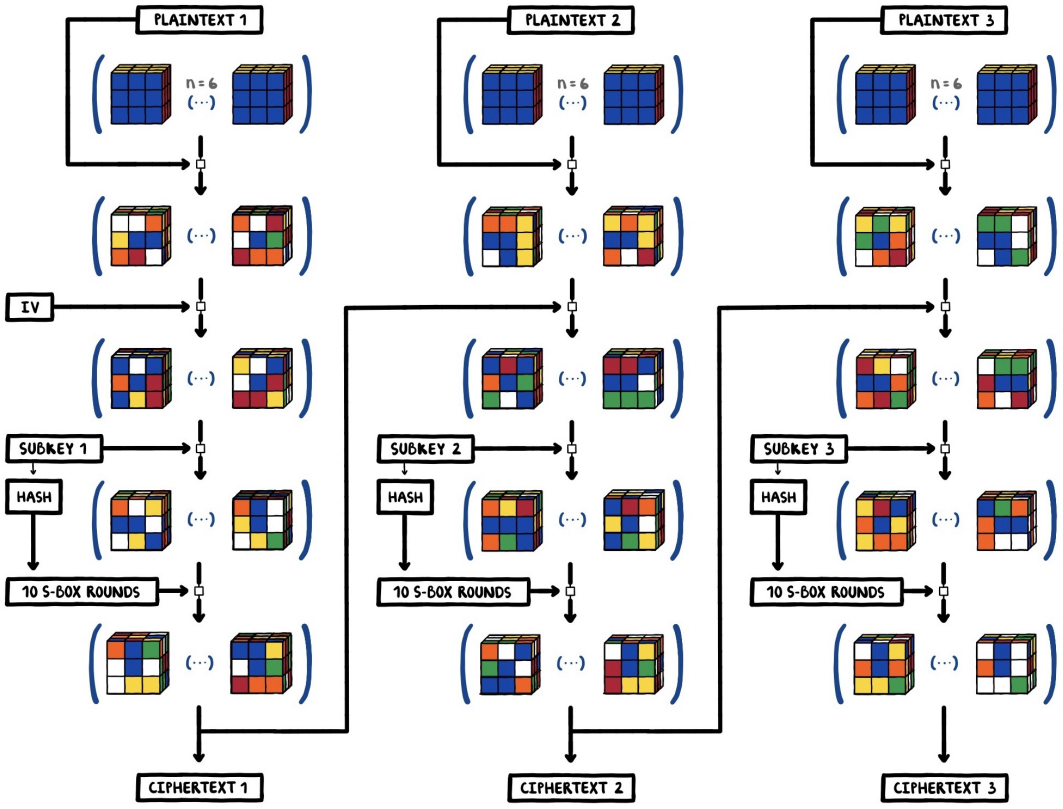


Figure 5: Rotacrypt Encryption Overview

The initial round begins by applying a cryptographically secure pseudorandom number generated Initialization Vector[12] (IV) deserialized into six unique scrambles. Each cube in the

[11] Encrypts chunks of plaintext data (blocks) using unique subkeys for each block.

[12] An Initialization Vector is a cryptographically secure random number that is applied during initial encryption, ensuring distinct encrypted data every time (even with equivalent plaintext inputs). This technique disrupts pattern recognition attempts, particularly in chosen-plaintext attacks.

tuple will have their own unique scramble at this stage, and a dynamic subkey[13] is then generated from the original key using a secure key derivation function. The subkey is likewise deserialized[14] into six unique scrambles. Each scramble is applied to their respective cube. The cubes finally undergo a series of deterministic Rubik's moves, mimicking the traditional S-box[15] and diffusion[16] operations in AES like SubBytes, ShiftRows, and MixColumns. These Rubik's moves are termed as SubCubes, ShiftFaces, and MixEdges respectively. It should be noted that the nature of layer operations intrinsically involves a form of diffusion, making the scheme's diffusion already naturally high without S-box operations. The resultant state of the cubes serves as the ciphertext for that block.

One may question the intent of an S-box operation if the layer operations already provide a high degree of diffusion. Layer operations are linear transformation[17] (this becomes evident in *3x3x3 Implementation*), and the S-box operation adds a layer of non-linearity to the encryption process. This non-linearity is critical to the security of the scheme, as it prevents the encryption process from being modeled as a linear system of equations.

3.2 Subsequent Rounds

For subsequent rounds, a part of the cube state from the previous round is extracted and used as the IV for the next block. New subkeys are generated dynamically by hashing the original key along with a salt (detailed in the *Key Generation / Sub-Key Generation* section), which is then converted into a Rubik's Cube scramble sequence. The same sequence of operations: SubCubes, ShiftFaces, and MixEdges, are then applied to these new blocks, followed by scrambling with the round-specific subkey.

The state of the cubes after the S-box operation is used to link subsequent blocks, ensuring that the entire encryption process influences each block. This design decision not only maximizes the use of the large Rubik's Cube state space but also provides strong cryptographic properties.

We are chaining blocks to increase diffusion. A blockchain if you must.

3.3 Final Round

(MAYBE... TBD) In the final round, the processed cubes go through an S-box transformation to further improve the security of the encrypted data. This S-box is carefully designed to maximize non-linearity and is dynamically generated based on the cube's final state.

[13] A subkey is a key derived from a master key for use in a particular cryptographic algorithm. In this scheme, subkeys are generated dynamically for each block

[14] Turning the cube's state into a simple format like a string or array for easier data handling.

[15] A substition box (S-box) is a cryptographic component that performs fixed, non-linear substitutions on input bits to produce output bits.

[16] Diffusion is a cryptographic concept that refers to the spreading of plaintext information throughout the ciphertext, making it difficult to decipher.

[17] Linear transformations are mathematical operations that preserve the structure of the underlying space, such as rotations and reflections.

3.4 Purpose

There is no purpose. I was told not to make a cryptosystem, so I did the opposite.

3.5 Intended Usage

¡···¿

4 3x3x3 Implementation

¡···¿

4.1 Data Structure Breakdown

¡···¿

4.2 Augmented SPEFFZ Mapping

¡···¿

4.3 Cyclic Transformations

5 Key Generation

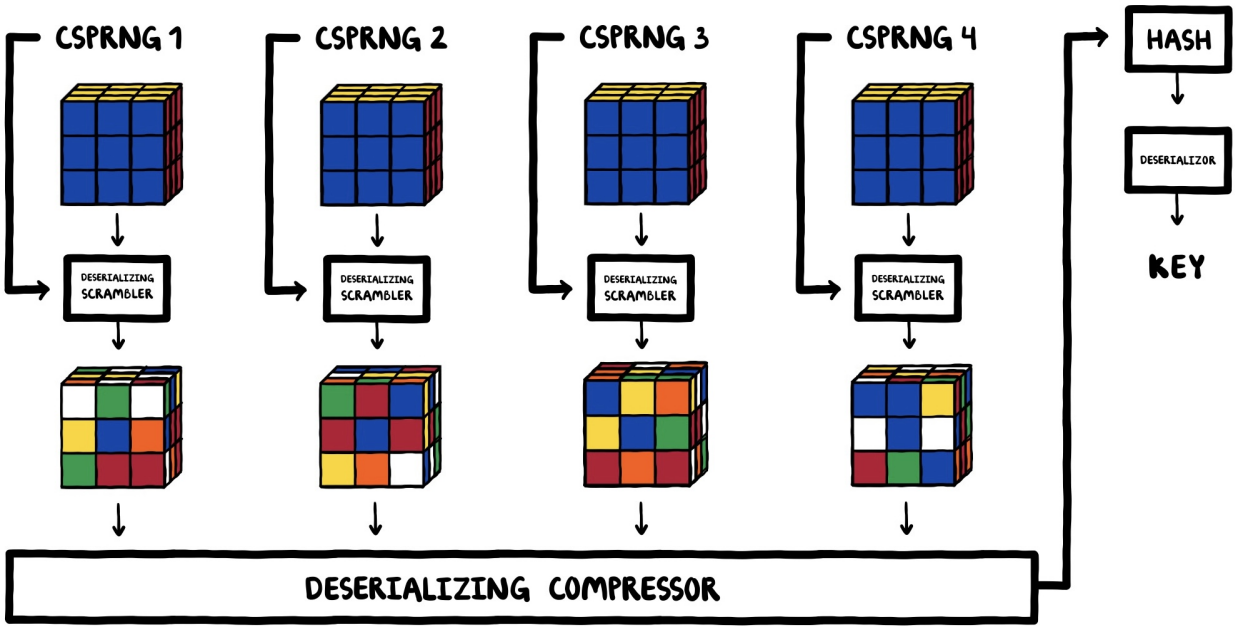


Figure 6: beep boop

5.1 4-Cube Initialization

¡...¿

5.2 Master-Key Serialization

¡...¿

5.3 Sub-Key Generation

¡...¿

6 Encryption

¡...¿

6.1 Plaintext Setup With S-Box Transformations On Chunks

¡...¿

6.2 Cube Mapping Procedure

¡...¿

6.3 Encryption Algorithm

¡...¿

7 Decryption

7.1 Beep Boop

8 Security Analysis

8.1 Immediate Reduction to AES

9 Codebase Architecture

9.1 Architecture Tree