Rubik's Encryption of Whatsapp Messages

Nijo Nelson[1], Michelle Aijo[2], Nikhil V S[3], Sania Saji[4], Shejina N M[5]

[1,2,3,4] *Student, Department of Computer Science & Engineering, IES College of Engineering, Thrissur, Kerala*

[5] *Assistant Professor, Department of Computer Science & Engineering, IES College of Engineering, Thrissur, Kerala*

Email_id: nelsonnijo11@gmail.com, michelleaijo@gmail.com, sani916sani@gmail.com, nikhilvs550@gmail.com, shejina@iesce.info

Abstract

Privacy concerns are at an all-time high, as is the right to keep such conversations secure and confidential. Users routinely are met with claims of end-to-end encryption. However, control is almost always with large corporations who can access data at will. Therefore, we propose a new system to be used with a Rubik's Cube based encryption technique to WhatsApp Web. This system encrypts and decrypts on the user's device, allowing messages to be viewed only by the sender and designated receiver. Client-side control of encryption shifts the balance of authority and control of the keys to the user, preventing access and building trust with the messaging system. This encryption scheme is designed to apply to stored text messages, with the intent of adding to the security of multimedia messages exchanged through the system.

Keywords: Client-Side Encryption, Rubik's Cube Algorithm, Data Privacy, End-to-End Security, Whatsapp Web, Cryptography.

DOI: https://doi.org/10.5281/zenodo.19000719

1. Introduction

Data privacy and digital security go hand-in-hand in today's world, and as a result, people use messaging services to provide and receive sensitive information. However, users should note that there is a privacy gap, particularly in services that utilize encryption and are operated by large, centralized companies. Trust is undermined due to users being unaware of and/or concerned about unauthorized access, data mining, and surveillance of their personal information.

Encryption via Rubik's Cube is a novel and highly effective. The concept shifts encryption and decryption to the client-side, allowing complete control by the user to protect their confidentiality. The Firefox web extension designed in this project works with WhatsApp Web and enables users to independently encrypt and decrypt their messages while preserving end-to-end encryption.

This project demonstrates that the fusion of spatial data manipulation and encryption improves the privacy, security, and trust of digital communication by Rubik's Cube encryption. More so, this project is now completing a more highly user-controlled and confidential environment in which to send messages freely.

1.1 Firefox Extension Architecture

The core of the system consists of a client-side middleware that operates on the Firefox browser. This middleware injects scripts into WhatsApp Web to capture messages locally, offering control over all message processing, and allowing the user to dictate the parameters of all cryptographic operations.

1.2 RUEN Encryption/Decryption Module

The Rubik's Cube Cipher's logic is encapsulated within a module that functions independently to modify messages. Message modifications (i.e., encryption) occur during sending, while messages are restored (i.e., decryption) during receiving to guarantee the highest level of security.

1.3 User Interface (UI) Integration

An easily operable extension targeting WhatsApp Web systems in relation to interactable buttons for easy access to encrypting, decrypting and view manipulator for hidden messages, all with no interference from the built-in systems

1.4 Data Store API and Key Management

This integrated module manages and verifies symmetric encryption keys stored locally in the browser's Data Store API, assuring that the correct key is used in the completion and validation of the authentication and decryption for every chat.

2. Literature Survey

Saxena et al.[1] Saxena and colleagues thoroughly assessed the security measures for virtual machines within the scope of the cloud computing environment. They explored real-world datasets concerning the effects of system resource management on the system and security. By performing a cloud security techniques meta-analysis, they describe the compromises needed on security and the ability to protect the system from modern threats. Research has shown how cloud systems can leave users vulnerable to attack, especially if there are configuration errors, and how this makes a case for flexible system encryption. This study suggested how to build a security system that is efficient and at the same time leads to the optimal combination of available resources and privacy to create Rubik's Cube-based encryption systems.

Barker et al.'s [2] cryptographic agility. The researchers argue that this is a necessity due to the evolving menace of quantum computing. More than a pure technical advancement, this capability is a fundamental change in strategy that calls for flexible adaptive design and architecture along with engineering agility. For technical agility, Barker et al. advocate modular design and the use of Cryptographic APIs which isolate the application code from the encryption algorithms, allowing for faster changes to the algorithm. For organizational agility, Barker et al. argue that this is primarily the responsibility of effective enterprise governance along with a supervisory control that tracks all deployed cryptographic systems and ensures that security policy and incident response plans account for the cryptographic changes to mitigate operational friction and exposure.

Sadia Syed and Albalawi [3], Sadia Syed and Albalawi examined the application of machine learning approaches to the improvement of strategic allocation of cloud resources and the enhancement of cloud computing. This research demonstrated whether and how smart algorithms could alter resource allocation on a continuous basis

in order to increase efficiency and decrease the cost of operating a cloud. However, a major contribution of this research to the field was the demonstration of the degrees of interdependence between the volume and quality of input data, the machine learning algorithm(s) employed, and the efficiency and effectiveness of the model produced. This demonstrated that although there appear to be no limits to the application of machine learning in cloud computing, there are considerable limitations to many applications that result from the lack of sufficient data and excessive processing requirements.

Renata Teófilo de Sousa et al. [4], the authors used the GeoGebra application along with the Rubik's Cube and demonstrated the advanced mathematical concepts required to solve the cube, in parallel to the algebraic operations. For high-entropy cryptography, the cube's combination of algebraic manipulations and structural changes provides the necessary intuition to model advanced encryption techniques'.

Swastik Kumar Sahu and Kaushik Mazumdar [5] go into great depth and detail, elaborating on the theoretical concepts, the quantum key distribution (QKD) model, and the ability to protect future digital data from quantum computer attacks. They also acknowledged the security of information using quantum methodology, and described the challenges of the expensive and complex technology, the short distances it can work over, and the differences between quantum computers and classical ones which may be the reason these methods have not been implemented.

Manoj V. Bhagwath and A. Rengarajan [6] created a multi-faceted cybersecurity enhancement. Their work achieved a novel and effective integration of two distinct and versatile cryptographic security paradigms using a deep learning approach. This work illustrated the extreme fragility of the deep learning approach to biometrics authentication, as the original, unsecured biometric input could then be transformed into secure cube permutations prior to authentication, thereby substantially augmenting the resiliency of the cybersecurity approach and illustrating the effectiveness of hybrid security systems to protect sensitive data.

Aviral Srivastav and Aryaman Kumar [7], who designed the hybrid of the Advance Encryption Standard (AES) which works at a faster speed, and the RSA (Rivest – Shamir – Adleman) which is more reliable during of the key exchange. The originality of their work was the protection of a key at the kernel level, a protection mechanism of risking the loss of cryptographic keys which is in danger of being lost through malware in the user-space or by remorseless intrusion. This modification on the system architecture increased the overall system strength in encryption and system resilience by addressing the key management and storage issue, and the system cryptographic inbalances.

Salkinder [8] conducted a thorough examination on a group-theoretic foundation of a generalized n × n × n Rubik's Cube, where he abstracted it to algebra and defined the movements and configurations of the puzzle at wrap of each rotational shifting of the puzzle. This study demonstrated the solvability of the puzzle at a given configuration and bounded the number of moves required to achieve the solution. Most importantly, this work provides the first widely recognized proof and validated the theoretical description of the design and degrees complexity of any cryptographic system that incorporates puzzle-based transformations.

Mohammed and Varol [9] compared and contrasted the attributes and security of computer and public key and private key parameters of cryptography and their corresponding applications. The study reviewed the symmetry algorithms and the RSA encryption algorithms and their key exchange functionality to understand modern asymmetric

and symmetric cryptography and create blueprints for modern cryptography.

Hena Rani Biswas et al. [10] discussed The Principles Of Chaos Theory And Its Multifaceted Applications In Real World Engineering Systems. This research explored the ways in which complex nonlinear unpredictable behaviours of chaotic systems can be harnessed. In particular, chaos theory offers a framework in the field of cryptography to develop pseudo random number generators, Design multiple layers of diffusion and construct many encryption algorithms That are sensitive to initial conditions and complex their behaviours are mixed to extensively secure the algorithms.

Erik D. Demaine et al. [11] conducted an extensive computational study of the algorithms used in solving Rubik's Cubes, determining the fewest number of moves needed to solve various positions. They provided asymptotic bounds which indicate the complexity of the transformations of the cube. This results in accurate understanding of the computational complexity involved in cube-based encryption/decryption (the forward process) and cube-based cryptanalysis (the backward process) facilitating the estimation of complexity for the cipher. This is important for cube-based cryptography for determining the effort required to perform brute force attack.

2.1 Review of Methodology

2.1.1 Block Level Design - 0

Figure 1: Block Level Design - 0

- This diagram illustrates the core idea of the RUENED system, which uses Rubik's Cube transformations for both encryption and decryption.
- It highlights the direct flow of user input messages into the encryption/decryption process, resulting in ciphertext output.
- The design abstracts away the internal mechanisms, focusing instead on the high-level functional relationship between input, RUENED, and encrypted output.

2.1.2 Block Level Design - 1

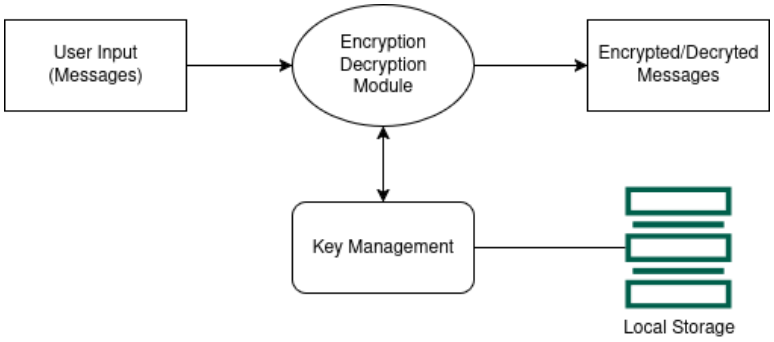

Figure 2: Block Level Design - 1

- This level introduces key management and local storage, emphasizing how encryption and decryption processes rely on secure key handling.
- The encryption/decryption module interacts dynamically with the key management unit, ensuring consistent key usage for message transformation.
- It represents an improved architectural view showing system modularity—separating data processing, key storage, and user interaction layers.

2.1.3 Block Level Design – 2

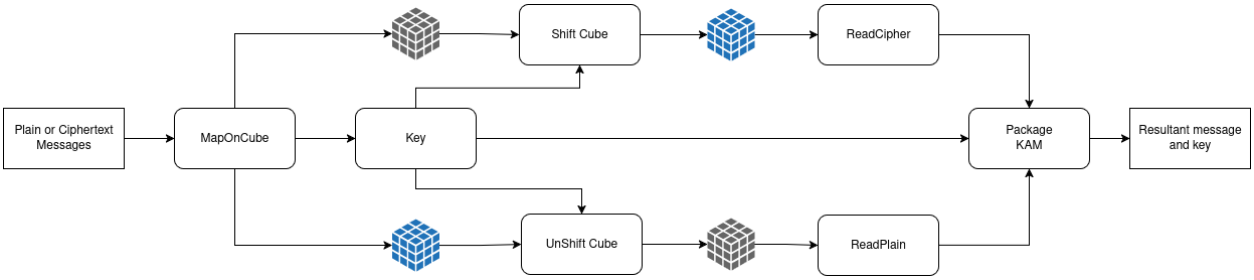

Figure 3: Block Level Design - 2

- This diagram breaks down the complete RUENED process, from mapping plaintext or ciphertext onto the Rubik's Cube to shifting/unshifting operations using a key.
- It details the transformation pipeline, including cube manipulation, key generation, reading cipher/plain text, and packaging the final output.
- The design captures both encryption and decryption workflows, showcasing the reversible and symmetric nature of the Rubik's Cube–based algorithm.

3. Review of Data Formats and Information Flow

3.1 Message Data

This data comprises both Plaintext Messages and Ciphertext Messages. Plaintext Messages are the original, readable messages entered or received by the user via WhatsApp Web. Ciphertext Messages are the encrypted, unreadable output generated by the RUEN algorithm, transmitted over the network, and later decrypted by the recipient. These records are critical for maintaining a private, user-controlled communication layer, thus enhancing confidentiality.

3.2 Encryption Key Information

This dataset includes both Symmetric Keys and Key Metadata. The Symmetric Keys (or shared secret) are the cryptographic keys used for both the encryption and decryption processes; they must be securely exchanged out-of-band between communicating parties. Key Metadata includes information like the chat ID, timestamp of creation, and key version, which lend context and manage the key's legitimacy within the extension. The accuracy and consistency of key management are crucial for successful message exchange.

3.3 Algorithmic Transformation Data

This data consists of the Cube Structure Data and Permutation Logs. Cube Structure Data is the intermediate

3D data array generated when the plaintext message is mapped onto the virtual Rubik's Cube structure, enabling the spatial manipulation aspect of the cipher. Permutation Logs are the specific sequences of Rubik's Cube moves and columnar transpositions applied during the encryption process, which form the basis of the key material. The fidelity of these records is paramount, as they directly dictate the strength and reversibility of the encryption.

3.4 Key Storage Specifications

This involves the Data Store API Records and Local Storage Entries. Data Store API Records represent the mechanisms and rules encoded within the extension's key management layer, automating and enforcing key retrieval and storage processes. Local Storage Entries consist of the actual data written to the user's browser storage (or sync storage), tracking how keys are stored and ensuring they are accessible only to the authorized extension. This dataset must accurately reflect the key's location and state to highlight any issues or anomalies in retrieval.

3.5 Integration and Flow Data

This data includes Message Status Codes and Listener Events. Message Status Codes track the state of a message (e.g., "Ready for Encryption," "Ciphertext Sent," or "Decryption Failed"), ensuring the RUEN module is functioning correctly at each step.

3.6 System Utility Data

Utility data encompasses the Extension Status and Error Logs. Extension Status data measures whether the extension is active and configured correctly in the user's browser, important for assessing system availability. Error Logs document errors and issues related to key retrieval failures, decryption mismatches, or script injection problems, helping developers identify and resolve problems to maintain smooth operation.

3.7 User Interaction Data

The dataset includes information on Control Button Usage and Failure Notification Feedback. Control Button Usage data captures how users engage with the one-click encrypt/decrypt buttons, offering insights into usability and identifying areas that may require enhancement. Meanwhile, Failure Notification Feedback provides crucial details about user actions following an error (e.g., did they manually enter a new key?), enabling the system to be refined and improved based on real-world user interactions.

4. Implementation of a Rubik's Cube Encryption in a Firefox Extension

The implementation of the RUENED system focuses on integrating Rubik's Cube–based encryption and decryption seamlessly into real-time messaging platforms such as WhatsApp. The goal is to ensure that message security is strengthened without disrupting the user's usual communication experience. In a real-world setup, the system would function as an independent encryption layer that operates between the message input interface and the WhatsApp transmission process. When a user types a message and opts to encrypt it, the system intercepts the plain text before sending, processes it through the RUENED module, and outputs the encrypted ciphertext, which is then transmitted through WhatsApp's normal channels.

At the core of the implementation lies the Encryption–Decryption Module, which handles all transformation processes. This module receives the message and performs the Rubik's Cube–based encryption algorithm, where text is conceptually mapped onto a virtual cube. Using a unique key, the cube's layers are programmatically rotated to

rearrange the message characters, resulting in a complex yet reversible encrypted form. Upon reception, the recipient's module retrieves the key and executes the inverse operations to reconstruct the original message. This entire process occurs within milliseconds, maintaining real-time communication speeds.

To manage encryption keys securely, a Key Management System is incorporated. Each encryption instance generates or retrieves a unique key that defines the cube's shifting pattern. These keys can be locally stored in an encrypted form for temporary use or regenerated as needed, ensuring both flexibility and confidentiality. In cases where persistent storage is required, keys are securely saved with strict access permissions, preventing unauthorized retrieval. This mechanism ensures message integrity and prevents the exposure of sensitive encryption parameters.

The backend architecture integrates various components such as data mapping, key assignment, cube rotation logic, and message packaging. Initially, the message is broken into smaller chunks and distributed over a simulated 3D cube using a mapping function. The ShiftCube operation executes programmed rotations based on the key, producing the ciphertext, which is then read and formatted through the ReadCipher process. On decryption, the UnShiftCube operation reverses the exact transformations, followed by the ReadPlain process to restore readable text. The Package KAM component handles message–key binding, ensuring the ciphertext and decryption key remain synchronized during transmission.

In a fully developed version, this implementation can exist as a browser extension, standalone application, or background service that runs alongside WhatsApp. The system can also integrate into WhatsApp Web using automation APIs or Electron-based frameworks, intercepting message payloads before they are sent. With its modular structure, RUENED can extend beyond WhatsApp and support multiple communication platforms while maintaining a lightweight and non-intrusive design. This approach provides a strong layer of cryptographic protection that is both innovative and efficient, merging the creativity of Rubik's Cube mechanics with practical cybersecurity needs.

5. Results and Discussion

The research depicts the Firefox module as a vital online defense integrating an encryption scheme, one based on Rubik's Cubes, which is an untried but optimally secure cyber defense methodology, taking advantage of enormous configurations of the Rubik's Cube. Users are able to encrypt and decrypt messages across the local browser, where communication privately and command cryptographic amenity themselves, as outside cryptographic intervention are discarded. Users are able to hold cryptographic keys themselves and govern the Rubik's Cube encrypt on the communications. Directly placed on the online WhatsApp are controls that, with the press of a single control, all text communication are instantly encrypted, while also heightening message protection and privacy.

Overall, while the RUEN Firefox Extension successfully implements a novel, Rubik's Cube–based encryption method, providing users with greater privacy and control over their messages, its immediate adoption is limited by practical barriers. Unlike established, native applications, its functionality is currently confined to a single platform (WhatsApp Web) and a single data type (text). To achieve widespread use, the project requires concentrated efforts in development, focusing on technical expansion to support features like media encryption and group key synchronization, which are essential for seamless real-world communication.

Comparing it with existing systems, the results are overall positive. The existing systems that we decided to

compare to are Signal Protocol.

a. Signal Protocol: The Signal Protocol is one of the most secure end-to-end encryptions (E2EE) frameworks in use today. It employs a combination of the Double Ratchet Algorithm, Curve25519, and AES-256 to ensure message confidentiality and forward secrecy.

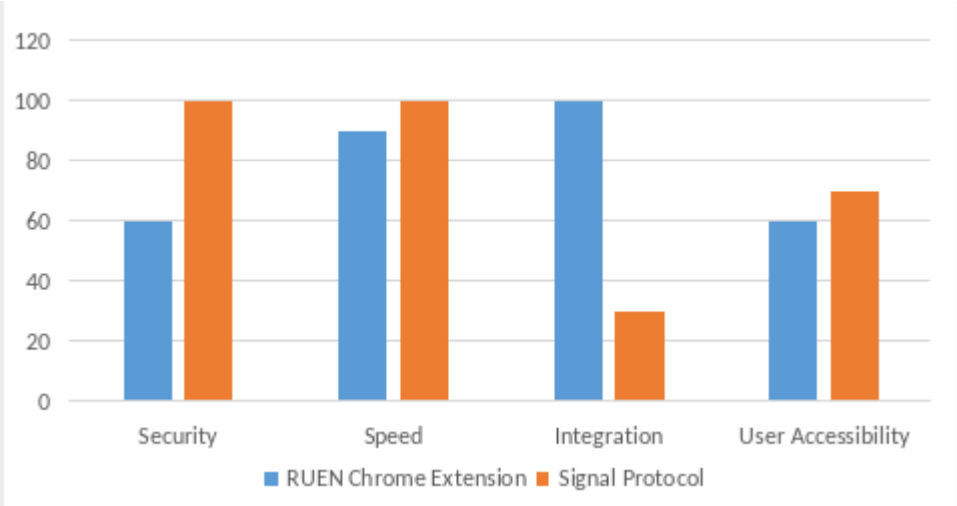

Figure 4: RUEN Firefox Extension VS Signal Protocol

6. References

[1]. Saxena et al., "Secure Resource Management in Cloud Computing: Challenges, Strategies and Meta-Analysis," IJCNS.org, 2025.

[2]. Barker et al., "Considerations for Achieving Crypto Agility: Strategies and Practices," NIST.gov, 2025.

[3]. Sadia Syed & Eid Mohammad Albalawi, "Optimizing Cloud Resource Allocation with Machine Learning: A Comprehensive Approach to Efficiency and Performance," ResearchGate, 2024.

[4]. Renata Teófilo de Sousa, Francisco Régis Vieira Alves & Ana Paula Florêncio Aires, "The Rubik's Cube and GeoGebra: A Visual Exploration of Permutation Groups," Revista do Instituto GeoGebra Internacional de São Paulo, 2024.

[5]. Swastik Kumar Sahu & Kaushik Mazumdar, "State-of-the-Art Analysis of Quantum Cryptography: Applications and Future Prospects," Frontiersin.org, 2024.

[6]. Manoj V. Bhagwath & A. Rengarajan, "Rubik's Encryption Combined with CNN for Biometric Authentication," IJIRCCE.com, 2024.

[7]. Aviral Srivastav & Aryaman Kumar, "A Robust Approach to Secure Data Encryption: AES RSA Hybrid with Kernel Key Protection," Research Square, 2023.

[8]. Daniel Salkinder, "A Group Theoretical Analysis of n × n × n Rubik's Cube," arXiv.org, 2021.

[9]. Abdalbasit Mohammed & Nurhayat Varol, "A Review Paper on Cryptography," ResearchGate, 2019.

[10]. Hena Rani Biswas, Md. Maruf Hasan & Shujit Kumar Bala, "Chaos Theory and Its Applications in Our Real Life," ResearchGate, 2018.

[11]. Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Anna Lubiw & Andrew Winslow, "Algorithms for Solving Rubik's Cubes," Springer, LNCS, 2011.