Hybrid AES–ECC and Steganographic Data Security Framework with Multi-Factor Authentication for Decentralized Storage

B. Rajesh Reddy[1*], Anem Gopi[2], Buddarthi Adithya[2], Pittala Kapildev[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering

[1,2]Kommuri Pratap Reddy Institute of Technology, Ghanpur, Ghatkesar, 501301, Telangana, India.

*Correspondence: B. Rajesh Reddy (rajeshreddy.reddy54@gmail.com)

ABSTRACT

The widespread adoption of centralized cloud infrastructures alongside decentralized storage paradigms like peer-to-peer networks and blockchain-driven platforms has resulted in user data being stored on remote systems outside direct ownership and control. Although encryption mechanisms are commonly implemented by service providers, the risk of data exposure persists due to insider threats, particularly from privileged administrators who may gain access to cryptographic keys and decrypt protected information. To address this concern, a self-protective data security architecture is proposed, combining hybrid Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC), image-based steganography, and Multi-Factor Authentication (MFA) to establish a secure end-to-end protection model. In this design, data is encrypted using AES to ensure computational efficiency, while ECC is utilized for secure key exchange, creating a layered encryption strategy where no single party can obtain complete access to both keys, thereby eliminating unauthorized decryption risks. Additionally, sensitive data is concealed using image steganography techniques, where encrypted content is embedded within normal-looking images, effectively masking its presence and making detection by adversaries highly improbable. Each file is further secured with a cryptographic hashcode to verify integrity, enabling immediate identification of any tampering attempts. User authentication is strengthened through MFA, requiring standard login credentials followed by a one-time password (OTP) sent to the user's registered email, significantly reducing the likelihood of unauthorized access even in the event of credential compromise. Furthermore, the framework integrates cybersecurity awareness modules that deliver educational content and real-time threat intelligence, encouraging users to adopt safer digital practices. By jointly leveraging strong encryption, covert data hiding, and rigorous authentication measures, this system ensures data confidentiality, integrity, and availability, while minimizing the risks associated with data breaches, unauthorized manipulation, and insider attacks in untrusted storage ecosystems.

Key words: Data Security, Hybrid Encryption, Advanced Encryption Standard (AES), Elliptic Curve Cryptography (ECC), Image Steganography, Multi-Factor Authentication (MFA), Insider Threat Protection

1. INTRODUCTION

In the present digital environment, the increasing number of cyber threats targeting organizations, governments, and individuals has created a strong demand for reliable information security mechanisms. Consequently, developers and researchers are continuously exploring advanced techniques to safeguard the privacy and confidentiality of data transmitted across communication networks. With the rapid migration of services from local infrastructures to the Internet, the importance of protecting information has grown significantly [1]. To keep pace with the advancing capabilities of cryptanalysis, encryption methods have been enhanced with greater complexity, making it more difficult for attackers to break secured data; however, encrypted information often draws attention from adversaries as a challenge to decode. In contrast, steganography reduces such attention by hiding the very existence of the data itself, thereby minimizing attack possibilities. Steganography is a method of

embedding secret data within a cover medium, and the term originates from the Greek language, meaning hidden or covered writing. Its primary goal is to conceal the presence of a message within another medium using various secret communication techniques [2]. Earlier techniques of steganography included methods such as microdots and invisible ink, whereas modern approaches utilize digital media formats like images, audio, and video files. The integration of steganography with encryption enhances security by ensuring that even if hidden data is detected, it remains unreadable, thereby confusing attackers and making extraction extremely difficult. This has led to extensive research aimed at improving data hiding techniques, particularly for secure communication of sensitive information. Recent trends also explore the use of steganography as an initial step in establishing secure communication channels instead of relying solely on predefined cryptographic keys or certificates. While steganography focuses on concealing data, steganalysis works to detect and extract hidden information, similar to how cryptanalysis functions in encryption systems. The objectives of steganalysis include identifying suspicious data carriers, determining the presence of hidden payloads, and retrieving embedded information. Therefore, the key challenges in designing effective steganography systems involve resisting detection techniques and ensuring that modifications remain imperceptible to the Human Visual System (HVS), especially in multimedia data [3].

In recent years, multi-level steganography techniques have gained attention for improving both hiding efficiency and payload capacity, which are critical factors in designing robust systems. Maintaining the visual quality of the stego image to closely match the original cover image is essential, as any noticeable distortion may increase the likelihood of detection. These requirements vary depending on user needs and the specific steganography method employed. This research focuses on developing an efficient and secure data hiding approach by combining steganography with encryption algorithms to ensure confidentiality and integrity. The proposed method aims to generate stego images with minimal perceptual differences, thereby preventing detection through visual or analytical attacks. Additionally, encryption techniques such as Advanced Encryption Standard (AES) and Blowfish are applied to secure the hidden data before embedding, ensuring that even if extracted, the information remains protected from unauthorized access [4].

Disinformation refers to intentionally false or misleading information that is deliberately spread. This study also addresses the challenge of detecting unauthorized alterations in hidden confidential data, while improving protection mechanisms against misuse. A verification layer is incorporated to ensure data integrity, enabling the detection of any modifications that deviate from the original content. The research ultimately aims to develop a highly effective data hiding system by integrating steganography with strong encryption techniques, ensuring secure transmission of sensitive information. Furthermore, by preserving high visual similarity between the cover and stego images, the approach minimizes the risk of detection through steganalysis or visual inspection. The inclusion of encryption algorithms such as Advanced Encryption Standard (AES) and Blowfish further strengthens the protection of the concealed message, preventing unauthorized access even in the event of extraction [5].

2. LITERATURE SURVEY

Khan et al. [6] introduced a GAN-based privacy-preserving scheme designed for hybrid cloud IoT environments. Their approach uses GAN to generate masks based on sensory data transformation values and trapdoor keys, ensuring that sensitive information is securely protected in private cloud environments. The system was evaluated using metrics such as PSNR, computation time, retrieval time, and storage overhead, demonstrating its ability to provide efficient and scalable privacy protection.

Budati et al. [7] proposed a flexible and adaptive data hiding algorithm designed to securely embed multiple secret messages while maintaining strong resistance against modern steganalysis attacks. Their approach generates an abstracted stego image during the embedding process, while also enabling

reconstruction of the original image, which enhances both reversibility and usability. A key strength of the method lies in its adaptability, as it can be applied to different image types, including high-dynamic-range (HDR) images and conventional low-dynamic-range color images. This flexibility allows the algorithm to be used across a wide range of real-world multimedia applications while maintaining high security and robustness against detection techniques. Lan et al. [8] presented an IoMT-based privacy-preserving model that is specifically designed to ensure secure handling of sensitive medical data with minimal computational overhead. Their approach integrates an efficient searching mechanism over encrypted data, allowing quick retrieval without compromising security. The encryption framework combines a modified Caesar cipher with ECDH and DSA, ensuring both secure key exchange and authentication. This hybrid approach not only protects patient data from unauthorized access but also enhances resistance to various cryptanalysis attacks. Additionally, the system is optimized for resource-constrained environments, making it suitable for IoMT applications where efficiency and security must coexist.

Abikoye et al. [9] proposed a security scheme that focuses on simplicity, efficiency, and robustness in protecting sensitive data. Their approach is designed to defend against multiple attack types, including statistical, differential, and brute-force attacks, which are commonly encountered in cryptographic systems. Through extensive experimental evaluation, the proposed method demonstrated superior performance compared to existing techniques, both in terms of computational efficiency and level of security. The balance between simplicity and strong protection makes this approach particularly suitable for practical implementations where resource constraints are a concern. Mfungo et al. [10] introduced an advanced steganographic technique that utilizes edge pixel-based data embedding to improve both capacity and security. The method begins by applying multiple edge detection algorithms, which are then combined using a logical OR operation to maximize the number of usable edge pixels. To further enhance embedding capacity, a morphological dilation process is applied to the detected edges. The system then computes both LSB and LBP codes for these pixels and combines them with secret data using XOR operations, ensuring secure and efficient embedding. This approach significantly increases resistance to detection while improving embedding efficiency. Sultana et al. [11] proposed a hybrid crypto-steganography technique for concealing ransomware data within high-resolution video frames. Their approach first encrypts the sensitive data using AES, ensuring confidentiality, and then embeds the encrypted data into video frames using LSB-based steganography. The embedding process is performed using a random pixel selection strategy, which increases unpredictability and reduces the likelihood of detection. This method effectively combines encryption and steganography to provide a dual layer of security, making it highly suitable for protecting sensitive multimedia data.

Almomani et al. [12] introduced a lightweight and noise-resilient steganography solution tailored for IoT environments, where devices often operate under limited computational resources and unstable communication conditions. Their approach evaluates the accuracy and reliability of hidden data under noisy conditions by using multiple modulation and coding schemes (MCS). To simulate real-world communication environments, AWGN is applied to the transmitted data. The system also considers various wireless communication technologies such as cellular networks, WiFi, and vehicular communication systems. This comprehensive evaluation ensures that the proposed method maintains data integrity even in challenging and noisy transmission conditions. Djebbar et al. [13] proposed a robust security framework aimed at enhancing the protection of banking and financial transactions. Their method integrates biometric authentication techniques, including face recognition and voice recognition, with image steganography to secure communication channels. Additionally, they introduced the use of Fibonacci sequences within a DSSS encryption system to improve randomness and security. A DWT-based embedding technique is also employed to hide sensitive data within digital images. This multi-layered approach ensures strong protection against unauthorized access and

enhances the overall reliability of financial data transmission systems. Tabirca et al. [14] explored the integration of artificial intelligence with cybersecurity to develop adaptive and intelligent defense mechanisms against evolving cyber threats. Their proposed Malicious Alert Detection System (MADS) combines adaptive learning techniques with a neighborhood-based voting mechanism to improve detection accuracy. The system is designed to address real-world challenges such as scalability and rapid threat detection, making it suitable for large-scale security applications. By leveraging AI, the model can continuously learn and adapt to new attack patterns, thereby improving overall system resilience.

3. PROPOSED SYSTEM

The increasing prevalence of cyber threats, unauthorized intrusions, and data privacy violations has made robust information security a critical requirement in modern digital systems, as conventional single-layer encryption approaches are no longer sufficient to defend against sophisticated and multi-vector attacks. To address these challenges, the proposed system introduces a comprehensive hybrid data security framework that integrates AES, ECC, image steganography, and MFA to provide multi-level protection, as illustrated in Fig. 1. The framework ensures end-to-end confidentiality, integrity, and authentication by combining fast symmetric encryption with secure asymmetric key exchange. Initially, the system enforces a secure user registration and authentication process, where OTP-based verification through an email server validates user identity and prevents unauthorized access. Upon successful authentication, users can upload sensitive data, after which an ECC-based public–private key pair is generated to establish a strong cryptographic foundation.

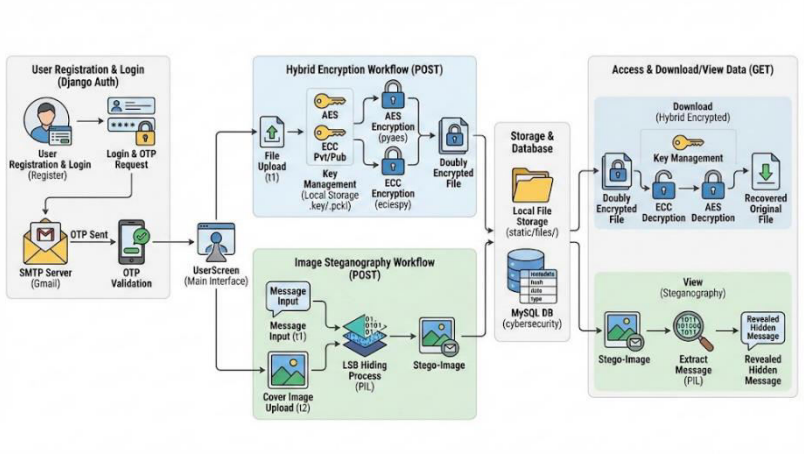

Fig. 1: System architecture

The uploaded data is encrypted using AES for high-speed and efficient data protection, while the AES key itself is encrypted using ECC to ensure secure key distribution and prevent key exposure. This hybrid encryption approach balances performance and security, making it suitable for real-time and large-scale applications. The encrypted data is then embedded into a cover image using the LSB steganography technique, which modifies pixel values in a way that is imperceptible to the human eye, thereby concealing the presence of sensitive information and adding an additional layer of protection against detection and interception. The resulting stego-image, along with encrypted metadata, is securely stored in the system using controlled access mechanisms supported by Django and managed through a MySQL database. Access control policies ensure that only authorized users can retrieve or manipulate stored data, thereby preventing unauthorized exposure even in the event of a system breach. When a user requests access to the protected data, the system performs a secure retrieval process in which the hidden encrypted content is first extracted from the stego-image, followed by decryption of the AES key using ECC, and finally decryption of the original data using AES. This step-by-step

decryption process guarantees accurate data recovery without any loss of integrity or confidentiality. Furthermore, the inclusion of MFA significantly enhances security by adding an additional verification layer beyond traditional credentials, effectively mitigating risks such as credential theft, brute-force attacks, and unauthorized logins. Overall, this multi-layered architecture not only strengthens data protection through encryption and concealment but also ensures secure access control, efficient processing, and resilience against modern cyber threats, making it a reliable solution for safeguarding sensitive information in contemporary applications.

3.1 STEGANOGRAPHY

Image Steganography is a technique used to conceal secret information within digital media such as images, audio, or video files without altering their visible appearance. Unlike encryption, which protects the content, steganography hides the existence of the message itself, making it difficult for attackers to detect. In image-based steganography, data is typically embedded into pixel values using methods like Least Significant Bit (LSB) modification. This approach ensures that the visual quality of the image remains almost unchanged while securely carrying hidden information. It is widely used in secure communication systems to provide an additional layer of protection alongside cryptographic techniques as illustrated in Fig. 2.

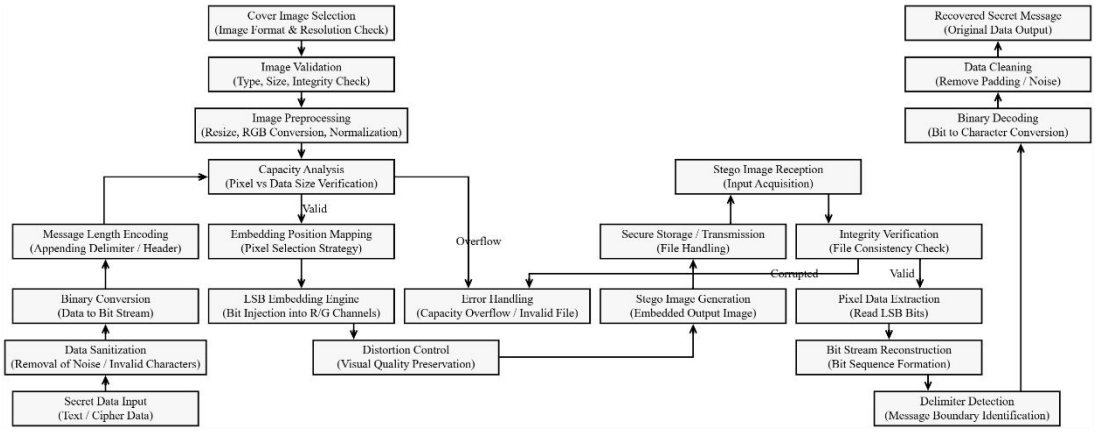

Fig. 2: Internal workflow of Steganography

Cover Image Selection and Preprocessing: The process begins by selecting a suitable cover image that will be used to hide the secret data. The image is analyzed for its size, resolution, and pixel capacity to ensure it can accommodate the hidden message. Proper preprocessing ensures minimal distortion and effective embedding.

Secret Data Preparation: The confidential message is prepared and converted into a binary format for embedding. This may include text, encrypted data, or any sensitive information that needs to be concealed. Converting the data into bits ensures compatibility with pixel-level manipulation.

Embedding Process (LSB Encoding): The binary data is embedded into the least significant bits of pixel values in the image. This modification causes negligible visual change, making the hidden data imperceptible to human eyes. The embedding process ensures that the data is distributed across the image securely.

Stego Image Generation: After embedding the secret data, a new image known as the stego image is generated. This image appears visually identical to the original but contains hidden information within its pixel structure. It can be safely transmitted or stored without raising suspicion.

Transmission and Storage: The stego image is transmitted over a communication channel or stored securely. Since the presence of hidden data is not obvious, it reduces the risk of interception or detection by unauthorized entities. This step ensures covert communication.

Extraction and Data Recovery: At the receiver's end, the hidden data is extracted by reversing the embedding process. The least significant bits are read and converted back into binary form to reconstruct the original message. This completes the secure data hiding and retrieval process.

4. IMPLEMENTATION DESCRIPTION

The system implementation integrates user authentication, encryption, steganography, and secure data storage within a Django-based web application. Users first register and log in to the system, where authentication is strengthened through an OTP verification mechanism sent to the user's registered email. Once authenticated, users can securely upload files that are protected using a hybrid encryption mechanism combining AES symmetric encryption and ECC for secure key management. The AES algorithm encrypts the file content, while ECC secures the encryption key to enhance overall data protection. In addition to encryption, the system supports image steganography, allowing secret messages to be hidden within image pixels using the LSB technique. All processed files are stored on the server, and a SHA-256 hash value is generated and stored in the database to maintain file integrity and detect tampering. When users request access to stored data, the system performs ECC and AES decryption to recover the original file before delivering it securely to the user. This integrated approach ensures confidentiality, integrity, and controlled access to sensitive information.

User Authentication and OTP Verification

- Separate user registration and login functionalities are implemented through Django view functions such as RegisterAction() and UserLoginAction(). These functions handle user input and manage authentication logic.
- User credentials including username, password, contact information, email, and address are stored in the MySQL database to maintain registered user records.
- During login, the system verifies the entered username and password by comparing them with the stored values in the register table.
- After successful credential verification, a random One-Time Password (OTP) is generated and sent to the user's registered email using the SMTP mail server.
- The user must enter the correct OTP through the OTPAction() function, after which a secure session is created to allow access to system features.

Database Integration using PyMySQL

- The application connects to the MySQL database using the pymysql.connect() function, enabling communication between the Django application and the database server.
- SQL queries are executed through cursor objects to perform operations such as inserting user records, verifying login credentials, and storing file details.
- The register table stores user information, while the files table stores uploaded file metadata such as filename, hash value, upload date, and security type.
- Database transactions are finalized using the commit() function to ensure that all inserted or updated records are permanently stored.

Hybrid Encryption using AES and ECC

- A hybrid encryption approach is implemented to secure uploaded files before storing them on the server.
- AES symmetric encryption is applied using the pyaes.AESModeOfOperationCTR() function to efficiently encrypt the file data.
- The AES encryption key is generated using the ecdsa.SigningKey.generate() method based on the SECP256k1 elliptic curve.
- After AES encryption, the encrypted data is further secured using Elliptic Curve Cryptography (ECC) through the encrypt() function provided by the ECIES library.

Elliptic Curve Key Generation and Management

- ECC public and private keys are generated using the generate_eth_key() function from the ECIES cryptographic library.
- The generated keys are stored in files such as pvt.key and pri.key so they can be reused securely without generating new keys each time.
- The public key is used during encryption operations, while the private key is used during the decryption process.
- This key management mechanism ensures secure encryption and controlled access to protected data.

Image Steganography using Least Significant Bit (LSB)

- Image steganography is implemented to hide confidential messages inside digital images without visibly altering the image appearance.
- The function generateBits() converts the secret message into binary bits before embedding them into image pixel values.
- The hideMessage() function embeds these bits into the least significant bits of the red and green color channels of image pixels.
- The hidden message can later be extracted using the extractMessage() function, which reconstructs the original text from the stored binary bits.

Secure File Storage and Integrity Verification

- After encryption or steganography, the processed file is stored on the server in the directory SecurityApp/static/files/.
- A SHA-256 hash value of the stored file is generated using the hashlib.sha256() function to maintain file integrity.
- The computed hash value, along with the filename, upload date, and security method, is stored in the database for verification purposes.
- This approach helps detect unauthorized modifications or tampering with stored files.

Secure File Retrieval and Decryption

- When a user requests to download a stored file, the system retrieves the encrypted file from the server storage.
- The encrypted file first undergoes ECC decryption using the private key to recover the AES-encrypted data.

- AES decryption is then performed using the generated symmetric key to restore the original file content.
- The decrypted file is finally returned to the user as a downloadable response using Django's HttpResponse mechanism.

5. CONCLUSION

The developed system effectively secures data transmission and storage by combining multiple protection mechanisms within a unified framework. It employs AES for efficient data encryption while utilizing ECC to ensure robust protection of encryption keys, forming a strong hybrid security model. The integration of steganography further conceals encrypted information within images, reducing the likelihood of detection by unauthorized entities. Data integrity is maintained through SHA-256 hashing, which helps verify that the content remains unchanged during operations. Additionally, OTP-based authentication strengthens access control by introducing an extra verification layer. Compared to conventional approaches, this system provides enhanced security along with improved performance and reliability. The implementation through Django offers a simple and user-friendly interface, enabling easy interaction while maintaining strong protection. Overall, the framework successfully achieves confidentiality, integrity, and secure authentication within a single platform.

REFERENCES

[1] Rizi, M.H.P.; Seno, S.A.H. A systematic review of technologies and solutions to improve security and privacy protection of citizens in the smart city. Internet Things 2022, 20, 100584.

[2] Saini, R.; Joshi, K.; Punyani, K.; Yadav, R.; Nandal, R.; Kumari, D. Interpolated Implicit Pixel-based Novel Hybrid Approach Towards Image Steganography. Recent Adv. Electr. Electron. Eng. (Former. Recent Patents Electr. Electron. Eng.) 2023, 16, 851–871.

[3] Vaishnavi, A.; Pillai, S. Cybersecurity in the Quantum Era-A Study of Perceived Risks in Conventional Cryptography and Discussion on Post Quantum Methods. J. Phys. Conf. Ser. 2021, 1964, 042002.

[4] Anthoniraj, S.; Karthikeyan, P.; Vivek, V. Weed Detection Model Using the Generative Adversarial Network and Deep Convolutional Neural Network. J. Mob. Multimedia 2022, 18, 275–292.

[5] Majeed, M.A.; Sulaiman, R.; Shukur, Z.; Hasan, M.K. A review on text steganography techniques. Mathematics 2021, 9, 2829.

[6] Khan; Khan, J.; Sehito, N.; Mahmood, K.; Ali, H.; Bari, I.; Arif, M.; Ghoniem, R.M. An Adaptive Enhanced Technique for Locked Target Detection and Data Transmission over Internet of Healthcare Things. Electronics 2022, 11, 2726. https://doi.org/10.3390/electronics11172726.

[7] Budati; Vulapula, S.R.; Shah, S.B.H.; Al-Tirawi, A.; Carie, A. Secure Multi-Level Privacy-Protection Scheme for Securing Private Data over 5G-Enabled Hybrid Cloud IoT Networks. Electronics 2023, 12, 1638. https://doi.org/10.3390/electronics12071638.

[8] Lan, C.-F.; Wang, C.-M.; Lin, W. A Novel Adaptive Image Data Hiding and Encryption Scheme Using Constructive Image Abstraction. Appl. Sci. 2023, 13, 6208. https://doi.org/10.3390/app13106208.

[9] Abikoye; Oladipupo, E.T.; Imoize, A.L.; Awotunde, J.B.; Lee, C.-C.; Li, C.-T. Securing Critical User Information over the Internet of Medical Things Platforms Using a Hybrid Cryptography Scheme. Future Internet 2023, 15, 99. https://doi.org/10.3390/fi15030099.

[10] Mfungo; Fu, X. Fractal-Based Hybrid Cryptosystem: Enhancing Image Encryption with RSA, Homomorphic Encryption, and Chaotic Maps. Entropy 2023, 25, 1478. https://doi.org/10.3390/e25111478.

[11] Sultana; Kamal, A.H.M.; Hossain, G.; Kabir, M.A. A Novel Hybrid Edge Detection and LBP Code-Based Robust Image Steganography Method. Future Internet 2023, 15, 108. https://doi.org/10.3390/fi15030108.

[12] Almomani; Alkhayer, A.; El-Shafai, W. A Crypto-Steganography Approach for Hiding Ransomware within HEVC Streams in Android IoT Devices. Sensors 2022, 22, 2281. https://doi.org/10.3390/s22062281.

[13] Djebbar Securing IoT Data Using Steganography: A Practical Implementation Approach. Electronics 2021, 10, 2707. https://doi.org/10.3390/electronics10212707.

[14] Tabirca; Dumitrescu, C.; Radu, V. Enhancing Banking Transaction Security with Fractal-Based Image Steganography Using Fibonacci Sequences and Discrete Wavelet Transform. Fractal Fract. 2025, 9, 95. https://doi.org/10.3390/fractalfract9020095.