International Journal of
Advanced Science Computing and Engineering

Journal Homepage www.ijasce.org/index.php/ijasce

PrasathSrinivasan Sushmitha: Algorithm for String Encryption and Decryption

J.S.Prasath J.Srinivasan [a], Sushmitha G.S [a]

[a] *Department of Computer Science and Engineering, Sapthagiri NPS University, Bengaluru, India*

Corresponding author: jsprasath@gmail.com

***Abstract*—Security is essential for communication of messages, data, and private information. Security threats and vulnerabilities increase rapidly due to the wide usage of wireless media for data transmission. Intruders can access and alter sensitive data and private information transmitted across the Internet. Security algorithms are important to ensure secure communication without unauthorized access or modification. This work proposes a novel cryptography scheme for encrypting text messages. This proposed algorithm encrypts the input string into ciphertext through a series of mathematical operations. The inverse operation of encryption is performed at the receiver to obtain the original string from the ciphertext. This proposed cryptography can be utilized for preventing theft and fraud in online money transactions, securing email messages, file encryption, securing web information, ensuring data confidentiality across the internet, assuring data security in remote access, securing the patients' information transmitted across wireless networks, enabling secure communication of the country's border information, preventing credit card fraud, securing the industrial continuous varying process parameters, encryption of WhatsApp messages, and securing the individual's private information.**

***Keywords*—Cryptography; encryption; decryption; security; string.**

Manuscript received 12 Dec. 2025; revised 9 Feb. 2026; accepted 20 Apr. 2026. Date of publication 31 Aug. 2026.

I. INTRODUCTION

Cryptography is closely related to cryptology and cryptanalysis. Cryptology is the concept that involves both cryptography and cryptanalysis. Cryptology includes a wider scope than cryptography. It also includes examining numerical structures, computations, and conjectural cryptographic properties. Cryptology reads the input plaintext and converts it into equivalent ciphertext as well as reading the ciphertext and converting it into plaintext. Cryptology involves encrypting and decrypting data. It also analyzes and breaks existing encryption algorithms. This paper presents an expanded cryptographic ID mechanism based on the Compact Knapsack problem [1]. This scheme ensures security against active attacks. Cryptography is widely used in our daily applications, including financial transactions, medical services, legal operations, and electronic banking systems in order to maintain data secrecy and protect private information from various attacks. It is used in ATM (Automated Teller Machines), where customers can deposit or withdraw money using the personal identification number (PIN). The bank stores the PIN ciphertext in its database and on credit and debit cards. This type of PIN storage is called one-way cryptography. The bank generates the ciphertext using the customer's PIN and its key. Mathematically, it is not possible to recover the plaintext PIN from the ciphertext even though the key is known. The encryption algorithm should be as simple as possible, perform fewer computations, and consume less power.

A low-energy data encryption algorithm based on the Advanced Encryption Standard (AES) is proposed to enhance security in data communication and reduce power consumption during encryption [2]. This data encryption algorithm enhances the performance of Internet of Things (IoT) devices. Assessing Elliptic Curve Cryptography (ECC) methods helps ensure security and data privacy in IoT-based devices [3]. Researchers should consider a variety of security threats and concerns. The optimal secure defense mechanism for Distributed Denial of Service (DDoS) attacks in IoT networks utilizes feature optimization and an intrusion detection system [4]. An improved quantum query optimization algorithm selects the optimal feature among many features, reducing the data dimensionality problem.

Figure 1 shows the components of cryptology. Cryptology involves different ways to convert plaintext into unreadable ciphertext and different ways to break ciphertext and identify

the plaintext. Cryptology deals with secure message communication and confidential data storage.

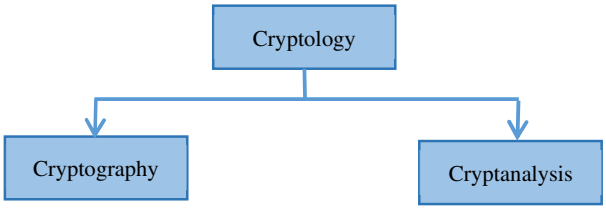

Fig. 1 Components of Cryptology

Cryptography protects messages and communications so only authorized parties can access confidential information transmitted across the internet. Today, cryptographic methods are designed to be unalterable, ensuring data security. Developing strong cryptography techniques is essential to secure and protect information transmitted across the medium from various attacks. Conventional cryptographic methods are less efficient, and attackers can easily hack messages and alter data. Modern cryptographic algorithms involve complex mathematical computations, and intruders may need years or even decades to identify the exact data, information, or even a single message. Researchers are still working to develop efficient, optimized, and strong encryption algorithms as the variety of attacks increases. A study by [5] addresses preferred security properties and existing attacks against the Industrial IoT. This work discusses traditional cryptographic tools used to secure recent developments in Industrial IoT networks. Another study by [6] segregates various security mechanisms, including symmetric encryption, asymmetric encryption for signatures, and access control. It provides data security and protects information gathered by Wireless Sensor Networks.

An optimized hybrid encryption algorithm is proposed that integrates Elliptic Curve Cryptography with the Advanced Encryption Standard [7]. It strengthens data security and efficiency, and it is used for smart home healthcare systems. Research on quantum communication and cryptography aims to develop strong encryption techniques that defend against attacks from quantum computers [8]. The study explores quantum technology in different applications, including quantum machine learning and quantum blockchain. The trends in lightweight cryptography and various encryption algorithms are compared using real-time data [9].

The key parameters taken for consideration are encryption time, memory utilization, and CPU utilization. Two Dynamic Searchable Symmetric Encryption algorithms are proposed to achieve forward and Type-1 backward security while maintaining strength when incoherent queries are issued [10]. This work reduces the information leakage of encryption. Various cryptographic algorithms, such as elliptic curve, hybrid, lightweight, and novel techniques, are evaluated [11]. Elliptic Curve Cryptography (ECC) is suitable for secure communication and is a choice for lightweight cryptography for Internet of Things (IoT) devices. A secure and effective cloud-based data-sharing system is proposed [12]. The dual timestamp management scheme introduces timestamp management in each ciphertext to support dynamic user groups. The framework is proposed to accomplish privacy-preserving statistical investigation on an encrypted database [13].

A cryptosystem based on binary vectors is proposed to accomplish complex logic expressions for statistical analysis on ciphertext. The ciphertext evaluation mechanism is designed, which permits the edge to clean the encrypted value before uploading [14]. The public-key encryption algorithm is investigated to implement the secure data evaluation mechanism. A multi-key searchable encryption scheme is proposed that is efficient and secures a data search algorithm, allowing the owner to grant users the ability to retrieve data from the ciphertext [15]. This novel multi-key data search mechanism is strong against unauthorized queries. An updated version of the Menezes-Vanstone elliptic curve cryptography scheme is proposed to secure text, image, audio, and video data [16]. It is inferred that the ciphertext file size is smaller than alternative methods. The novel method is proposed for creating stochastic and undeterminable keys for symmetric One-Time Pad cryptosystems [17].

The publicly available DNA sequences stored in genetic databases are used as a source. This work discusses the encryption stage of the cyber kill chain and its detection methods [18]. The system calls, I/O monitoring, and file system operations are performed using encryption-related activities. The characteristics of the homomorphic transform are utilized to make it more suitable for text encryption [19]. The simple union and symmetric difference operators are used for converting plaintext to ciphertext. A novel cryptography has been formulated that uses both Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC), along with clustering through the standard LEACH protocol, to enhance energy efficiency, data security, and network lifetime [20]. It addresses key-exchange issues, is simpler than ECC, and is more reliable than AES.

Cryptography protects information and communications using codes so that only those for whom the information is intended can read and process it. Cryptography involves procedures that use mathematical calculations called algorithms. Figure 2 shows that the input plaintext is converted into ciphertext, which is called as encryption. Cryptography converts input messages into unreadable text that is difficult for attackers to decipher or use to obtain the original information. These algorithms are used to keep private data secret, secure web information, and protect credit and debit card data and transactions.

Fig. 2 Cryptography

The objectives of cryptographic algorithms are confidentiality, integrity, non-repudiation, and authentication. The raw data and the information are encrypted to produce the ciphertext. This ciphertext is unreadable, and it assures confidentiality of information during transmission over the internet. Cryptography also ensures data integrity, so the information cannot be modified during communication. Cryptography also provides non-repudiation, meaning the sender cannot deny sending the information or their intent in transmitting it. Non-repudiation is a sequential, judicial concept that proves the genuineness of information or data communication by providing indisputable proof of both authenticity and integrity.

Cryptography provides message authentication, which involves identifying and validating the information sender. Authentication is the process of identifying the individuals who create and send information. Authentication mechanisms enable access control for devices by testing the user's information matches the information given in the database of authorized parties or a data authentication server. Authentication ensures that devices, operations, and organizational information remain confidential. User authentication is achieved by validating the user ID and password. In addition, users need to authenticate through face recognition, thumbprint, or biometric signature.

The proposed security scheme uses Improved Elliptic Key Cryptography to encrypt and decrypt data [21]. The source code stores private keys and other keys in non-volatile memory, and it prevents replay, Denial of Service, and Sybil attacks. The optimized quantum network integrates the Whale optimization algorithm with feed-forward and backpropagation algorithms [22]. The administrator keeps sensitive data on a server, and the document is encrypted using the encryption technique. An enhanced S-box-based Advanced Encryption Standard is proposed, along with a Runge-Kutta Optimization scheme, to ensure the confidentiality and integrity of medical information [23]. The level of security is validated using non-linearity, the Strict Avalanche Criterion, Differential Probability, the Bit Independence Criterion, and linear probability parameters.

The lightweight elliptic curve cryptographic algorithm for safeguarding resource-constrained devices such as IoT is proposed [24]. Elliptic curve cryptography provides higher security than the RSA algorithm for the same key size. A groundbreaking hybrid cryptographic scheme is proposed for securing data storage in cloud computing [25]. This algorithm uses a variety of features, including time-limited access control, adaptive key management, and dual security algorithms: RSA and Advanced Encryption Standard. Enhanced elliptic curve cryptography and chaotic mapping are proposed to improve data transmission security [26]. Cryptography and steganography are united to improve data security. The provable Fuzzy multi-keyword search mechanism is proposed along with adaptive security [27]. It uses locality-based hashing to hash incorrect words and map correct keywords to the same point. [28] proposes an effective privacy-preserving scheme for ciphertext traffic detection. This mechanism utilizes lightweight cryptographic operations to attain both privacy and security.

[29] proposes a broad range of cryptographic algorithms designed to protect sensitive information in cloud computing. The research addresses cryptographic schemes in cloud computing. Symmetric encryption is designed to protect cloud server information and to secure the transmission and reception of cloud server data [30]. A dual-encryption algorithm is implemented to transmit data securely.

Cryptography is classified into symmetric-key, asymmetric-key, and hash functions. In symmetric-key cryptography, the transmitter and receiver use the same key to encrypt and decrypt data or text messages. Symmetric-key mechanisms are faster and less complex to use, but the major challenge is securely transmitting the key between the transmitter and receiver. Commonly used symmetric-key cryptography includes the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES). Asymmetric key cryptography uses two keys for encrypting and decrypting data and messages. The sender utilizes the public key to convert the input plaintext into unreadable ciphertext. The receiver uses the private key to convert the ciphertext back into plaintext. This asymmetric key encryption strengthens security through a key pair. Commonly used asymmetric cryptography algorithms include RSA, Elliptic Curve Cryptography (ECC), and the Secure Shell Protocol (SSH). The algorithms and computational complexity of asymmetric cryptography is higher than those of symmetric cryptography.

Cryptography is also used to ensure data integrity. Hash functions ensure data integrity. Hash functions are a kind of cryptographic algorithm that creates a finite-length hash value. A hash algorithm reads the set of input data and converts it into a specific hash value. Hash functions are more effective since the variation of one character or a blank space in the plaintext would generate a distinct value. Applications, websites, or receivers can test data integrity by comparing the obtained hash value with the standard hash, and they can validate that the data has not been modified during communication. Hash functions are also widely used to protect user passwords without creating an insecure client-side database of confidential passwords. The online banking system only gathers and stores the hash value of customer passwords. If an intruder steals the bank's customer database, they should not be able to retrieve any passwords from the hash values.

The main purpose of cryptography is to protect the data, information, and messages during transmission across the internet. Cryptographic protocols can provide security between web browsers and web servers. Data transmission between a browser and a website occurs through a secure channel, so attackers cannot eavesdrop on the information. Cryptography is also used to secure email and WhatsApp communication. It provides end-to-end encryption and keeps users' data private. The importance and maintenance of data secrecy is addressed [31]. The paper discusses various security algorithms and compares their parameters. The paper addresses opinions on cryptography, its history, and modern cryptography [32]. The paper explores the computations involved in cryptographic algorithms. A modern hardware security component is proposed for cryptographic key generation [33]. The designed hardware module removes stored cryptographic keys and avoids attacks against them. The dynamic QR code payment system is presented to rectify the security issues [34].

The hardware section of the security component describes the algorithm flow and improves payment performance. This work discusses the security schemes used internally in the Android Operating System [35]. This work investigates security mechanisms and validates cryptographic schemes for energy efficiency based on computation time. The forward search privacy scheme proposes strengthening security [36]. It involves search operations over recently added documents that do not leak any data about past queries. A flexible lightweight encryption algorithm is designed and implemented [37]. It uses simple substitution and transposition to encrypt and decrypt data, requiring fewer computations on IoT devices.

A modern lightweight cryptographic algorithm is proposed for improving data security in cloud computing [38]. This algorithm is based on a 128-bit block cipher and uses a 128-bit key for data encryption. A smart residential load management scheme is designed in three parts, along with consumer security provisioning [39]. The computation of home load confirms how to manage the load, either to switch off the load or indicate extra usage of electricity. The privacy protection mechanism is proposed using an improved Honeypot algorithm [40]. It maintains data security against intrusion or other attacks.

Cryptanalysis is the art and science of retrieving the plaintext from the ciphertext. It analyzes weaknesses in cryptographic algorithms, and the plaintext is identified using these weaknesses without knowledge of the secret key. Figure 3 shows the input ciphertext being converted into plaintext. It reads the ciphertext, analyzes it, and recovers the plaintext without using the key.

Fig. 3 Cryptanalysis

Cryptanalysis is the process of analyzing and decoding ciphers, codes, and encrypted data without using the real key. A cryptanalyst tries to crack or break the encryption codes to identify the plaintext. The major job and responsibilities of a cryptanalyst is to gather, process, and analyze information in ciphertext, perform debugging of software code, examine the deficiencies of cryptographic algorithms, develop modern tools for cryptanalysis, and develop techniques for exploiting shortcomings in computer networks. Cryptanalysts serve in various sectors including government, private organizations, legal and criminal cases, academic institutions, banking and finance, military, and medical fields. Cryptanalysts should have vast knowledge of advanced mathematical operations, encryption processes, programming languages, and data structures.

Public sectors use cryptanalysis to decode encrypted information from other countries. Industries and enterprises use cybersecurity goods and services to evaluate their security features. Usually, cyber-terrorists use cryptanalysis to uncover cryptosystem vulnerabilities rather than launch a brute-force attack. Hackers are often categorized as black-hat or white-hat. Black-hat hackers use cryptanalysis to commit cybercrimes, whereas white-hat hackers use cryptanalysis to perform penetration testing and verify security strength. The research focuses on cybersecurity related to big data [41]. The research addresses big data tools and their protection for cybersecurity. A modified security algorithm, together with a dedicated hardware key using an embedded system, is proposed [42]. This algorithm produces a 1024-bit key size for asymmetric encryption and a 256-bit key size for symmetric encryption. An identity-based encryption transformation scheme is proposed that combines two well-established encryption algorithms: identity-based encryption (IBE) and identity-based broadcast encryption [43]. A concrete identity-based encryption transformation is designed which is based on bilinear groups and verify its security against powerful attacks. A secure ciphertext duplication mechanism based on data popularity is proposed [44]. Ciphertext policy attribute-based encryption is applied to preserve the tags. The homomorphic encryption encapsulated difference expansion technique is proposed for reversible data hiding in ciphertext [45]. Key-switching and bootstrapping methods are presented to control the ciphertext expansion and decryption failure of homomorphic encryption. The security mechanism is implemented for wastewater monitoring using an embedded system and the internet [46]. An embedded system encrypts dissolved oxygen and pH values and sends them through a wireless medium. An experimental and numerical approach to optical key distribution using quantum cryptography with the BB84 protocol has been achieved [47]. The security of quantum cryptography is improved using a one-time pad scheme and chaotic signal. The key generation algorithms of conventional cryptography and the need for strong security are discussed [48]. The different ways of attaining rigid security in public-key cryptography is addressed. The selection of identity-based asymmetric encryption is addressed [49]. This identity-based cryptography offers end-to-end information security across IoT-enabled industrial operations. The modern encryption scheme based on coupled map lattice is proposed for image security [50]. This encryption method utilizes a randomly generated secret key, sub-key-based substitution, and a confusion algorithm to improve the security, sensitivity, and robustness. This modern technique is incorporated to secure data storage [51]. In this scheme, biometric-based statistical features are explored to create a user code word.

II. Materials and Method

A. *Cryptanalytic Attacks*

Cryptanalysis is categorized into three types: analysis of ciphertext alone, analysis of known ciphertext/plaintext pairs, and analysis of selected plaintext or selected ciphertext. A cryptanalytic attack aims to identify weaknesses in cryptographic algorithms. This attack depends on the quality of the algorithms and the intelligence available about plaintext characteristics. The plaintext can be a string, a group of strings, or code written in any programming language. Knowledge of the plaintext is essential before testing an attack. Figure 4 shows the various types of cryptanalytic attacks. These include known-plaintext attacks, chosen-plaintext attacks, ciphertext-only attacks, man-in-the-middle attacks, and adaptive chosen-plaintext attacks. Other attacks include birthday attacks, side-channel attacks, brute-force attacks, dictionary attacks, hybrid attacks, reverse brute-force attacks, credential stuffing, and differential cryptanalysis attacks.

B. *Known-Plaintext Attack (KPA)*

This attack type is based on known plaintext-ciphertext pairs. Intruders map plaintext to ciphertext to identify the secret key. Intruders examine correlations between known plaintext and ciphertext to identify patterns or consistencies that might expose the key or the algorithm. Attackers can easily gather the known data and messages. Attackers can use frequency analysis to determine the most repeated characters or symbols in the ciphertext and match them to their corresponding plaintext.

C. Chosen-Plaintext Attack (CPA)

This attack selects random plaintext, retrieves the related ciphertext, and tries to determine the secret key. This attack is less complex than known-plaintext attacks, but it has a low success rate.

D. Ciphertext-Only Attack (COA)

Attackers try to identify the secret key and the plaintext from the known ciphertext. The success rate of this attack depends entirely on the intruders' cryptanalytic knowledge. This type of attack is difficult to implement, more time-consuming, and requires attackers to test many combinations to determine the plaintext.

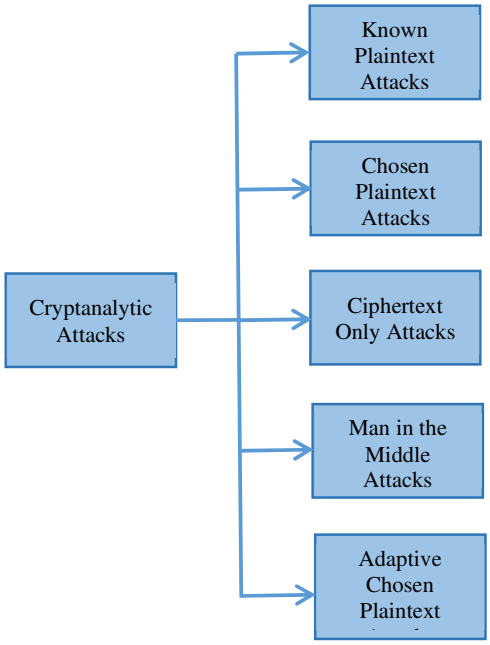

Fig. 4 Types of Cryptanalytic Attacks

E. Man-In-The-Middle (MITM) attack

This type of attack allows intruders to listen to information in communication between the sender and receiver through a secured channel. An attacker may create a fake website between the customer and the bank's actual webpage. The various types of Man-in-the-Middle attacks include eavesdropping on messages transmitted through Wi-Fi, hacking of electronic mail messages, spoofing of Internet Protocol (IP), Hypertext Transfer Protocol Secure (HTTPS), Domain Name System (DNS), Address Resolution Protocol (ARP), and removal of Secure Socket Layer (SSL).

MITM attacks are difficult to detect without utilizing suitable mechanisms. Authentication and authorization are essential to detect attacks, and they require additional investigation. Organizations and enterprises should have standard procedures and preventive measures to nullify MITM attacks. Organizations can mitigate MITM attacks by encrypting the Wireless Access Point (WAP), authenticating the public key, ensuring data security during communication, avoiding public Wi-Fi, using a Virtual Private Network (VPN), and ensuring network security through an intrusion detection system.

F. Adaptive Chosen-Plaintext Attack (ACPA)

Attackers choose the plaintext and encrypt the plaintext multiple times. The plaintext is divided into smaller texts and encrypted to obtain the ciphertext. Based on the ciphertext, select another text for encryption, and this process is repeated until all plaintext is converted into ciphertext. In addition to these cryptanalytic attacks, intruders can use other attacks. These are birthday attacks, side-channel attacks, brute-force attacks, and differential cryptanalysis attacks.

G. Birthday Attack

This attack uses the probability that two or more people in a group share the same birthday. The birthday attack is a cryptographic attack based on the birthday paradox. This attack identifies collisions in a hash function.

H. Side-Channel Attacks

This type of attack exploits information leakage from a physical cryptosystem. Intruders gather information by analyzing indirect information, such as power consumption, electromagnetic leaks, or even sound, to reveal sensitive data like cryptographic keys or personal information. Common side-channel attacks include timing attacks, power analysis attacks, electromagnetic attacks, and others.

I. Brute-Force Attacks

Attackers use trial and error to identify login details, passwords, and secret keys. Intruders test passwords randomly until they find the correct one. A brute-force attack is a simple way to gather secret information, and it can be highly successful. This type of attack takes more time to guess a password, and it becomes more complex and expensive with a longer key length. Hackers can easily access private information because many people create weak passwords. Many users create passwords without combining numbers, special characters, and uppercase and lowercase letters because they are difficult to remember.

Brute-force attacks take different forms. A simple brute-force attack occurs when the hacker maintains a list of usernames and tries to guess the passwords for different usernames. The attacker repeats this process until they retrieve the exact username and password combination. Brute-force attacks include dictionary attacks, hybrid attacks, reverse brute-force attacks, and credential stuffing.

J. Dictionary Attacks

A dictionary attack is a type of brute-force attack in which the intruder uses public, easily recognizable words and phrases from a dictionary to guess passwords and personal identification numbers (PINs). Users often choose simpler password combinations and passwords that are easy to remember. This makes dictionary attacks easier, as skilled attackers can quickly identify simpler passwords. But dictionary attack efforts may be unsuccessful where individuals have a sophisticated set of passwords and not use mere alphabets or numbers as their passwords. The effects of dictionary attacks can be reduced by frequently changing passwords and using two-factor authentication. It can also be prevented by using strong passwords that include a random combination of lowercase and uppercase letters, numbers, and special characters.

K. Hybrid Attacks

This attack combines conventional brute-force attacks with dictionary attacks. Attackers maintain a list of known words and try to match the combinations. This attack starts with dictionary words and then adds numbers, special characters, or changes letter cases. It also considers patterns such as appending or prepending numbers or symbols to dictionary words. Hybrid attacks are faster than brute-force attacks because they narrow down the possibilities using common password variations. Hybrid attacks are effective because people often use common words or patterns as passwords. You can prevent hybrid attacks by using strong passwords that include numbers, special characters, and uppercase and lowercase letters. Also, add multi-factor authentication to strengthen security. Implement a lockout scheme after a set number of failed attempts.

L. Reverse Brute Force Attacks

A reverse brute-force attack is a type of brute-force attack in which an intruder uses a common password against multiple usernames to gain access to a network. The objective is to gather user account details without authorization by forcing the same password for all users. Reverse brute-force attacks often target enterprises by identifying account names, leaked account databases, or commonly accessible account lists. Reverse brute-force attacks start with the intruder knowing the password but not the username. Attackers verify these passwords against several possible usernames or encrypted files until they recover the exact combination. The best way to prevent reverse brute-force attacks is to keep passwords secure. Industries should have standard rules and policies for creating, updating, and securing passwords. Enterprises can also incorporate two-factor authentication or multi-factor authentication.

M. Credential Stuffing

Credential stuffing is a cyberattack in which cybercriminals use hijacked login details from one system to access an unrelated system. Credential stuffing attacks rely on the assumption that people reuse the same user ID and password across multiple accounts. Maintaining the login details of one account may provide access to other unrelated accounts. When an illegitimate party obtains the exact username and password, credential stuffing enables a quick attack to log in to other accounts that may use the same user data.

N. Differential Cryptanalysis Attack

This attack involves examining pairs of plaintexts and their related ciphertext to identify patterns in the security algorithm. It can be efficient against block ciphers with specified characteristics.

O. PrasathSrinivasan Sushmitha Algorithm for Text Encryption and Decryption

This proposed novel encryption and decryption method is named the PrasathSrinivasan Sushmitha algorithm. This proposed algorithm encrypts text through a series of mathematical operations. The algorithm describes the steps involved in text encryption. The text encryption reads the input string and converts it into ciphertext.

P. PrasathSrinivasan Sushmitha Encryption Algorithm

1. Step 1: Read the input text.
2. Step 2: Assign the numbers from ‘1’ to ‘26’ to the letters in uppercase from ‘A’ to ‘Z’ or in lowercase letters from ‘a’ to ‘z’ and write the corresponding number for the given input string.
3. Step 3: Compute the cube of each decimal digit.
4. Step 4: Represent the cubed decimal digit in its equivalent Binary Coded Decimal (BCD) form.
5. Step 5: Compute the 1’s complement of the obtained BCD.
6. Step 6: Convert the resultant 1’s complement into equivalent hexadecimal.
7. Step 7: Represent the hexadecimal value in the form of a square matrix. The number of rows and columns in the square matrix depends on the number of digits obtained in hexadecimal.
8. Step 8: After forming the square matrix, if there is any unfilled elements, assign the value as zero.
9. Step 9: Exchange diagonal elements.
10. Step 10: Shift rows towards the bottom one time for a 2x2 matrix, shift rows twice towards the bottom for a 3x3 matrix, shift rows bottom three times for a 4x4 matrix, shift rows bottom four times for a 5x5 matrix, and so on.
11. Step 11: Shift columns one time towards the left for a 2x2 matrix, twice towards the left for a 3x3 matrix, shift columns left three times for a 4x4 matrix, shift columns left four times for a 5x5 matrix, and so on.
12. Step 12: Represent each element in its equivalent BCD form.
13. Step 13: The resultant BCD value is the ciphertext.

The initial step of this proposed encryption algorithm is to read the string. The alphabet in lowercase letters from ‘a’ to ‘z’ or in uppercase letters from ‘A’ to ‘Z’ is assigned the values from numbers ‘1’ to ‘26’, respectively. The assigned numbers for the alphabet are to be written separately for the given input string. Next, compute the cube of each decimal number. After cubing, represent the result in Binary Coded Decimal (BCD) form. The 1’s complement is computed for this obtained BCD value. The value obtained after 1’s complement is converted into a hexadecimal value. Represent this hexadecimal value in a square matrix. The number of rows and columns depends on the number of hexadecimal digits. For unfilled elements in the square matrix, assign the value as zero and exchange the diagonals of the square matrix. Then, shift rows towards the bottom one time for a 2x2 matrix, shift rows twice towards the bottom for a 3x3 matrix, shift rows three times towards the bottom for a 4x4 matrix, and so on. After shifting the rows, shift the columns once to the left for a 2x2 matrix, twice to the left for a 3x3 matrix, three times to the left for a 4x4 matrix, and so on. After shifting rows and columns of the square matrix, represent each element in equivalent BCD form. The BCD value obtained is the ciphertext.

Q. Prasath Srinivasan Sushmitha Decryption Algorithm

1. Step 1: Receive the ciphertext in BCD form.
2. Step 2: Convert the BCD value to equivalent hexadecimal.
3. Step 3: Represent the hexadecimal values in the form of a square matrix.
4. Step 4: Shift columns one time to the right for a 2x2 matrix, shift columns twice to the right for a 3x3 matrix, shift columns right three times for a 4x4 matrix, shift columns right four times for a 5x5 matrix, and so on.
5. Step 5: Shift rows one time towards the top for a 2x2 matrix, shift rows twice towards the top for a 3x3 matrix, shift rows top three times for a 4x4 matrix, shift rows top four times for a 5x5 matrix, and so on.
6. Step 6: Exchange diagonal elements.
7. Step 7: Discard the last element if it is zero.
8. Step 8: Compute 1's complement for the obtained hexadecimal value.
9. Step 9: Represent the resultant 1's complement in its equivalent decimal number.
10. Step 10: Compute the cube root of the decimal value separately.
11. Step 11: Assign the equivalent alphabet from the resultant cube root separately.
12. Step 12: Combine the alphabets to form the string which is the original input text.

The decryption is the reverse process of encryption. The ciphertext is received in BCD form. Convert this BCD value into an equivalent hexadecimal value. Next, represent this hexadecimal value as a square matrix. In this square matrix, shift columns one time to the right for a 2x2 matrix, shift columns twice to the right for a 3x3 matrix, shift columns right three times for a 4x4 matrix, shift columns right four times for a 5x5 matrix, and so on. Next, shift rows up once for a 2x2 matrix, twice for a 3x3 matrix, three times for a 4x4 matrix, four times for a 5x5 matrix, and so on. After shifting columns and rows, exchange the diagonal elements. Discard the last element if it is zero. Compute the 1's complement of the obtained hexadecimal value and represent it as an equivalent decimal value. Then compute the cube root of each decimal value. Next, assign the equivalent alphabet to each resultant cube root and combine the alphabets to form the original input text.

III. Result and Discussion

This proposed novel encryption and decryption algorithm is simulated using Python. This algorithm involves a series of mathematical operations, including conversion, complement, shifting rows and columns of the matrix, and representing the ciphertext in BCD format. Figure 5 shows the Python output of the proposed cryptographic algorithm with the input string named "KARTHIKEYAN" in capital letters. The proposed encryption algorithm produces an encrypted text consisting of zeros and ones, as shown in the output. The proposed decryption algorithm produces "KARTHIKEYAN", as shown in the output, which matches the input string. The input characters and the decrypted text are all in capital letters. This proposed cryptographic algorithm encrypts and decrypts the input string with capital letters.

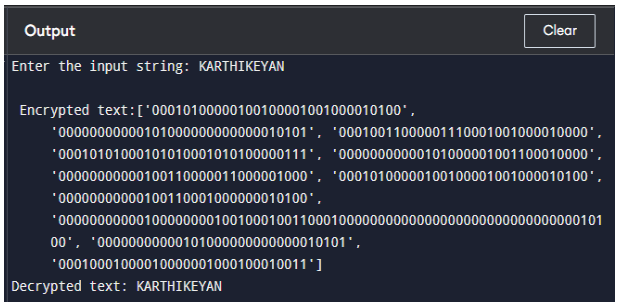

Fig. 5 Output of Proposed Cryptographic Algorithm with input string given in capital letters

Figure 6 shows the Python output of the proposed cryptographic algorithm with only the first character of the input string being a capital letter. The input string is "Ganesan," and the proposed cryptographic algorithm produces a sequence of zeros and ones as the encrypted text. The proposed decryption algorithm produces "Ganesan," as shown in the output, which is similar to the input string. This proposed cryptographic algorithm accepts both capital letters and small letters and performs encryption and decryption.

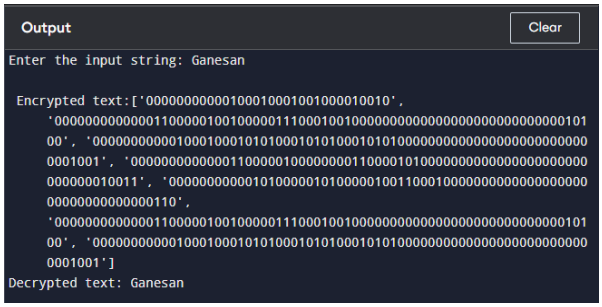

Fig. 6 Output of Proposed Cryptographic Algorithm with only the first character of input string given capital letter

Figure 7 shows the Python output of the proposed cryptographic algorithm with the input string is small letters. The input string given is "sangeetha," and the encrypted text obtained from the proposed cryptographic algorithm is a group of zeros and ones. The decrypted text obtained through this proposed decryption algorithm is "sangeetha," shown in the output, which is similar to the input string. This proposed cryptographic algorithm can also encrypt and decrypt an input string with only lowercase letters.

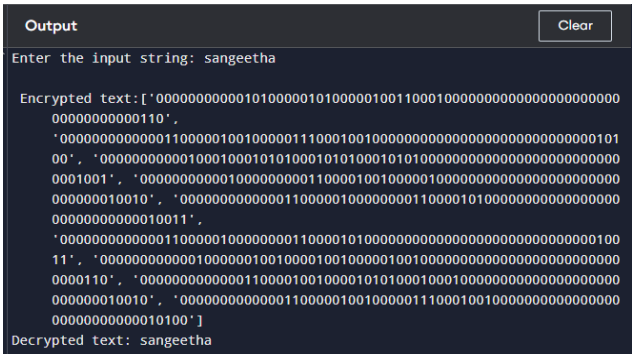

Fig. 7 Output of Proposed Cryptographic Algorithm with input string given small letters

Figure 8 shows the Python output of the proposed cryptographic algorithm with input that contains both letters and numbers. The input string is “Rajesh 537268,” and the proposed cryptographic algorithm produces a sequence of zeros and ones as the encrypted text. The decrypted text obtained through this proposed decryption algorithm is “Rajesh 537268,” shown in the output, which is similar to the input string. This proposed cryptographic algorithm can also encrypt and decrypt input with letters first and numbers next.

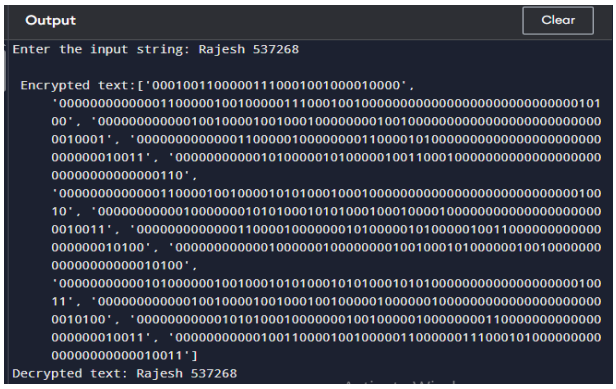

Fig. 8 Output of Proposed Cryptographic Algorithm with input String given first and Numbers next

Figure 9 shows the Python output of the proposed cryptographic algorithm with input containing both numbers and letters. The input string is “74583 Dinesh,” and the encrypted text produced by this proposed cryptographic algorithm is a sequence of zeros and ones. The decrypted text obtained through this proposed decryption algorithm is “74583 Dinesh,” shown in the output, which is similar to the input string. This proposed cryptographic algorithm can also encrypt and decrypt inputs with numbers first and letters next.

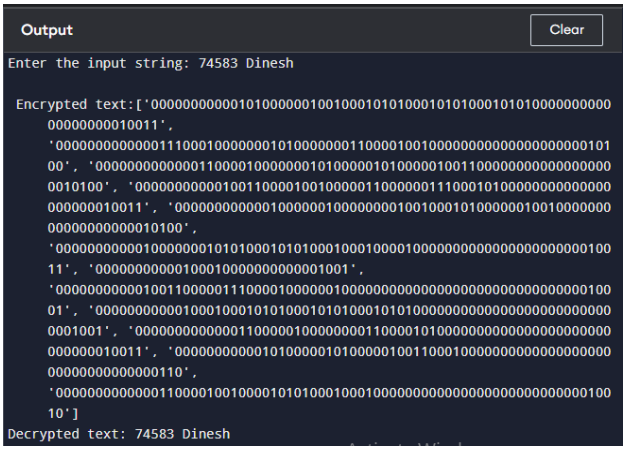

Fig. 9 Output of Proposed Cryptographic Algorithm with input Numbers given first and String next

Figure 10 shows the Python output of the proposed cryptographic algorithm with input that contains both letters and special characters. The input string given is “Suresh #&%$!” and the encrypted text obtained from this proposed cryptographic algorithm is a group of zeros and ones. The decrypted text obtained through this proposed decryption algorithm is “Suresh #&%$!” shown in the output, which is similar to the input string. This proposed cryptographic algorithm can also encrypt and decrypt inputs with both letters and special characters.

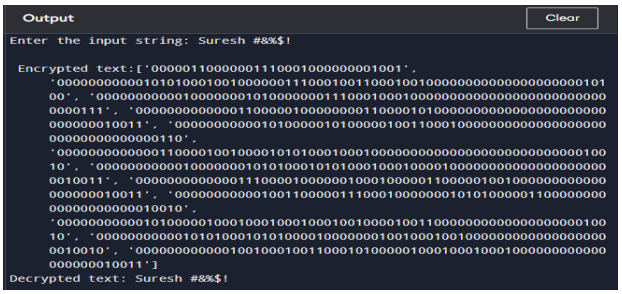

Fig. 10 Output of Proposed Cryptographic Algorithm with input String given first and Special Characters next

Figure 11 shows the Python output of the proposed cryptographic algorithm with an input that contains a combination of letters, numbers, and special characters. The input string given is “Shiva 5274 &$%#” and the encrypted text obtained from this proposed cryptographic algorithm is a group of zeros and ones. The decrypted text obtained through this proposed decryption algorithm is “Shiva 5274 &$%#” shown in the output, which is similar to the input string. This proposed cryptographic algorithm can also encrypt and decrypt input containing letters, numbers, and special characters.

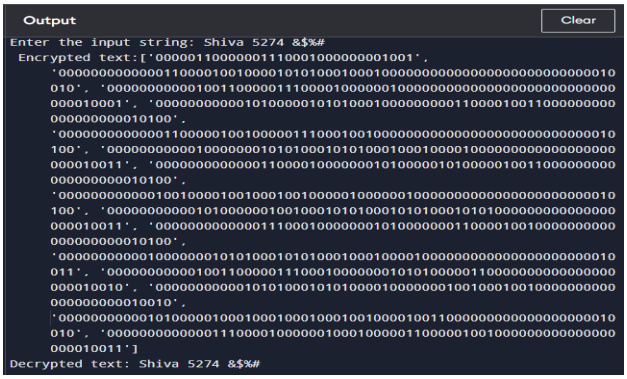

Fig.11 Output of Proposed Cryptographic Algorithm with input given is a combination of strings, Numbers, and Special Characters

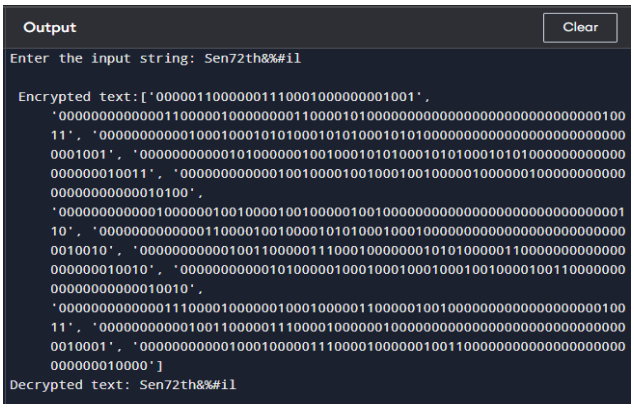

Fig. 12 Output of Proposed Cryptographic Algorithm with input given is a mixture of strings, Numbers, and Special Characters

Figure 12 shows the Python output of the proposed cryptographic algorithm with an input that contains a mixture of letters, numbers, and special characters. The input string given is “Sen72th&%#il” and the encrypted text obtained from this proposed cryptographic algorithm is a group of zeros and ones. The decrypted text obtained through this proposed decryption algorithm is “Sen72th&%#il”, as shown in the output, which is similar to the input string. This proposed cryptographic algorithm can also encrypt and decrypt input with a mixture of letters, numbers, and special characters.

The proposed cryptographic algorithm can encrypt and decrypt strings with uppercase letters, lowercase letters, combinations of uppercase and lowercase, combinations of letters and numbers, and combinations of letters, numbers, and special characters. This proposed cryptographic algorithm offers wide range of security applications include encryption of string, string with numbers and special characters, email messages, WhatsApp messages, SMS (Short Message Service), bank account holder name, bank account number, ATM pin number, bank fixed deposit and recurring deposit number, various insurance policy number, various ID cards such as Aadhaar number, PAN number, Passport number, Voter ID number, Smart card number, College students name, students roll number, faculty name, faculty ID, corporate employee name, employee ID, confidential messages related to bank operations, industrial operations, country borders, central and state government service oriented, online transactions, various transportation related messages include bus, train, and flight. The information and messages related to the various applications mentioned above can be encrypted and transmitted over the Internet.

This proposed cryptographic algorithm aims to ensure the confidentiality of information transmitted over the Internet. The benefit of this proposed algorithm is that the ciphertext length increases significantly with variations or increases in the number of characters in the input plaintext. In addition, there is no limitation on the size of the characters in the input string. This proposed cryptographic algorithm accepts a mixture of letters, numbers, and special characters and performs encryption and decryption. Attackers cannot detect the original input string since the encrypted text is a combination of zeros and ones. This proposed cryptographic algorithm offers strong security against unauthorized access and secures information across a wide range of applications.

IV. Conclusion

The Internet is widely used to communicate process data, private information, online banking transactions, patients’ health conditions, border information, and credit and debit card transactions. Intruders try to gain access and misuse private information, which leads to the loss of original data. Attackers can monitor online transactions worldwide. They try to obtain individuals' account details and transfer money without authorization. As a result, bank customers lose money. Security is essential to keep sensitive data, messages, and information secret during transmission across the internet. This proposed work is a novel encryption and decryption method named the Prasath Srinivasan Sushmitha algorithm for String Encryption and Decryption. This proposed algorithm involves a series of mathematical operations to convert the input string into ciphertext. This ciphertext can be transmitted worldwide over the internet. Decryption is performed in reverse of the encryption to convert the ciphertext into the original string. The benefit of this proposed encryption algorithm is that it uses simple mathematical operations and ensures the confidentiality of data, private information, and messages. This proposed algorithm can be suitable for usage in wide range of applications, including text encryption, securing online transactions of amounts and account details, securing email messages, securing medical data related to patients, securing industrial process information, and securing personal information.

References

[1] G. S. Rizos and K. A. Draziotis, "Cryptographic primitives based on compact knapsack problem," *J. Inf. Secur. Appl.*, vol. 83, p. 103781, Jun. 2024, doi: 10.1016/j.jisa.2024.103781.

[2] L.-W. Chen, K.-L. Tsai, F.-Y. Leu, W.-C. Jiang, and S.-T. Tseng, "Time parameter based low-energy data encryption method for mobile applications," *Comput. Model. Eng. Sci.*, vol. 140, no. 3, pp. 2779–2794, 2024, doi: 10.32604/cmes.2024.052124

[3] A. E. Adeniyi, R. G. Jimoh, and J. B. Awotunde, "A systematic review on elliptic curve cryptography algorithm for Internet of Things: Categorization, application areas, and security," *Comput. Electr. Eng.*, vol. 118, p. 109330, Aug. 2024, doi:10.1016/j.compeleceng.2024.109330.

[4] J. S. Prasath, V. I. Shyja, P. Chandrakanth, B. K. Kumar, and A. Raja Basha, "An optimal secure defense mechanism for DDoS attack in IoT network using feature optimization and intrusion detection system," *J. Intell. Fuzzy Syst.*, vol. 46, no. 3, pp. 6517–6534, Mar. 2024, doi:10.3233/jifs-235529.

[5] N. Mishra, S. K. Hafizul Islam, and S. Zeadally, "A survey on security and cryptographic perspective of Industrial-Internet-of-Things," *Internet Things*, vol. 25, p. 101037, Apr. 2024, doi:10.1016/j.iot.2023.101037.

[6] M. Al-Zubaidie and R. A. Muhajjar, "Integrating trustworthy mechanisms to support data and information security in health sensors," *Procedia Comput. Sci.*, vol. 237, pp. 43–52, 2024, doi:10.1016/j.procs.2024.05.078.

[7] O. Popoola, M. A. Rodrigues, J. Marchang, A. Shenfield, A. Ikpehai, and J. Popoola, "An optimized hybrid encryption framework for smart home healthcare: Ensuring data confidentiality and security," *Internet Things*, vol. 27, p. 101314, Oct. 2024, doi:10.1016/j.iot.2024.101314.

[8] V. Vasani, K. Prateek, R. Amin, S. Maity, and A. D. Dwivedi, "Embracing the quantum frontier: Investigating quantum communication, cryptography, applications and future directions," *J. Ind. Inf. Integr.*, vol. 39, p. 100594, May 2024, doi:10.1016/j.jii.2024.100594.

[9] C. J. Ramakrishna, D. B. K. Reddy, B. K. Priya, P. P. Amritha, and K. V. Lakshmy, "Analysis of lightweight cryptographic algorithms for IoT gateways," *Procedia Comput. Sci.*, vol. 233, pp. 235–242, 2024, doi: 10.1016/j.procs.2024.03.213.

[10] H. Dou, Z. Dan, P. Xu, W. Wang, S. Xu, T. Chen, and H. Jin, "Dynamic searchable symmetric encryption with strong security and robustness," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 2370–2384, 2024, doi:10.1109/tifs.2024.3350330.

[11] K. Sasikumar and S. Nagarajan, "Comprehensive review and analysis of cryptography techniques in cloud computing," *IEEE Access*, vol. 12, pp. 52325–52351, 2024, doi:10.1109/access.2024.3385449.

[12] G. Xu, S. Xu, J. Ma, J. Ning, and X. Huang, "An adaptively secure and efficient data sharing system for dynamic user groups in cloud," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 5171–5185, 2023, doi:10.1109/tifs.2023.3305870.

[13] L. Chen, Y. Mu, L. Zeng, F. Rezaeibagha, and R. H. Deng, "Authenticable data analytics over encrypted data in the cloud," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1800–1813, 2023, doi:10.1109/tifs.2023.3256132.

[14] L. Xu, X. Yuan, Z. Zhou, C. Wang, and C. Xu, "Towards efficient cryptographic data validation service in edge computing," in *Proc. 2022 IEEE World Congr. Services (SERVICES)*, IEEE, 2022, pp. 14–14, doi:10.1109/services55459.2022.00027.

[15] C. Hahn, H. Yoon, and J. Hur, "Multi-key similar data search on encrypted storage with secure pay-per-query," *IEEE Trans. Inf.*

Forensics Security, vol. 18, pp. 1169–1181, 2023, doi:10.1109/tifs.2023.3236178.

[16] E. K. Heru, Faisal, and H. Pranoto, "File encryption application using Menezes-Vanstone elliptic curve cryptography based on Python," *Procedia Comput. Sci.*, vol. 227, pp. 651–658, 2023, doi:10.1016/j.procs.2023.10.569.

[17] F. Beggas and A. Lounici, "Generation of random sequences using DNA cryptography for OTP encryption," *Biosystems*, vol. 234, p. 105064, Dec. 2023, doi:10.1016/j.biosystems.2023.105064.

[18] K. Begovic, A. Al-Ali, and Q. Malluhi, "Cryptographic ransomware encryption detection: Survey," *Comput. Secur.*, vol. 132, p. 103349, Sep. 2023, doi:10.1016/j.cose.2023.103349.

[19] A. Vishnoi, A. Aggarwal, A. Prasad, M. Prateek, and S. Aggarwal, "Text encryption for lower text size: Design and implementation," *Mater. Today Proc.*, vol. 79, pp. 278–281, 2023, doi:10.1016/j.matpr.2022.11.118.

[20] S. Urooj, S. Lata, S. Ahmad, S. Mehfuz, and S. Kalathil, "Cryptographic data security for reliable wireless sensor network," *Alexandria Eng. J.*, vol. 72, pp. 37–50, Jun. 2023, doi:10.1016/j.aej.2023.03.061.

[21] P. Ramadevi, S. Ayyasamy, Y. Suryaprakash, C. Anilkumar, S. Vijayakumar, and R. Sudha, "Security for wireless sensor networks using cryptography," *Meas. Sensors*, vol. 29, p. 100874, Oct. 2023, doi:10.1016/j.measen.2023.100874.

[22] H. Kadry, A. Farouk, E. A. Zanaty, and O. Reyad, "Intrusion detection model using optimized quantum neural network and elliptical curve cryptography for data security," *Alexandria Eng. J.*, vol. 71, pp. 491–500, May 2023, doi:10.1016/j.aej.2023.03.072.

[23] S. Gadde, J. Amutharaj, and S. Usha, "A security model to protect the isolation of medical data in the cloud using hybrid cryptography," *J. Inf. Secur. Appl.*, vol. 73, p. 103412, Mar. 2023, doi:10.1016/j.jisa.2022.103412.

[24] R. Achary, C. J. Shelke, K. Marx, and A. Rajesh, "Security implementation on IoT using CoAP and elliptical curve cryptography," *Procedia Comput. Sci.*, vol. 230, pp. 493–502, 2023, doi:10.1016/j.procs.2023.12.105.

[25] D. Shivaramakrishna and M. Nagaratna, "A novel hybrid cryptographic framework for secure data storage in cloud computing: Integrating AES-OTP and RSA with adaptive key management and time-limited access control," *Alexandria Eng. J.*, vol. 84, pp. 275–284, Dec. 2023, doi:10.1016/j.aej.2023.10.054.

[26] M. I. Reddy, M. P. Reddy, R. O. Reddy, and A. Praveen, "Improved elliptical curve cryptography and chaotic mapping with fruitfly optimization algorithm for secure data transmission," *Wireless Netw.*, vol. 30, no. 3, pp. 1151–1164, Nov. 2023, doi:10.1007/s11276-023-03554-8.

[27] Q. Tong, Y. Miao, J. Weng, X. Liu, K.-K. R. Choo, and R. Deng, "Verifiable fuzzy multi-keyword search over encrypted data with adaptive security," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2022, doi:10.1109/tkde.2022.3152033.

[28] D. Chen, H. Wang, N. Zhang, X. Nie, H.-N. Dai, K. Zhang, and K.-K. R. Choo, "Privacy-preserving encrypted traffic inspection with symmetric cryptographic techniques in IoT," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17265–17279, Sep. 2022, doi:10.1109/jiot.2022.3155355.

[29] L. Zhang, H. Xiong, Q. Huang, J. Li, K.-K. R. Choo, and J. Li, "Cryptographic solutions for cloud storage: Challenges and research opportunities," *IEEE Trans. Services Comput.*, vol. 15, no. 1, pp. 567–587, Jan. 2022, doi:10.1109/tsc.2019.2937764.

[30] M. Nadeem, A. Arshad, S. Riaz, S. W. Zahra, S. S. Band, and A. Mosavi, "Two layer symmetric cryptography algorithm for protecting data from attacks," *Comput. Mater. Continua*, vol. 74, no. 2, pp. 2625–2640, 2023, doi:10.32604/cmc.2023.030899.

[31] P. Garg and D. Kumar Singh, "Analysis of cryptographic encryption algorithm design to secure IoT devices: A review," *Mater. Today Proc.*, vol. 51, pp. 810–814, 2022, doi:10.1016/j.matpr.2021.06.240.

[32] D. Kumar Sharma, N. Chidananda Singh, D. A. Noola, A. Nirmal Doss, and J. Sivakumar, "A review on various cryptographic techniques & algorithms," *Mater. Today Proc.*, vol. 51, pp. 104–109, 2022, doi:10.1016/j.matpr.2021.04.583.

[33] M. H. Murtaza, H. Tahir, S. Tahir, Z. A. Alizai, Q. Riaz, and M. Hussain, "A portable hardware security module and cryptographic key generator," *J. Inf. Secur. Appl.*, vol. 70, p. 103332, Nov. 2022, doi:10.1016/j.jisa.2022.103332.

[34] Y. Zhou, B. Hu, Y. Zhang, and W. Cai, "Implementation of cryptographic algorithm in dynamic QR code payment system and its performance," *IEEE Access*, vol. 9, pp. 122362–122372, 2021, doi:10.1109/access.2021.3108189.

[35] A. Ometov, K. Zeman, P. Masek, L. Balazevic, and M. Komarov, "A comprehensive and reproducible comparison of cryptographic primitives execution on Android devices," *IEEE Access*, vol. 9, pp. 54625–54638, 2021, doi:10.1109/access.2021.3069627.

[36] J. Li, Y. Huang, Y. Wei, S. Lv, Z. Liu, C. Dong, and W. Lou, "Searchable symmetric encryption with forward search privacy," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 1, pp. 460–474, Jan. 2021, doi:10.1109/tdsc.2019.2894411.

[37] M. A. F. Al-Husainy, B. Al-Shargabi, and S. Aljawarneh, "Lightweight cryptography system for IoT devices using DNA," *Comput. Electr. Eng.*, vol. 95, p. 107418, Oct. 2021, doi:10.1016/j.compeleceng.2021.107418.

[38] F. Thabit, S. Alhomdy, A. H. A. Al-Ahdal, and S. Jagtap, "A new lightweight cryptographic algorithm for enhancing data security in cloud computing," *Global Transitions Proc.*, vol. 2, no. 1, pp. 91–99, Jun. 2021, doi:10.1016/j.gltp.2021.01.013.

[39] C. Chiñas-Palacios, J. Aguila-Leon, C. Vargas-Salgado, E. X. M. Garcia, J. Sotelo-Castañon, and E. Hurtado-Perez, "A smart residential security assisted load management system using hybrid cryptography," *Sustain. Comput. Informat. Syst.*, vol. 32, p. 100611, Dec. 2021, doi:10.1016/j.suscom.2021.100611.

[40] A. Mondal and R. T. Goswami, "Enhanced honeypot cryptographic scheme and privacy preservation for an effective prediction in cloud security," *Microprocess. Microsyst.*, vol. 81, p. 103719, Mar. 2021, doi:10.1016/j.micpro.2020.103719.

[41] D. B. Rawat, R. Doku, and M. Garuba, "Cybersecurity in big data era: From securing big data to data-driven security," *IEEE Trans. Services Comput.*, vol. 14, no. 6, pp. 2055–2072, Nov. 2021, doi:10.1109/tsc.2019.2907247.

[42] J. S. Prasath, U. Ramachandraiah, and G. Muthukumaran, "Modified hardware security algorithms for process industries using Internet of Things," *J. Appl. Secur. Res.*, vol. 16, no. 1, pp. 127–140, Mar. 2020, doi:10.1080/19361610.2020.1736458.

[43] H. Deng, Z. Qin, Q. Wu, Z. Guan, R. H. Deng, Y. Wang, and Y. Zhou, "Identity-based encryption transformation for flexible sharing of encrypted data in public cloud," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3168–3180, 2020, doi:10.1109/tifs.2020.2985532.

[44] S. Zhang, H. Xian, Z. Li, and L. Wang, "SecDedup: Secure encrypted data deduplication with dynamic ownership updating," *IEEE Access*, vol. 8, pp. 186323–186334, 2020, doi:10.1109/access.2020.3023387.

[45] Y. Ke, M.-Q. Zhang, J. Liu, T.-T. Su, and X.-Y. Yang, "Fully homomorphic encryption encapsulated difference expansion for reversible data hiding in encrypted domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 8, pp. 2353–2365, Aug. 2020, doi:10.1109/tcsvt.2019.2963393.

[46] J. S. Prasath, S. Jayakumar, and K. Karthikeyan, "Real-time implementation for secure monitoring of wastewater treatment plants using Internet of Things," *Int. J. Innov. Technol. Exploring Eng.*, vol. 9, no. 1, pp. 2997–3002, Nov. 2019, doi:10.35940/ijitee.a9123.119119.

[47] M. H. Al Hasani and K. A. Al Naimee, "Impact security enhancement in chaotic quantum cryptography," *Opt. Laser Technol.*, vol. 119, p. 105575, Nov. 2019, doi:10.1016/j.optlastec.2019.105575.

[48] G. Wu, F. Guo, and W. Susilo, "Generalized public-key cryptography with tight security," *Inf. Sci.*, vol. 504, pp. 561–577, Dec. 2019, doi:10.1016/j.ins.2019.07.041.

[49] R. Hodgson, "Solving the security challenges of IoT with public key cryptography," *Netw. Secur.*, vol. 2019, no. 1, pp. 17–19, Jan. 2019, doi:10.1016/s1353-4858(19)30011-x.

[50] S. Kumar, M. Kumar, R. Budhiraja, M. K. Das, and S. Singh, "A cryptographic model for better information security," *J. Inf. Secur. Appl.*, vol. 43, pp. 123–138, Dec. 2018, doi:10.1016/j.jisa.2018.10.011.

[51] G. Panchal and D. Samanta, "A novel approach to fingerprint biometric-based cryptographic key generation and its applications to storage security," *Comput. Electr. Eng.*, vol. 69, pp. 461–478, Jul. 2018, doi:10.1016/j.compeleceng.2018.01.028.