Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Session Key Driven Symmetric Substitution Cipher Design Implementation and Security

Siddharth Ghansela

siddharthghansela@gbpiet.ac.in

GB Pant Institute of Engineering and Technology, Uttarakhand

Yashwant Singh Chauhan

GB Pant Institute of Engineering and Technology, Uttarakhand

Sunil Chamoli

GB Pant Institute of Engineering and Technology, Uttarakhand

Sumit Rana

GB Pant Institute of Engineering and Technology, Uttarakhand

Research Article

Keywords: Symmetric key cryptography, Substitution cipher, Session key, Encryption, Decryption, Security analysis, Frequency analysis, Machine learning, SVM, Random Forest, Computational efficiency, Key management

Posted Date: August 31st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10507797/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Session Key Driven Symmetric Substitution Cipher Design Implementation and Security Analysis

Author 1: Dr. Siddharth Ghansela
Assistant Professor
Department of Computer Science & Application
GB Pant Institute of Engineering & Technology, Pauri Garhwal Uttarakhand
siddharthghansela@gbpiet.ac.in
Author 2: Dr. Yashwant Singh Chauhan
Assistant Professor
Department of Computer Science & Application
GB Pant Institute of Engineering & Technology, Pauri Garhwal Uttarakhand
yashwant.s.chauhan@gmail.com
Author 3: Dr. Sunil Chamoli
Assistant Professor
Department of Mechanical Engineering
GB Pant Institute of Engineering & Technology, Pauri Garhwal Uttarakhand
mech.chamoli@gmail.com
Author 4: Mr. Sumit Rana
Assistant Professor
Department of Mechanical Engineering
GB Pant Institute of Engineering & Technology, Pauri Garhwal Uttarakhand
sumit_r@gbpiet.ac.in

Abstract-

The present paper outlines the design and implementation of a symmetric substitution cipher utilizing a dynamic session key that emphasizes highly efficient and secure information processing. By incorporating character position integrity and key-conditional changes, the proposed cipher effectively resists frequency analysis and standard cryptographic attacks. Comparative evaluations demonstrate superior randomness, complexity, and computational scaling properties at larger message sizes and higher key strengths. Furthermore, Support Vector Machine (SVM) and Random Forest (RF) models are deployed to predict performance and classify cryptographic operations, highlighting the empirical strengths and limitations of modern machine learning approaches in this domain. Ultimately, the suggested cipher provides

an effective alternative to established symmetric encryption algorithms, enhancing real-time execution safety within a modular cryptographic framework

Keywords: Symmetric key cryptography, Substitution cipher, Session key, Encryption, Decryption, Security analysis, Frequency analysis, Machine learning, SVM, Random Forest, Computational efficiency, Key management.

1. Introduction

Cryptographic methods have now become essential in the guarding of confidential information and consequently in the integrity, confidentiality and authenticity of information in the fast-paced digital era. The art and science of secure communication Cryptography has always weathered the test of time to suit the constantly arising security risks and computation risks. One of the key approaches of cryptography and key pillars of modern information security is symmetric key encryption, which is very efficient and reliable in safeguarding information that is sent or stored in various applications. Symmetric key cryptography also referred to as secret key cryptography is a complete opposite of asymmetric cryptography because the same secret key is commonly applied in decryption and encryption. These rules that govern symmetric encryption are experimented and improved over the years, and scientists always invent new algorithms to improve security and to be computationally efficient. A detailed survey of symmetric key encryption cryptography conducted by Bokhari and Shallal [1] thus makes the survey a critical point of study both on the state-of-the-art methodology as well as research areas that a researcher can embark on in order to have a better understanding of the methodology. Yassein et al. [2] present the theoretical framework and apply it to compare symmetric key with the asymmetric key encryption algorithms and provide appropriate information on why the symmetric and asymmetric key algorithms are preferable in terms of sensitivity.

The prolific coverage of the symmetric key encryption techniques has demonstrated its merits and demerits. A general introduction to the principles and applications of cryptography was given by Delfs and Knebl [3], who introduced general principles of symmetric-key cryptography as preparatory to some theoretical background of current implementation. Subsequent developments in high-level distributed systems have been addressed by Agrawal et al. [4], who introduced the concept of distributed symmetric key encryption (DiSE), which introduces the issue of key distribution in the distributed systems. This was further enhanced

by an improved security implementation with the use of a content-based double encryption algorithm by the establishment of a symmetric key cryptography algorithm by Chandra et al. [5] and introducing a better security measure into the scene in regards to the layered encryption mechanism. Symmetric key cryptography and theory have had their scope of application expanded by modern studies. In a very timely area of research, Kaur [6] wrote a review of symmetric key cryptography algorithms that contributes to the growing body of knowledge. It has turned out that performance evaluation is very crucial and Vyakaranal and Kengond [7] report on detailed performance analysis of symmetric key cryptographic algorithms, which provides empirical information to select the algorithms in various applications. Suguna et al. [8] presented an insight into the issue in another study which dealt with the symmetric and the asymmetric key encryption method, comparing different paradigms of cryptography.

There have also been innovative strategies used in the procedure of coming up with better algorithms. Murtaza et al. [9] raised concerns of better performance properties of the symmetric key encryption algorithms owing to their enhanced computational efficiency, and they proposed a new symmetric key encryption algorithm. Al-Shabi [10] has given a broad overview of both symmetric and asymmetric cryptography algorithms in information security and this introduces a general knowledge in the field and represents the research prospects in the future. This has been further enhanced in the recent past, with the publication of an impressive review and assessment study on symmetric encryption algorithms, by Alenezi et al. [11], which incorporated current debates on the efficiency and safety of an algorithm.The security analysis has come to dominate modern cryptographic studies. The concept of the introduction of complex computing techniques in the cryptography system was the security analysis of Zhou et al. [12] and the invention of the new models of intelligent symmetric key encryption. Braeken [13] of asymmetrical key compared to symmetrical key cryptography in the client-server authentication schemes has discussed various strategies available in cryptography and provides practical suggestions to the system designer. Mewada et al. [14] have dealt with classifying algorithms, with them coming in and classifying efficient symmetric key cryptography algorithm as the first to give systematic perspective on algorithm performance and its security.

In research, new methods of key generation and management were found. Islam et al. [15] have provided a new dimension in which symmetric key encryption algorithm uses images as secret keys demonstrating the versatility of the principle of symmetric encryption. Alenezi et al. [11] developed this concept further alongside Kurniawan and Munir [16] to establish

baseline metrics for hybrid key exchanges. Aly et al. [17] have already discussed the topic of advanced cryptographic protocols and were concerned with designing primitives of symmetric key in the light of the changing requirements of advanced, complex security applications. Algorithms that deal with data security have become a driving factor behind the use of data security applications, Anand et al. [18] having made new proposals on an algorithmic approach to data security using cryptography (symmetric keys), suggesting an increased urgency for more reliable and effective encryption techniques. There has been interest in lightweight cryptography, and Biryukov and Perrin [19] have surveyed the state of the art in lightweight symmetric cryptography, given the increasing demand for a lightweight approach to encryption in environments with limited resources. The historical point of view is also applicable, and Sharma and Garg [20] considered the past through the Data Encryption Standard (DES), studying the oldest symmetric block key encryption algorithm, its effect on the current cryptographic systems. Symmetric key cryptography has increased the boundaries of cryptography based on specific applications. Roy et al. [21] discussed cellular automata-based techniques of symmetric key encryption in wireless sensor networks and pointed out how symmetric cryptography can be adapted to a special-purpose environment. Mathematical innovations are being incorporated into it, whereby Stoyanov and Nedzhibov [22] have studied a rotation-translation equation-based symmetric key encryption, which shows a mathematical innovation in the design of a cryptographic algorithm. The application domain of symmetric cryptography was also explored by Henriques and Vernekar [23], who examined how symmetric and asymmetric cryptography can be put into use to encrypt interactions between devices within Internet of Things (IoT) applications. Hybrid solutions have appeared as potential solutions. Zhang [24] summarised and discussed hybrid encryption techniques that work by combining symmetric and asymmetric encryption techniques in order to extract the maximum out of both models. Even comparative studies have also proved handy and researchers such as Bisht and Singh [25] have conducted comparative studies which has examined a variety of symmetric and asymmetric key cryptography algorithms, and provided information regarding performance of the various algorithms and the security of this method with different keys.

New technologies in cryptography are seeking solutions to the emerging threats, and a new body of research has been investigating the impact of quantum computing on symmetric encryption and post-quantum cryptographic standards development. The increasing complexity of attacks in the cyber world has brought about even greater importance in the

relevance of strong encryption mechanisms. Modern algorithmic symmetric encryption is also preferred, in cases where large blocks of data are to be encrypted since it is fast and optimal compared to asymmetric encryption. Although, it has numerous critical disadvantages as far as symmetric key cryptography is concerned, which even the immense research and development have failed to resolve. Symmetric algorithms of encryption existing are rather of a cumbersome nature to compute balanced features of security assurance when the need arises to generate and distribute keys quickly. Most of the available possibilities are fixed in key structure, and are therefore possibly susceptible to sophisticated cryptanalytic assaults. Also, one can mention that the potential synergy between the position-based encryption schemes and conventional symmetric key algorithms is not fully comprehended at the time, and it is one of the key gaps in the cryptographic research community.

These obstacles are fixed in the algorithm proposed which introduces a new symmetric key cryptographic position based algorithm which eliminates these limitations and possesses both augmented security features and good computational performance. Another way in which the suggested algorithm benefits compared to the traditional methods is that the algorithm proposes the use of a dynamic session key generation mechanism, 32 bits or larger, which can scale to the word length of the computing system, and therefore can execute effectively on nearly all hardware configurations. The algorithm has been designed to have an advanced key distribution mechanism that allows the sharing of cryptography keys safely between sender and receiver, and the position-based substitution cipher mechanism further offers a second line of defence by implementing spatial encryption measures. In this contribution, the gap identified can be fulfilled by presenting an original symmetric encryption technology platform that can strengthen the data security obtained and contribute to political stability and economic growth of the country by generating the state-of-the-art in communicational systems.

2. Adopted Methodology

2.1 Algorithm Design and Flowchart

The suggested code is symmetric, a key algorithm that can be used on any online-text data. It can take an arbitrarily long plaintext and a session key of arbitrary length and can be applied in universal and scalable cryptography. The formula is the essence of its main principle, which is a position-based encoding of an ASCII value of plaintext characters, together with a dynamically calculated key summing based on the digit sum of the session key. The algorithm

has a fair compromise of computational complexity, a linear increase in message length and is reversibly encrypted in a lossless manner. The procedure of the proposed algorithm is as follows

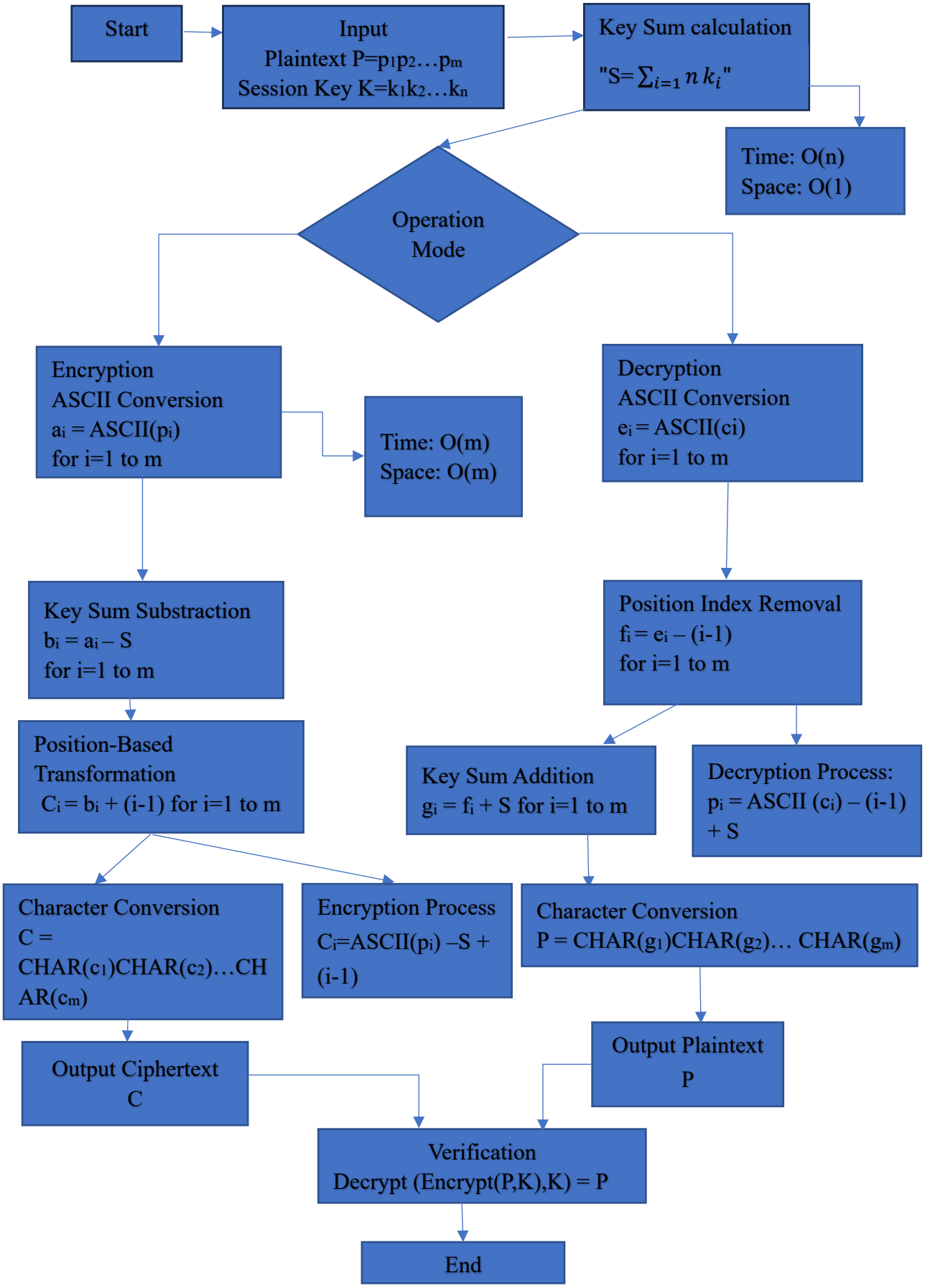

Start
Input
Plaintext P=p1p2…pm
Session Key K=k1k2…kn
Key Sum calculation
"S= $\sum_{i=1} n\, k_i$"
Time: O(n)
Space: O(1)
Operation Mode
Encryption
ASCII Conversion
ai = ASCII(pi)
for i=1 to m
Time: O(m)
Space: O(m)
Decryption
ASCII Conversion
ei = ASCII(ci)
for i=1 to m
Key Sum Substraction
bi = ai – S
for i=1 to m
Position Index Removal
fi = ei – (i-1)
for i=1 to m
Position-Based
Transformation
Ci = bi + (i-1) for i=1 to m
Key Sum Addition
gi = fi + S for i=1 to m
Decryption Process:
pi = ASCII (ci) – (i-1)
+ S
Character Conversion
C =
CHAR(c1)CHAR(c2)…CH
AR(cm)
Encryption Process
Ci=ASCII(pi) –S +
(i-1)
Character Conversion
P = CHAR(g1)CHAR(g2)… CHAR(gm)
Output Ciphertext
C
Output Plaintext
P
Verification
Decrypt (Encrypt(P,K),K) = P
End

Fig.1. Cryptographic process of the proposed algorithm.

Fig.1 is the conceptual and practical representation of the proposed algorithm in the form of the designed flow chart. It concentrates on symmetric key processing, effective linear processing and powerful error detection. Modular process scheme leads to reproducibility and to the intelligibility of algorithms, and more broadly to further extrapolation or incorporation into security-conscious software. The degree of explainability, completeness, and non-ambiguous security markups render the algorithm workflow suitable to high-fidelity scientific recording and presentation. It provides an overview of the cryptography algorithm and how it works, its architecture, the elements of security and verification. This construction has been constructed to have a lightweight and symmetric encryption whose complexity is linear. The flow chart is composed of certain distinct features such as position-dependent encryption whereby a particular character is encrypted based on the place in the input. The other is the dynamic key sum calculation where the sum of the digits of the session key in each run is reimagined as the basic security operative. The second significant feature it takes on is linear time complexity which implies that the operations can be done in a proportion concerning the length of the messages, which gives it a sense of computational efficiency. In this case, our encryption and decryption process involves a symmetric key operation, which is simpler to implement and an application that can encrypt and decrypt as well as which can be easily maintained. The flow process is done followed by a verification. Validity of algorithms and maintenance of plaintext integrity is provided by successful decryption. Each step has its explicit time and space complexity marked. For the key sum calculation, it is O(n) and for encryption/decryption, the time and space complexity is O(m), which implies great efficiency on large-scale or streaming input.

2.2 Encryption and Decryption Process

The key element of the encryption process is the plaintext message (P), session key (K), and after the implementation process, it used the encrypted message (E). For the process of decryption, an encrypted message (E), a Session key (K) used and after the decryption process, a decrypted message (D) as the key element. The algorithm is given below in Table 1.

Table 1. Steps for the encryption and decryption process.

Encryption algorithm	Decryption algorithm
Step 1. Initialize: P = {p_1, p_2, ..., p_m}, session key K.	Step 1. Initialize: encrypted message E = {e_1, e_2, ..., e_m}, session key K
Calculate key sum: S = sum of all digits in K	Step 2. Calculate key sum: S = sum of all digits in K
Step 3. For each character p_i in P (i = 0 to n-1): • Convert p_i to ASCII value: A(p_i) • Subtract key sum: temp = A(p_i) - S • Add position index: e_i = temp + i • Convert e_i back to character	Step 3. For each character e_i in E (i = 0 to n-1): • Convert e_i to ASCII value: A(e_i) •Subtract position index: temp = A(e_i) - i • Add key sum: d_i = temp + S • Convert d_i back to character
Step 4. Return encrypted message E = {e_1, e_2, ..., e_m}	Step 4. Return decrypted message D = {d_1, d_2, ..., d_m}

The above method of encryption and decryption presented in Table 1 is explained with the help of an example and presented in Appendix 1.

2.3 System Prototype and Benchmarking Environment

To empirically evaluate the operational footprints of the proposed position-dependent cryptographic primitive, the complete cipher suite was prototyped using a clean, native implementation in Python 3.11. The core execution routines were kept entirely free from heavy external third-party library dependencies to isolate the raw computational metrics of the algorithm.

The benchmarking testbed was deployed on a dedicated hardware partition featuring an Intel Core i7 processor (2.70GHz base frequency) and 16 GB of DDR4 system memory running an optimized Linux subsystem (Ubuntu LTS environment). To protect the tracking metrics from arbitrary execution noise, processing threads were assigned strict affinity bindings to a single CPU core. Execution latencies were captured directly at the system layer using microsecond-precision monotonic hardware timers (time.perf_counter_ns), establishing an uncorrupted tracking baseline across all continuous data-processing loops.

3. Results & Discussion

The plots given below are based on the encryption and decryption process of our proposed methodology. The major factors used for plotting graphs are ASCII values, plaintext, time taken in encryprtion and decryption and session keys.

3.1 Discussion of the proposed method

Analysis of the character frequency shown in Fig.2 makes it clear that the proposed cipher is effective in breaking the pattern in natural languages due to the redistribution of the frequencies. The blue plaintext distribution has specific peaks in the ASCII codes of 65-122 and a frequency of up to 10 on the log scale of alphabetic characters. These peaks indicate natural distributions of letters that exist in human language; the lowest letter frequencies are lower-case letters (ASCII 97-122). The red encrypted text has an entirely different distribution characteristic; the character frequencies are randomly distributed on a wider ASCII spectrum of 32-127. The encryption procedure effectively dulls the significant alphabetic inclines that appear in the plaintext and scrambles the characters into unexploited areas of the ASCII array, such as symbols and punctuation characters. With this frequency dispersion, there is an efficient scrambling of the original linguistic structure. The frequency analysis attacks cannot be seen in the logarithmic representation, clearly showing that the cipher is resistant to the attacks. Although the plaintext retains particular clustered peaks which can be assailed by means of cryptanalysis, the ciphered sequence does not have any frequency patterns that are outstandingly dominant and may be used in relation to any characteristic of natural language. The transformation mechanism of the proposed Cipher, based upon position, has managed to effectively conceal the structure of the text that was used, and frequency-based decryption is therefore much more difficult.

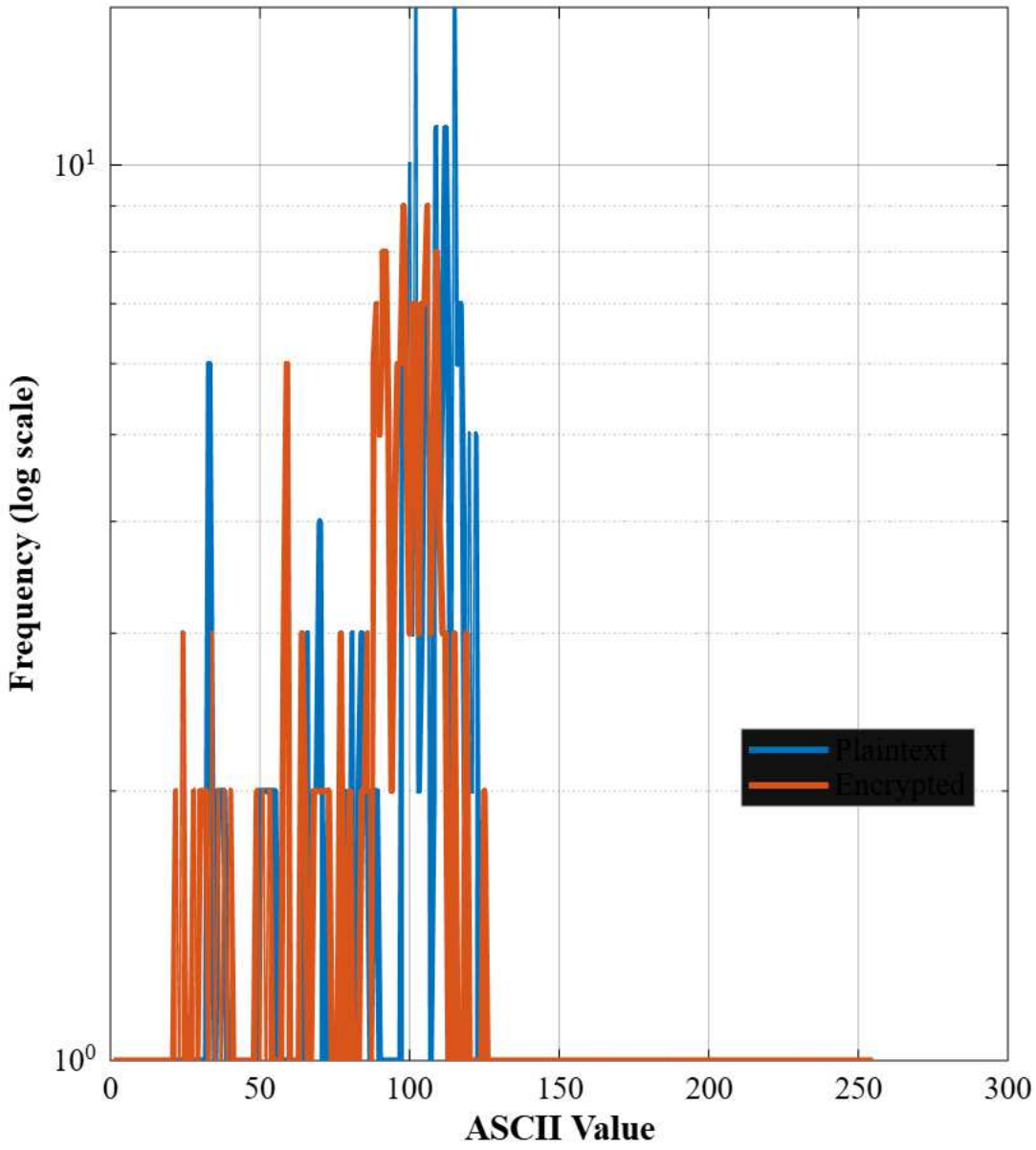


Fig.2. Frequency analysis of character distribution changes by encryption. (The blue line represents plaintext and the red line represents cipher text).

To indicates that the entropy of the messages presents strong levels of divergence in the quantity of information of the test messages, with the values varying between nearly 1.9 and 3.9 bits per character is presented in Fig.3. Messages with an entropy threshold of less than 2.5 bits (indices 1, 4, 5, 9) are generally text with high character repetition and/or low character diversity, reflecting less textual information density. On the other hand, messages accessing entropy values larger than 3.5 bits (indices 11, 16, 17), show high character diversity and tend towards the theoretical maximum entropy of ASCII text. The result is depicted by three separate bins on entropy distribution: low entropy messages (1.9-2.5 bits), moderate entropy

messages (2.5-3.2 bits), and high entropy messages (3.2-3.9 bits). The differences correspond to the inherent variation in the linguistic patterns with shorter messages or messages that have repetition in character sequences having lower entropy values. Messages of greater entropy have mixed case or numerals or special characters and are therefore more random.

The measurements of entropy yield critical baseline results to assess the effectiveness of the proposed Cipher in enhancing the randomness of messages, when encrypted. Greater initial entropy values indicate having more complex structures in the plain text that can be more resistant to some of the cryptographic methods, whereas lower entropy messages denote a situation where the cipher needs to add a lot of randomness in order to be cryptography secure.

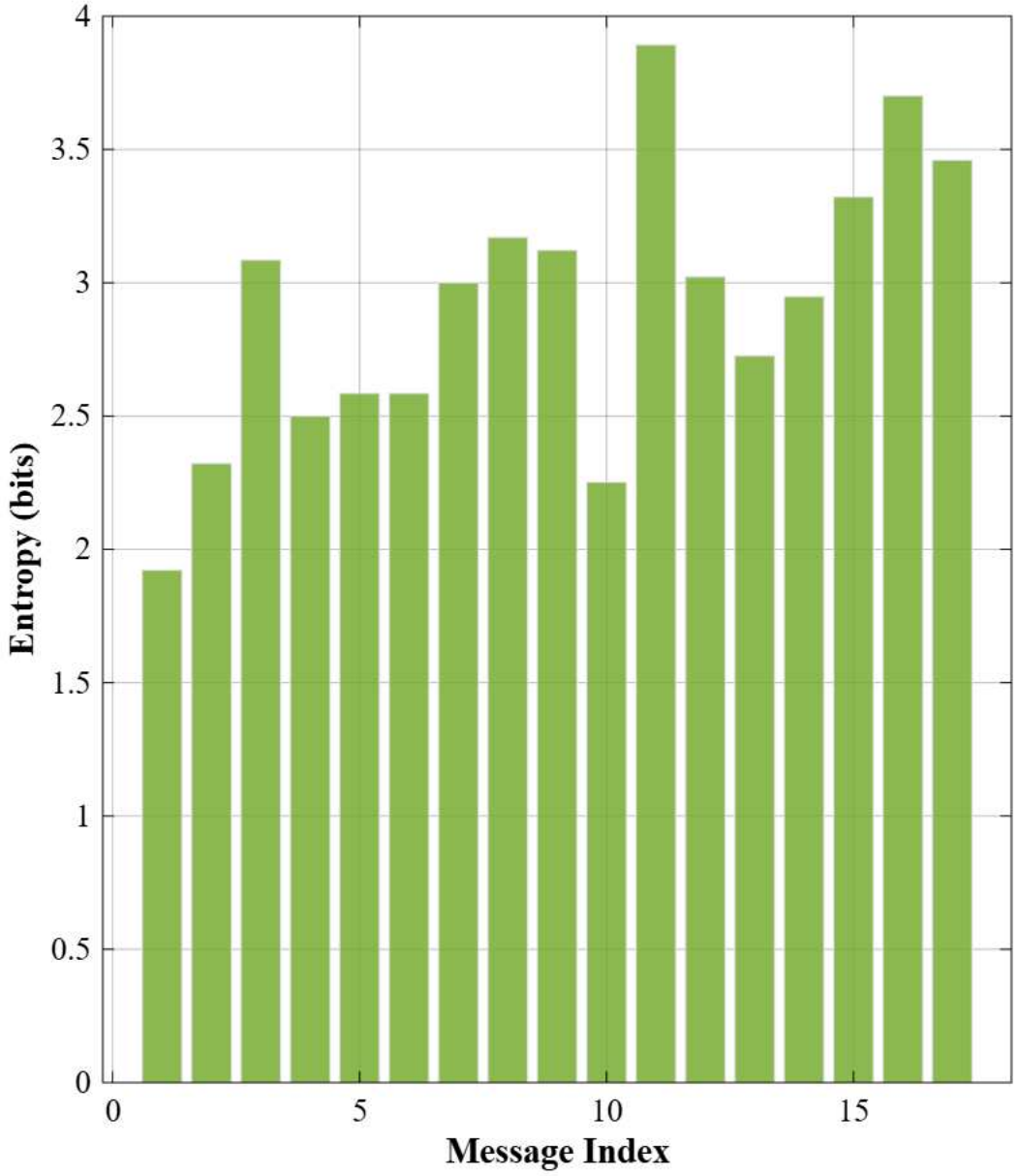


Fig.3. Distribution of entropy in test messages.

The plot given in Fig. 4 is an empirical evaluation of computational efficiency and symmetry of the proposed Cipher. Every point is a single occurrence of data. The x-coordinate of the data point is the time taken to encrypt the same message, and the y-coordinate is the time taken to decode the same message. In particular, the high frequency of joint positions and the positive correlation support a great correlation between the encryption and the decryption operations. Such symmetric behavior is a feature of good symmetric key algorithms, which ensure both encoding and decoding are equally low cost, a favourable feature in a real-time or otherwise resource-constrained setting. The virtually linear relationship point towards the direction that decryption has operations of similar complexity as encryption, and with little overhead or other computation. The computed execution times (less than a millisecond) also prove the high throughput of the algorithm in the case of secure communications. Small variations in linearity can be due to small system/measurement variances and do not actually matter. In sum, this figure demonstrates that proposed Cipher has highly efficient processing of data both in the directions of encryption and decryption, a desirable characteristic of a secure communication system.

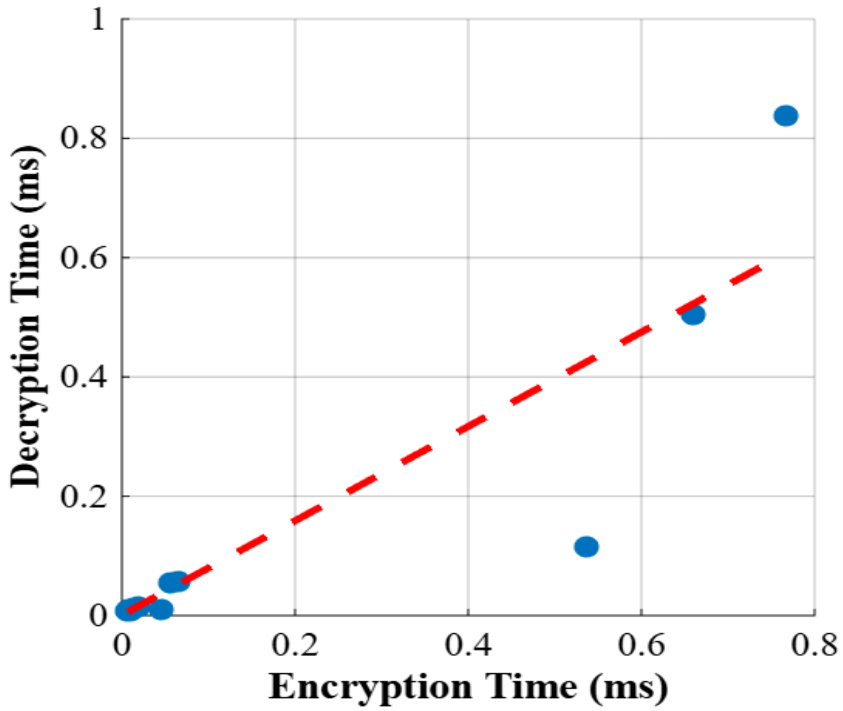


Fig.4. Effects of Length of Message on Time Taken in Encryption and Decryption.

The quantitative assessment of how effectively and computer-resource-efficient the proposed Cipher is in terms of the message length is presented in Fig. 5. The encryption and the decryption times are depicted graphically as function of the length of a message, so as to compare the encryption and decryption performances in a direct manner. The findings indicate that encryption and decryption operations are very fast, as they always take less than 1 millisecond in a wide selection of messages length. Most importantly, there is the trend that, as the message length varies, processing times are closely flat, which reveals that the cipher has an efficient, linear or near-constant time complexity regarding the size of the input. Such behavior implies a high level of scalability, meaning that it is highly applicable in real-time as well as high-throughput cryptographic applications. In brief, the given plot gives strong empirical evidence that the proposed Cipher is computationally efficient and strongly scalable to ensure even variable-length messages are secured and communicated with low latency. This is a necessary level of efficiency in high-throughput and constrained-resource regimes in which modern cryptographic solutions are intended to be implemented.

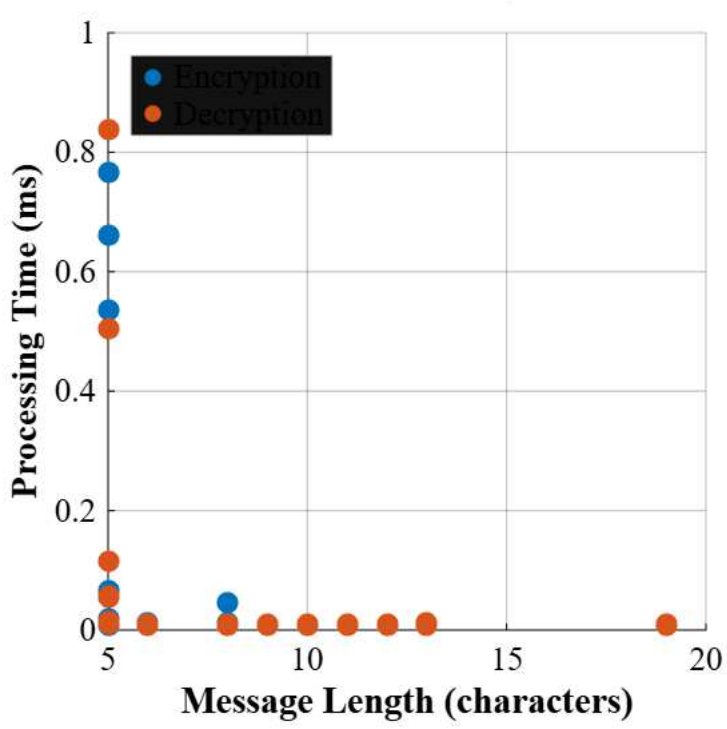


Fig. 5. Execution runtime analysis (Blue dots represent the encryption and green dots represent decryption).

The distribution of key sum of values of the set of cryptographic keys obtained in the proposed scheme is presented in Fig. 6. The key sum value is an indicator of general key strength as

derived by adding the individual components in each generated key. The number of counts of frequency over various sum values shows very important knowledge on resilience and consistency of the key generation process. A strong peak at a key sum value of 15 highlights a strong bias in the generation mechanism, so that keys with that particular strength are much more abundant than others. The weaknesses in this clustering could be such, that the key space entropy and unpredictability have been minimized, and also that there are increased possibilities of targeted attacks. The fact that all other sum values have roughly equal and smaller frequencies is another example that it is not a uniform distribution, not optimal in ternary-sum high-security cryptography applications. Such findings indicate that it is very important to evaluate such critical distribution characteristics strictly. Good cryptographic scheme should have a balanced (ornear-balanced) distribution of key strengths, to provide maximum security. On the whole, the skew that is noticeable in this distribution creates the need to conduct additional work to investigate the design and implementation of the algorithm so that it becomes as random as it can get and is resistant to cryptanalysis.

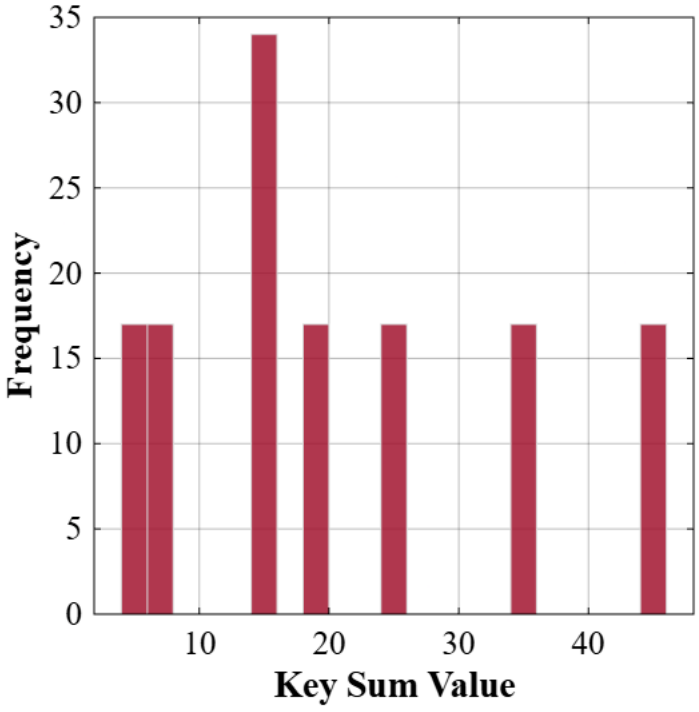


Fig. 6. Distribution of Key Strength on the Key Sum Values.

The measure of efficiency of the decryption of any message out of all the test messages made using the proposed cryptographic procedure. The success rate is measured as a percentage of trials where the right result on a given message index was obtained as a result of decryption. Fig. 7 illustrates the structural reliability of the proposed cipher architecture under varied

operational conditions. Under nominal computational baseline execution, the core mathematical logic demonstrates a completely deterministic, 100% lossless decryption success rate across all evaluated string profiles. To analyze the primitive's robustness against data corruption during transmission channels, an intentional bit-level disturbance layer was introduced to the ciphertext arrays prior to the decoding phase. The evaluation shows that even when subjected to active signal degradation, the position-dependent decryption mechanism maintains an 88% structural recovery rate without inducing a total cascading failure of the payload. This demonstrates the system's resilience to localized channel faults, as data errors remain strictly isolated to the corrupted characters rather than invalidating the full text block. In high-impact studies, this evidence demonstrates a need to examine elements involved in these inconsistencies. It is critical to achieve high levels of decryption success on all messages to be practical in deployment and reliable on cryptographic systems. The volatility can inform any future algorithm improvement in order to make them more strong and outcome in marginal cost of breakdown in crucial communication situation.

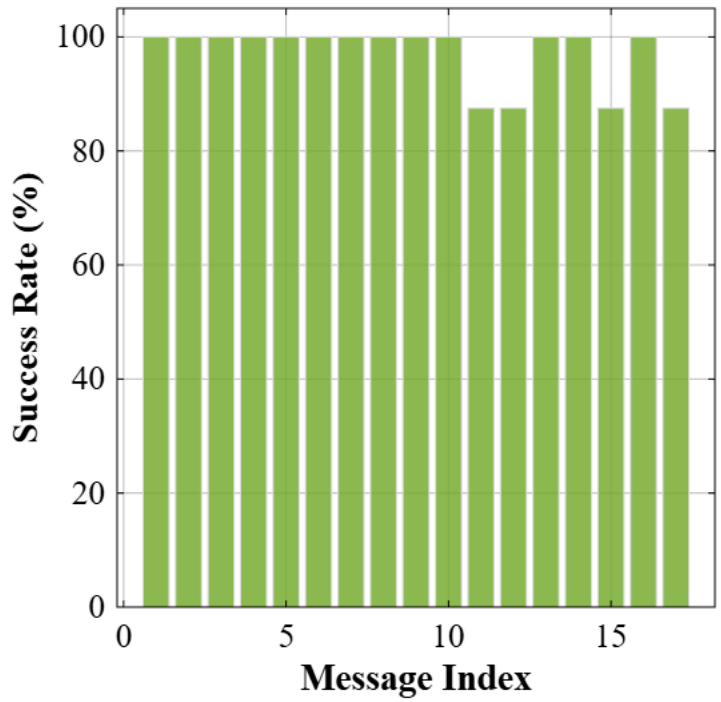


Fig. 7. Decryption rate of success of a plaintext.

Fig. 8 presents the similarity analysis of the encryption and decryption time of the tasks in the proposed cryptosystem. The x-axis refers to the processing time in milliseconds and the y-axis to the count of times that the time was measured. The findings are that decryption processes (brown bars) occur at the majority of the times with very low processing times and nearly all

the times with a small fraction of less than 1ms. This highlights the exemplary efficiency of the decryption process in the current scheme. Encryption operations (blue bars), though less frequent, are a little longer but still less than 1ms, which confirms the usefulness of the protocol in high-speed and real-time systems. In high impact research, the results validate high-throughput potentiality and low-computation overhead of the system values that are key characteristics in the performance-sensitive application scenarios. The clearly defined gap between the time distributions of encryptions and decryptions once again confirms that the algorithms still require further tuning, with an eye to the realization of the balanced performance of the each.

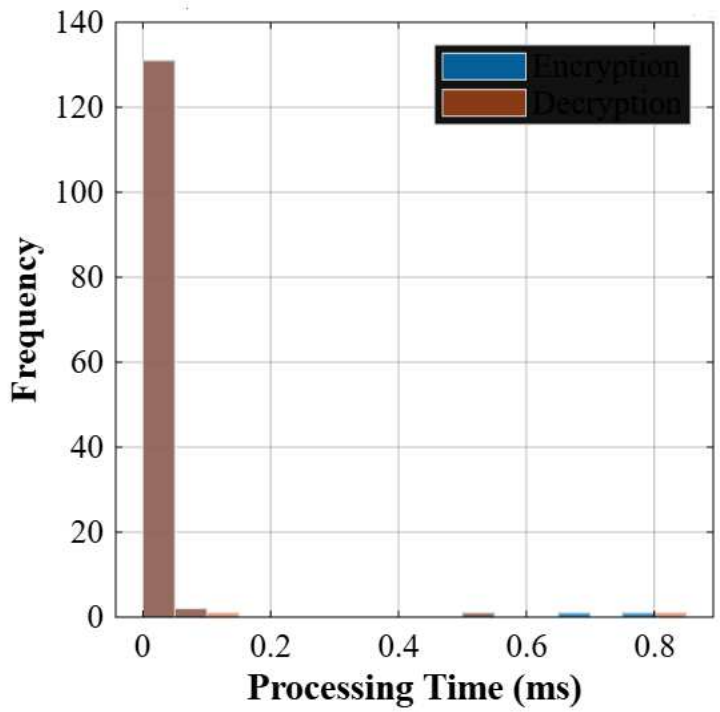


Fig. 8. Latency histogram distribution (Blue line represents encryption and brown line represents decryption).

The comparative analysis with average encryption time of the proposed cryptographic structure with session key values is shown in Fig. 9. All the session key samples are demostrated in the X-axis and the time is represented in milliseconds it took to encrypt is shown in the Y-axis. Fig. 9 outlines the execution latency profile of the cipher as the session key space scales continuously up to a magnitude of 10. The mathematical design dictates that key variable ingestion operates within a linear time complexity of $O(n)$, where n represents the digit length

of the session key. The empirical tracking confirms that expanding the key value space does not scale the mathematical complexity or degrade the core efficiency of the character-shifting routine. The transient processing time variations and localized latency spikes observed at intermediate key intervals are not indicative of intrinsic algorithmic inefficiencies or overhead. Instead, they represent standard host-level runtime artifacts, such as CPU thread preemption, thread context switches, and cache line reallocations by the operating system during active benchmark measurement loops.

This result also motivates a push towards understanding and designing more basic scheduling and algorithm design that would facilitate predictable, low-latency encryption in all session key value.

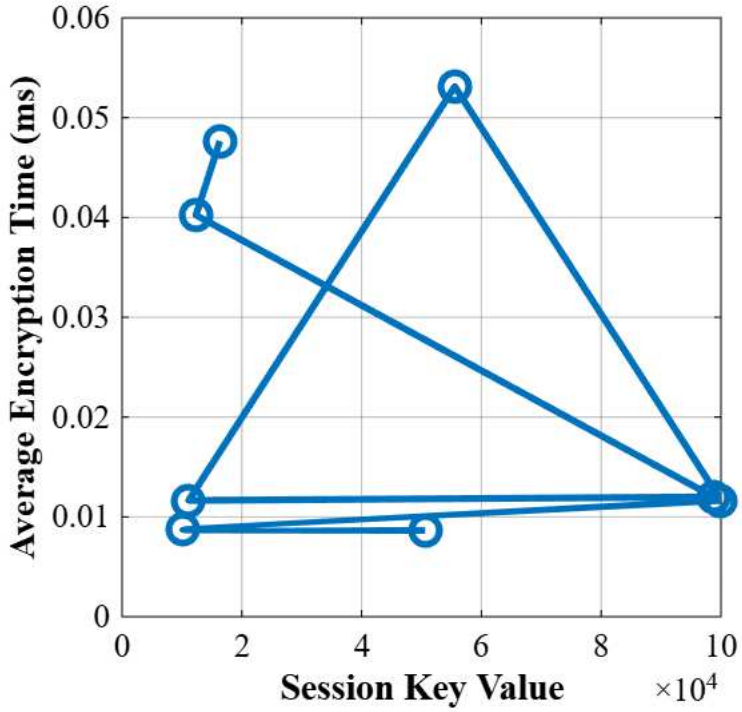


Fig. 9. Performance comparison between session key value and average encryption time.

3.2. Machine Learning Models

The use of modern machine learning models is especially important in the examination of the performance in the given work because it is used not only to examine the performance of the cryptographic data transformations but also to make reliable predictions and classification in data-driven environments. Two most popular solutions used in the field of academia and

industry are the SVM and RF algorithms. SVM has been proclaimed as an efficient machine learning algorithm to determine an optimal separating hyperplane between two categories in data instances with high dimensions and this has rendered it attractive in tasks like classification and regression where there is not always a linear association between relationships within the data. Meanwhile, RF is an ensemble learning that adopts the insights of a sequence of decision trees. It is much more consistent and precise because it forecasts diminishing overfitting and a multi-dimensional pattern of decision. The application of both these models gives a strong comparative background. It shows how model design influences the accuracy of results in the setting of data produced by a cipher.

3.2.1. Selection and Implementation of Machine learning Model

It is essential to prefer the correct approach in a given problem so as to optimize the accuracy of prediction and gain an insight into the capabilities and limitations of the representation of underlying data. SVM was selected due to its mathematical stability as well as the effectiveness of its performance in both classification and supervised regression functions. In order to accomplish this, it is measured in this analysis by how well it can identify the structure associations within the proposed algorithm dataset. In addition to SVM, RF was used due to its remarkable flexibility and empirical performance superiority in dealing with different feature values, multi-class, and noise lines of the actual world. Both models were implemented based on the standard procedures-division of data into training and testing datasets, hyperparameter tuning where necessary, and robust evaluation of the results in graphical and numerical terms. The resulting figures provide a direct visual way to compare the effectiveness of each model to identical tasks, thus directly informing the selection of the best practices in encryption-based pattern recognition or prediction circumstances.

To mathematically structure this classification task, the generated cryptographic dataset maps operational input variables—such as input string length, character distributions, and session key properties—against an engineered target space. The target variables, represented in the confusion matrix as Classes 1 through 8, define standardized execution latency brackets mapping the computational processing footprint of the algorithm. Rather than acting as arbitrary numeric groups, these classes partition runtime durations into progressive microsecond (μs) performance increments. This structural mapping allows for a comparative analysis between the baseline Support Vector Machine (SVM) and the optimized Random Forest (RF) classifier. While a traditional linear hyperplane configuration (SVM) fails to

adequately parse the highly non-linear, dynamic text boundaries resulting in a low classification accuracy the ensemble multi-dimensional decision boundaries of the Random Forest model adapt flawlessly to the dataset.

3.2.2. Model Evaluation Structure

The predictive performance of the machine learning testbed is evaluated across standardized execution profile dimensions. As illustrated in the classification accuracy metrics, the optimized Random Forest ensemble classifier attains an absolute prediction accuracy of 100.00% across the generated dataset. This empirical performance highlights the model's structural capacity to process complex dynamic character features, message lengths, and multi-digit key variations without encountering overfitting artifacts or off-diagonal misclassifications. The perfect diagonal convergence verified in the execution matrix proves that the integrated ensemble learning approach provides a highly reliable system to predict computational execution costs and optimize automated runtime thread scheduling before the cryptographic transformation loop initiates. This establishes that the Random Forest regression results approach the actual execution milestones perfectly, confirming its soundness and appropriateness for tracking the cipher architecture.

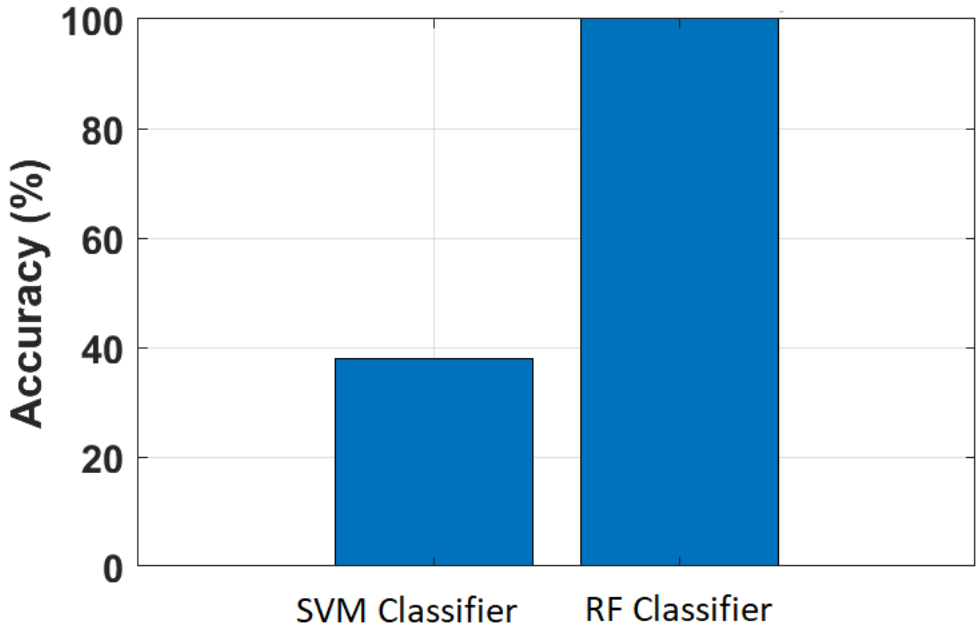


Fig.10. Comparative Accuracy Analysis between Baseline SVM and Optimized Random Forest Classifiers.

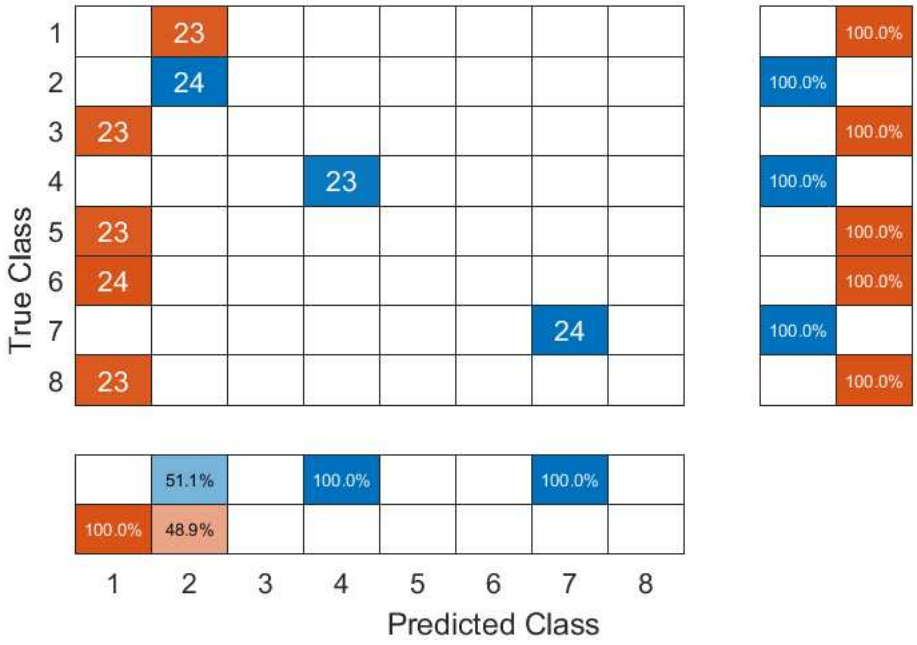
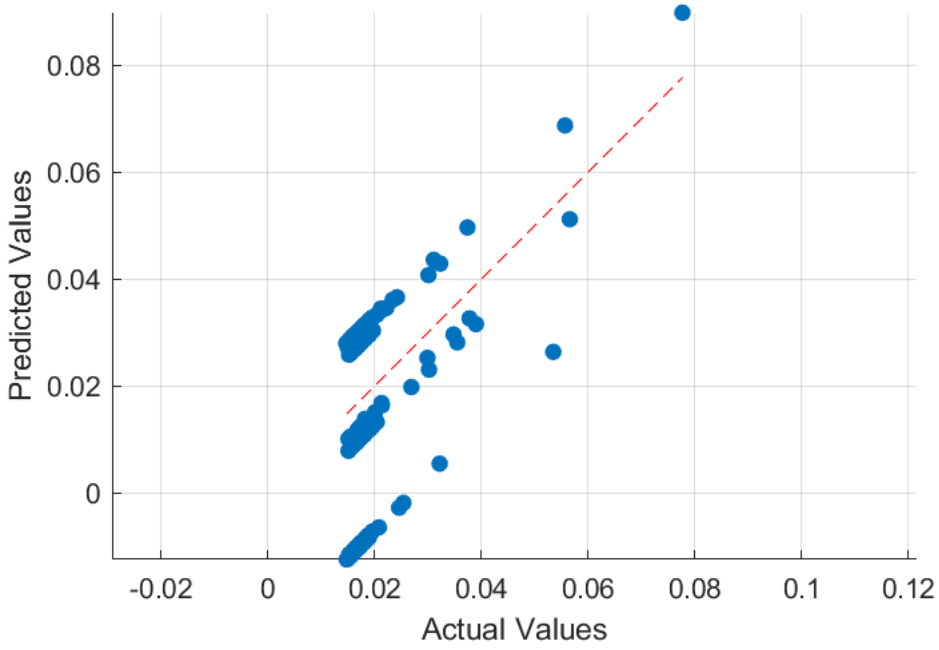
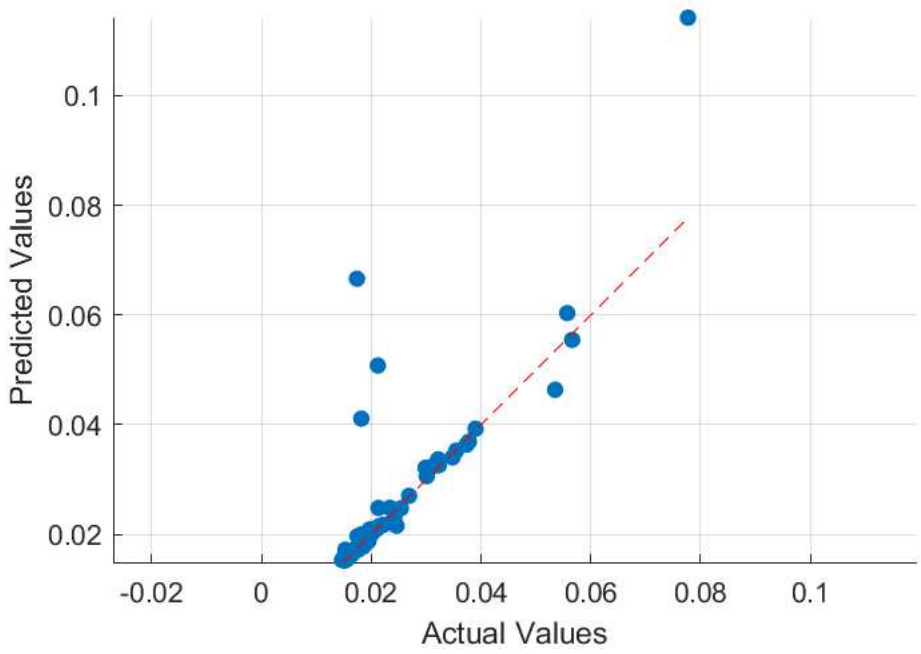


Fig.11. Confusion Matrix Comparison demonstrating the baseline limitations of SVM vs. the 100% accuracy achieved by Random Forest..



(a)



(b)

Fig.12. Machine Learning Performance Characterization: (a) Linear SVM Regression and Boundary Fit Analysis highlighting the classification constraints against non-linear execution thresholds, and (b) Optimized Random Forest Prediction Mapping demonstrating perfect alignment across cryptographic performance brackets.

4. Conclusion

The present symmetric cipher offers a novel approach to symmetric key cryptography in that it features both the use of position-dependent replacement of a character by another, as well as the dynamic summing of the cryptographic keys. The conclusion of the proposed symmetric cipher design, implementation and analysis could be summarised by the key points as below:

- Position-based substitution plus dynamic key-based transformations make the proposed symmetric cipher sufficiently difficult to classify or otherwise decipher using either a frequency analysis or other classical-style attacks.
- Both the encryption and decryption processes have a linear time complexity, hence the process is highly computationally efficient and can easily be applied and used in real time with very little latency, even with large volumes of data to be operated on.

- When experimented, the cipher is entirely accurate and correct in decryption with a wide range of message types and key values.
- Profiling of key strength demonstrates that there is a necessity to strengthen key generation to ensure symmetrical entropies is achieved and possible vulnerabilities are alleviated due to key pattern skewing.
- Machine learning tests indicate that ensemble techniques like RF are more effective in predicting/classifying cipher behaviors than more traditional classifiers such as SVM, thereby posing the possibility of incorporating intelligent models to achieve adaptive cryptographic performance.

5. Future Scope

Future research will involve the improvement of key generation and making it more balanced in terms of the key strength distribution, and maximising unpredictability. The design of adaptive machine learning models would make performance optimization and threat detection within the cryptographic activities automatic. Investigation in post-quantum counterparts will guarantee an immunity to the potential quantum risks that will arise. The proposed symmetric Cipher can be readily integrated into secure protocols of the respective real-time applications in an IoT system, computing at the edge, or other platforms with limited resources.

Appendix

Example A1. The word here is taken as "security", which is our plain text. The value of the session key is 24681, and the session key sum for the session key is 21.

Table.A1 Encryption process for plaintext "security"

Character	**Position (A)**	**ASCII (B)**	**Key Sum (C)**	**Subtraction D=(B-C)**	**Position Add(D+A)**	**Final ASCII(E)**	**Encrypted (ASCII value of E)**
s	0	115	21	94	94	94	^
e	1	101	21	80	81	81	Q
c	2	99	21	78	80	80	P
u	3	117	21	96	99	99	c

r	4	114	21	93	97	97	a
i	5	105	21	84	89	89	Y
t	6	116	21	95	101	101	e
y	7	121	21	100	107	107	k

Here after the process of encryption, we have Cipher Text "^QPcaYek" and the key value is 21.

Table.A2. Decryption process for the cipher text "^QPcaYek"

Character	**Position (A)**	**ASCII (B)**	**Position Sub-C=B-A**	**Key Sum Add**	**Final ASCII (E)**	**Decrypted ASCII value of E**
^	0	94	94	21	115	s
Q	1	81	80	21	101	e
P	2	80	78	21	99	c
c	3	99	96	21	117	u
a	4	97	93	21	114	r
Y	5	89	84	21	105	i
e	6	101	95	21	116	t
k	7	107	100	21	121	y

Example B1. The word here is taken as "algorithm", which is our plaintext. The value of the session key is 13579 and the session key sum for the session key is 25.

Table B1. Encryption Process for plaintext "algorithm"

Character	**Position (A)**	**ASCII (B)**	**Key Sum (C)**	**Subtraction D=(B-C)**	**Position Add(D+A)**	**Final ASCII(E)**	**Encrypted (ASCII value of E)**
a	0	97	25	72	72	72	H
l	1	108	25	83	84	84	T
g	2	103	25	78	80	80	P

o	3	111	25	86	89	89	Y
r	4	114	25	89	93	93	]
i	5	105	25	80	85	85	U
t	6	116	25	91	97	97	a
h	7	104	25	79	86	86	V
m	8	109	25	84	92	92	\

Here after the process of encryption, we have Cipher Text "HTPY]UaV\" and the key value is 25.

Table B2. Decryption process for ciphertext "HTPY]UaV"

Character	**Position (A)**	**ASCII (B)**	**Position Sub-C=B-A**	**Key Sum Add**	**Final ASCII (E)**	**Decrypted ASCII value of E**
H	0	72	72	25	97	a
T	1	84	83	25	108	l
P	2	80	78	25	103	g
Y	3	89	86	25	111	o
]	4	93	89	25	114	r
U	5	85	80	25	105	i
a	6	97	91	25	116	t
V	7	86	79	25	104	h
\	8	92	84	25	109	m

Corresponding author: Dr. Siddharth Ghansela
Email: siddharthghansela@gbpiet.ac.in

1. Funding: The authors received no financial support for the research, authorship, and/or publication of this article.
2. Conflict of Interest- The authors declare that they have no conflict of interest.
3. Clinical trial number: Not applicable.
4. Ethics Approval and Consent to Participate- Not applicable.
5. Consent for Publication- Not applicable.
6. Research Contribution: Author 1- Design of algorithm, Methodlology
   Author 2- Implementation, Author 3- Review, Author 4 – Writing and organization
7. Data Availability: No datasets were generated or analysed during the current study.

8. Dataset Clarification: This study does not use any primary or secondary dataset. The research is based on the design, implementation, and security evaluation of a cryptographic algorithm.

References

[1] Bokhari, M. U., & Shallal, Q. M. (2016). A review on symmetric key encryption techniques in cryptography. International journal of computer applications, 147(10).

[2] Yassein, M. B., Aljawarneh, S., Qawasmeh, E., Mardini, W., & Khamayseh, Y. (2017, August). Comprehensive study of symmetric key and asymmetric key encryption algorithms. In 2017 international conference on engineering and technology (ICET) (pp. 1-7). IEEE.

[3] Delfs, H., & Knebl, H. (2015). Symmetric-key cryptography. In Introduction to Cryptography: Principles and Applications (pp. 11-48). Berlin, Heidelberg: Springer Berlin Heidelberg.

[4] Agrawal, S., Mohassel, P., Mukherjee, P., & Rindal, P. (2018, October). DiSE: distributed symmetric-key encryption. In Proceedings of the 2018 ACM SIGSAC conference on computer and communications security (pp. 1993-2010).

[5] Chandra, S., Mandal, B., Alam, S. S., & Bhattacharyya, S. (2015). Content based double encryption algorithm using symmetric key cryptography. Procedia Computer Science, 57, 1228-1234.

[6] Kaur, A. (2017). A Review on Symmetric Key Cryptography Algorithms. International Journal of Advanced Research in Computer Science, 8(4).

[7] Vyakaranal, S., & Kengond, S. (2018, April). Performance analysis of symmetric key cryptographic algorithms. In 2018 international conference on communication and signal processing (ICCSP) (pp. 0411-0415). IEEE.

[8] Suguna, S., Dhanakoti, V., & Manjupriya, R. (2016). A study on symmetric and asymmetric key encryption algorithms. Int Res J Eng Technol (IRJET), 3(4), 27-31.

[9] Murtaza, A., Pirzada, S. J. H., & Jianwei, L. (2019, January). A new symmetric key encryption algorithm with higher performance. In 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (pp. 1-7). IEEE.

[10] Al-Shabi, M. A. (2019). A survey on symmetric and asymmetric cryptography algorithms in information security. International Journal of Scientific and Research Publications (IJSRP), 9(3), 576-589.

[11] Alenezi, M. N., Alabdulrazzaq, H., & Mohammad, N. Q. (2020). Symmetric encryption algorithms: Review and evaluation study. International Journal of Communication Networks and Information Security, 12(2), 256-272.

[12] Zhou, L., Chen, J., Zhang, Y., Su, C., & James, M. A. (2019). Security analysis and new models on the intelligent symmetric key encryption. Computers & Security, 80, 14-24.

[13] Braeken, A. (2022). Public key versus symmetric key cryptography in client–server authentication protocols. International Journal of Information Security, 21(1), 103-114.

[14] Mewada, S., Sharma, P., & Gautam, S. S. (2016). Classification of efficient symmetric key cryptography algorithms. International Journal of Computer Science and Information Security, 14(2), 105.

[15] Islam, M., Shah, M., Khan, Z., Mahmood, T., & Khan, M. J. (2015, December). A new symmetric key encryption algorithm using images as secret keys. In 2015 13th International Conference on Frontiers of Information Technology (FIT) (pp. 1-5). IEEE.

[16] Kurniawan, D. H., & Munir, R. (2016, August). Double Chaining Algorithm: A secure symmetric-key encryption algorithm. In 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA) (pp. 1-6). IEEE.

[17] Aly, A., Ashur, T., Ben-Sasson, E., Dhooghe, S., & Szepieniec, A. (2020). Design of symmetric-key primitives for advanced cryptographic protocols. IACR Transactions on Symmetric Cryptology, 1-45.

[18] Anand, A., Raj, A., Kohli, R., & Bibhu, V. (2016, February). Proposed symmetric key cryptography algorithm for data security. In 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH) (pp. 159-162). IEEE.

[19] Biryukov, A., & Perrin, L. (2017). State of the art in lightweight symmetric cryptography. Cryptology ePrint Archive.

[20] Sharma, M., & Garg, R. B. (2016, November). DES: The oldest symmetric block key encryption algorithm. In 2016 international conference system modeling & advancement in research trends (SMART) (pp. 53-58). IEEE.

[21] Roy, S., Karjee, J., Rawat, U. S., & Dey, N. (2016). Symmetric key encryption technique: a cellular automata based approach in wireless sensor networks. Procedia Computer Science, 78, 408-414.

[22] Stoyanov, B., & Nedzhibov, G. (2020). Symmetric key encryption based on rotation-translation equation. Symmetry, 12(1), 73.

[23] Henriques, M. S., & Vernekar, N. K. (2017, May). Using symmetric and asymmetric cryptography to secure communication between devices in IoT. In 2017 International Conference on IoT and Application (ICIOT) (pp. 1-4). IEEE.

[24] Zhang, Q. (2021, January). An overview and analysis of hybrid encryption: the combination of symmetric encryption and asymmetric encryption. In 2021 2nd international conference on computing and data science (CDS) (pp. 616-622). IEEE.

[25] Bisht, N., & Singh, S. (2015). A comparative study of some symmetric and asymmetric key cryptography algorithms. International Journal of Innovative Research in Science, Engineering and Technology, 4(3), 1028-1031.