International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 4, August 2024, pp. 4306~4314
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i4.pp4306-4314

Advancing cryptographic security: a novel hybrid AES-RSA model with byte-level tokenization

Renuka Shone Durge, Vaishali M. Deshmukh
Department of Computer Science and Engineering, Prof. Ram Meghe Institute of Technology and Research, Badnera-Amravati, India

Article Info

Article history:

Received Feb 8, 2024
Revised Apr 6, 2024
Accepted Apr 16, 2024

Keywords:

AES algorithm
Byte level tokenizer
Encryption algorithms
RSA algorithm
Tokenization

ABSTRACT

As cyberattacks are getting more complex and sophisticated, stringent, multi-layered security measures are required. Existing approaches often rely on tokenization or encryption algorithms, both of which have drawbacks. Previous attempts to ensure data security have primarily focused on tokenization techniques or complex encryption algorithms. While these methods work well on their own, they have proven vulnerable to sophisticated cyberattacks. This research presents new ways to improve data security in digital storage and communication systems. We solve data security issues by proposing a multi-level encryption strategy that combines double encryption technology along with tokenization. The first step in the procedure is a byte-level byte-pair encoding (BPE) tokenizer, which tokenizes the input data and adds a layer of protection to make it unreadable. After tokenization, data is encrypted using Rivest–Shamir–Adleman (RSA) to create a strong initial level of security. To further enhance security, data encrypted with RSA has an additional layer of encryption applied using the advanced encryption standard (AES) method. This article describes how this approach is implemented in practice and shows how it is effective in protecting data at a higher level than single-layer encryption or tokenization systems.

Corresponding Author:

Renuka Shone Durge
Department of Computer Science and Engineering, Prof. Ram Meghe Institute of Technology and Research
Anjangaon Bari Road, Badnera-Amravati, Maharashtra, India
Email: renuka434@gmail.com

1. INTRODUCTION

The growing intricacy of cyberattacks underscores the pressing want for data in the era of digitalization. Traditional encryption methods, which are primarily dependent on data security, are becoming less effective in the face of quantum and increasingly sophisticated cyberattacks. This study identifies a critical flaw in the security system as it stands and suggests a fix. Our byte-pair encoding (BPE) tokenizer is trained at the byte level using the giga word and book corpus datasets. This tokenizer produces a robust fusion model when coupled with Rivest–Shamir–Adleman (RSA) and advanced encryption standard (AES) encryption methods. This paradigm seeks to enhance cryptographic security beyond what is currently achievable while offering defense against attacks utilizing both classical and quantum computing. Our contribution is twofold: first, we strategically combine the RSA and AES algorithms to construct a cryptographic framework, and second, we creatively use connected datasets to better train the tokenizer. Encryption [1] turns plain text into perplexing visual material by employing computations and a key. As though the data assessed by this component was available to the design collector. Using the key to return the image's contents to their original form is the opposing tactic, referred to as decryption. The encryption's level of security is determined by the computational power and key secret. By functioning as a secret code, encryption shields private messages and bank account proof of identity (POI) from unauthorized users when they are sent across a network. Even while information has always been protected by encryption, unexplored technologies like quantum computing are making it more difficult to violate these agreements.

Long considered the cornerstone of data security, encryption is the process of converting readable data into an unintelligible form that can only be decrypted by authorized personnel with the required decryption key. But not all encryption algorithms are created equal, and variables like algorithm complexity and key strength frequently affect how successful an algorithm is. Cyberattacks are becoming more sophisticated as digital technologies develop, putting people, companies, and governments at risk of identity theft, illegal data access, and data breaches. Standard encryption algorithms like RSA [2] and AES [3] are frequently used in traditional data security techniques. Even with their relative effectiveness, these strategies are not immune to the constantly changing strategies used by cybercriminals. The cornerstones of data security, encryption mechanisms, are not infallible. The security of RSA, a widely used public-key cryptosystem, hinges on the challenge of computing large integers. Nonetheless, the susceptibility of the RSA algorithm has been exposed with the emergence of quantum computing. Likewise, AES, known for its speed and security, has limitations, particularly concerning key management and the potential for side-channel attacks. Using the widely known book corpus dataset and the giga word dataset, we describe a robust technique in this study. The following is a description of this paper's contributions: i) Giga word and book corpus datasets combined to train the byte level BPE tokenizer, ii) Building a fusion model that combines the RSA, AES, and byte-level BPE tokenizer algorithms.

2. THE COMPREHENSIVE THEORETICAL BASIS AND THE PROPOSED METHOD

In the modern world, where we utilize numerous technologies to share and save data, protecting your data is crucial. We must make sure that this information is safe from any threats and that only authorized individuals have access to it. As technology develops, we must find more effective methods to safeguard our data. To safeguard our data, we must combine several security measures because relying solely on one is insufficient. Information encryption by the old method is not as secure as it once was, and knowledgeable hackers can simply break it. Even the most robust encryption algorithms, such as RSA and AES, have specific flaws that leave them open to kinds of cyberattacks. Research by Roy *et al*. [4]. give the complete study's conclusions regarding the special traits, robustness, and more effective encryption keys that are appropriate for the AES and RSA algorithms, as well as how to combine them to create a potent security architecture and using tokenization for virtual storage. Vahdati *et al.* [5] proposes a novel technology using RSA algorithm that has an impact in IIoT industry. There was another study by Mojisola *et al*. [6] that tackles the RSA algorithm limitations by introducing more complexity and randomness to data, where this data is fed into the algorithms along with a greater number of RSA keys. Shivaramakrishna and Nagaratna [7] proposed a novel hybrid encryption model with a combination of RSA and AES-OTP that helps in protecting cloud solutions. This solution uses asymmetric encryption for data privacy and helps in data security and privacy. Another work by Alfani *et al.* [8] proposed a combination of AES algorithm and digital signature algorithm (DSA) that safeguards the textual data where AES algorithm is used for encrypting and decrypting data and DSA is used for authenticating the data using hash algorithm.

Alani [9] analyzed different techniques for encryption and decryption where some of the techniques include machine learning based assaults, algorithm cryptanalysis, and encryption algorithms. Pandey *et al*. [10] proposed a new method for breaking optical encryption methods that do not involve accessing plaintext pairs. The authors demonstrate that two-dimensional deep learning techniques can be used to predict target words from unknown text in real time. To obtain clear words, this method requires developing a virtual optical encryption system to collect training data and using two deep neural networks (DNNs) for decoration and denoising. Kuppuswamy *et al.* [11] proposed a hybrid encryption approach that combines RSA with a novel symmetric key algorithm where this technique, uses both symmetric and asymmetric encryption techniques to increase data securities, One such researcher Sood *et al*. [12] provide a detailed comparison of the encryption algorithms RSA, DES, and AES with respect to their merits and demerits under different scenarios. Akter *et al*. [13] proposes a combination of AES-128 with RSA hybrid model that aims to protect data in cloud applications by using hash-based message authentication code (HMAC) for ensuring authenticity and integrity of data. Sangwan [14] uses a hybrid combination of block type symmetric key algorithm and Huffman algorithm to encrypt the text where two private keys are used. Wang and Li [15] suggest test equipment description language (TEDL) a new symmetric text encryption technique which employs word vector tables and hyperparameters with deep learning to convert input data into encrypted text. Jagadeesh *et al*. [16] proposes a hybrid encryption This approach aims to optimize encryption and decryption times by utilizing the strengths of both AES and elliptic curve cryptography (ECC), offering a smaller key size and increased efficiency in cloud data security. Chakkaravarthy [17] created a hybrid model which combines blockchain and AES and its purpose is to improve medical application data security through ensuring the privacy and accuracy of patient data accessed or stored in the cloud. Samanth *et al*. [18] explored a lightweight encryption technique named CLEA-256 for secure encryption of text messages as well as images in an internet of drones (IoD) networks' scenario. It explains how CLEA-256 is suitable for Internet of Things applications with resource-limited devices because it provides higher security and performance than traditional algorithms especially for aerial data and standard photos. Singh and Garg [19] presents a symmetric key encryption algorithm which uses ASCII values of text data for encrypting information securely. This way represents an easy yet effective solution to encrypting text information through turning plain text into cypher text with a randomly generated key used in order to protect any transmitted message.

From 2018 to 2022, Kalvikkarasi and Saraswathi [20] compares hybrid cryptography techniques. The aim is to increase the security of data in cloud computing by merging the effectiveness of symmetric and ease of use of asymmetric cryptography as well as providing an answer to researchers who are investigating cloud security. A study by Srikumar and Pande [21] suggests that evolutionary algorithm became a well-established field during 1990s and 2000s, which are seen as its mature years. Yet it was different from when scientists began using EAs for solving various problems such as those in data mining. According to Sriram *et al*. [22], blockchain technology can help improve cybersecurity in different types of businesses too. The text points out that this technology makes blockchain hard for unauthorized parties to tamper with or delete information through secure management, storage, and distribution of it. Mohammed *et al.* [23] presents an enhanced AES algorithm specifically designed for secure storage of images in clouds. It deals with integrating advanced encryption methods and digital signatures for protecting private photographs against unwanted viewing and manipulation, while optimizing AES for improved security and performance within cloud environments. Radhi and Ogla [24] gives a detailed analysis on these algorithms and concludes that Blowfish is the most efficient algorithm as it is good in terms of time as well as memory usage. Pande *et al*. [25] present a publicly verifiable threshold key sharing technique to study the security of consumers' personal keys in blockchains. Collaborating nodes verify the added keys upon receipt. When sharing a key, the middle-shared node does not work properly.

3. METHOD

We used a combination of two datasets i.e. giga word and book corpus. Giga word dataset consists of news data extracted from news websites and sources. Table 1 displays an example of a giga word dataset. While book corpus dataset consists of data from collection of books, it has numerous genres and it is very useful in encryption tasks the book corpus dataset is exemplified in Table 2.

Table 1. Example of giga word dataset

Title	Global Markets Respond to Economic Summit
Content	In the wake of the recent economic summit, global markets have shown a mixed response. While some indices rallied on news of renewed trade agreements, others dipped amidst concerns over rising oil prices

Table 2. Example of book corpus dataset

Title	Shadows in the Twilight
Content	As the sun dipped below the horizon, casting elongated shadows across the cobblestone streets, Elara quickened her pace. The air was thick with the scent of rain, and distant thunder whispered promises of a storm.

The combination of giga word and book corpus provides a robust and varied textual foundation for the research. The method starts with the raw input text that needs to be secured in the suggested architecture shown in Figure 1. The text is initially tokenized into sub word units by the model using a byte level BPE tokenizer. By taking this step, the text data becomes less complicated, and the encryption process becomes more efficient. The RSA technique is used to encrypt the tokenized output. A public key is used for encryption and a private key is used for decryption in RSA. By transforming the tokenized data into a ciphertext that can only be decrypted with the matching private key, the RSA encryption secures it.

The encryption key required for the subsequent encryption layer can be safely transmitted using this encryption layer. Both original and decrypted text are passed from MD5 checksum which is a cryptographic hash function that produces 128-bit hash value. It is used to verify the data integrity. For the above methodology, given below is the algorithm of text encryption and decryption process.

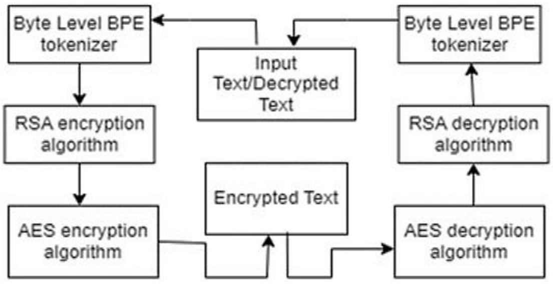

Figure 1. Proposed architecture

Algorithm. Text encryption and decryption process

```
Input: Original Text
Output: Decrypted Text
Function md5_checksum(string):
      return haslib.md5(string.encode(┤)).hexdigest()
#Calculate MD5 checksums
checksum1=md5_checksum(Input)
Function TokenizeText(input_text):
           input_text←input_text.strip()
           input_text←" ".join(input_text.split())
           tokenizer←ByteLevelBPETokenizer()
           encoded_output←tokenizer.encode(input_text)
           tokenized_ids←encoded_output.ids
           Return tokenized_ids
#RSA Key Generation and Serialization
Output: private_key (RSA Private key), public_key (RSA Public Key)
Funcrion GenerateSerializeRSAKeys():
             private_key ← rsa.generate_private_key(key_size=2048)
            public_key ← private_key.public_key()
            Return (private_key,public_key)
#RSA Encryption
Input: tokenized_ids, public_key
Output: rsa_ciphertext
Function RSAEncrypt(tokenized_ids,public_key ):
          #Convert tokenized ids to bytes
          byte_data←str(tokenized_ids.encode()
          #Encrypt using public key
          rsa_ciphertext←public_key.encrypt(byte_data,padding.OAEP()
          Return rsa_ciphertext
#Fernet Key Generation and Encryption
Input:rsa_ciphertext
Output: fernet_ciphertext
Function FernetEncrypt(fernet_ciphertext):
          #Encrypt using Fernet
          cipher_suite←Fernet(key)
          fernet_ciphertext←cipher_suite.encryption(rsa_ciphertext)
          Return fernet_ciphertext
#Full Encryption Process
Input: input_text (String)
Output: fernet_ciphertext
Function FullEncryptionProcess(input_text):
          tokenized_ids←TokenizeText(input_text)
          (private_key,public_key)←GenerateSerializeRSAKeys()
          rsa_ciphertext←RSAEncrypt(tokenized_ids,public_key)
          ferner_ciphertect←FernetEncrypt(rsa_ciphertext)
          Return fernet_ciphertext
#Full Decryption Process
Function FullDecryptionProcess(fernet_ciphertext,private_key ):
             key_file_path←'secret_key.key'
             #Decrypt using Fernet
              cipher_suite←Fernet(key)
               rsa_ciphertext←cipher_suite.decrypt(fernet_ciphertext)
               #Decrypt using RSA
               decrypted_data←private_key.decrypt(rsa_ciphertext,padding.OAEP()
               original_text←tokenizer.decode(decrypted_data)
               Return original_text
#Calculate MD5 checksums
```

```
checksum2=md5_checksum(Input)
if(checksum1==checksum2):
     print("Both encrypted text and decrypted text are the same)
```

4. RESULTS AND DISCUSSION

The result of the model demonstrates the effectiveness of layered encryption in providing robust security. The fixed-length output of RSA encryption, followed by the additional layer of Fernet encryption, ensured both the security of the encrypted data and the integrity of the encryption process. For example: “Sky is Beautiful”. Then the input text “Sky is Beautiful” is tokenized using a byte level BPE tokenizer. The tokenized output is:

```
Tokenized output: [55,2034,381,12447,462,1386]
```

This tokenization converts the text into a sequence of token IDs which are numerical representation of the text. Followed by this the tokenized output is encrypted using RSA. RSA encryption with a key size of 2048 bits produces a cyphertext of 256 bytes, regardless of the input size. The output of the RSA encryption is given below.

```
gAAAAABln6v2_Pd47Ep6Lu3gmpJgSWGnJvwGL9JKXtbUwX2ZVoJu2F90qAcfXOuacijxOlL3SlbwX4zmMFY1e
R4TXC8O3QvpgZtxD4o8IAILVoNB0b1SlIEFAMQPBPwVNIwSuD0Syyr6_bEgM5U6cQmq_seN4QeT__ZQOUNhP9
eP8bNm1MM7Vwzih10xs891EOzXfSxqtIsHmYVS1DfjMH_kqS2qJZHGAsThTNJolWzHzwLZZErh8CHeu3bv-
thDqvUL_259c4DhIsrRMbXNB13jqcshCZBdotOo3bvHOBA99Pw2Nobv2x1vesONTcRmzM8EyPY9yIdZjm9szw
J_2411yfmm0Gxo0pq5W7a1SrOWh3rD9uNha_ULSzwfaJnMzhCvG1K5vOT0OEead_qnnBRQPEyWZ8vidmE79Pv
jqf0C9TF8=
```

The RSA encrypted data is further encrypted using AES in cipher block chaining (CBC) model with 128-bit block size. The length of Fernet ciphertext is 440 bytes. The length is larger than the RSA ciphertext due to fernet encryption adds a version byte, timestamp, initialization vector and HMAC for integrity checking. Furthermore, the Fernet uses padding which increases the size, and the output of the encryption process is given below:

```
gAAAAABln6xBOUfprv1_Oagi3CNKWJph0XRNtKoRTv3RieYUxOSfv5Ru_rIpxUGDpVK
ReM9fPyHeB7dVV6H6B3rEG2YayvvFxlkluOofTFkeG2NhjXwCrwOoLACWLwPUFrPzM7
IcXR2jZHKlT_apeKgTTn7767S0zv11b-284gQ164xJLVaOXDIl7zwrqljFSPEZxL-
3G9qStjT1QjqFCgB4Tax83YywkvvVJ2XelbUbMdZ4J3329IASuagbtQwn7cdSSRO2TD
o2E4ma7Weiao7agoe6zMacvRT9e_IyR9BirM1qRnZaeKvqzjeFHUlRvGsuxbnwTLh1I
Obwt5qH_owMbS-O8twO6O3IPdJ9rxRs9df0_IUh50lTrZgOSKX-
rxeUueGalUryVhd_CGHuN25aLzD-S0U72b7AJG-K5Fz0gCsueUoj27A=
```

Now the process of decryption, we need to decode the cipher text obtained from AES to get the cipher text of RSA. The decrypted AES cipher text is given below.

```
\x86E%\xa2\xac\xb2de\x0b\x8dgO\xf9\xd0*\xce\x1f\xf9P\x9f9\xcez\xc5\xb2\'\xc1eWR\x86k/\xc6\x
dd\x0f\xdc\x1b\'\rK.\x13\x95\x13\x87}\x835B\xd3\x0et\xf1\xf8\xcc\x96\xe4\xacu\x93\x9682B\xe
e\xf1>\x82\xbb,\x1b\x93H\x00\xb3\x8am\xa2`\x80\xdcF\xa3\x0e\x11\xa9\xf2\xf6\xb3\x9c\xca\xd5
\x15\xbb\x95\x81ve\x05D\xd5\x9bH\xc4\x83\xdd\xf9\xf3\xda\xc5n%\xeb\xa1h\x1e\x7f*\x91Jd\x90\
xe8\xd6\xf0yko\xd7\xf7\xc9\x0c\xd9\xc1\x876\xe13\x1c\xf9$\xcc\xfb\xe7A\x7fY\x1a\x1b\xcf\x10
\\s\xaae\t\xf1\xb2z%\xa28\\\x11\xae\x18\xf1\x08\'G\xa9\xff\xc2Ch\xe52q\x82\x83\xa5\x9c\xba\
xab\x1f\x000P\x06\xa3\x00b\xb6q\x02\x1e\xb3$\xf6"\xbf\xc9%\xb5\x15WG\xcb\xb8\xbe\x8c\xf6\xe
a\xdc\x01!\xda\xca\x94\xa1\x86)\x04\xe11XR\x81\xf3\xb8+-
\xfe\x82\xd5`\r\xf8\xcb\x9b\x1f\xe3=\x13j\rE\x92%\xf4\x99p\x9a\xf7
```

Clearly the decrypted cipher text is completely different from AES encrypted cipher text. The decrypted RSA ciphertext is then decrypted using the RSA private key to get back to the original tokenized output.

```
Tokenized output: [55,2034,381,12447,462,1386]
```

Finally, the tokenized output obtained from RSA decryption is converted back into the original text using the tokenizer, where each numerical token ID is mapped back to its corresponding word or sub word unit. The text "Sky is Beautiful" is converted into tokens, encrypted with RSA, then encrypted again with AES for added security. To read the message, the process is reversed: AES decryption is applied, followed by RSA decryption, and then the tokens are converted back into the original text.

Both the MD5 checksums used returned a fixed size string of bytes and were equal. The significance of MD5 lies in its ability to produce a unique 'fingerprint' of a file or a string of text. If even a single bit in the input changes, the resulting hash will be completely different. This makes it a valuable tool for verifying integrity. Figure 2 illustrates the time taken for encryption varies with the input data size. An increase in data size usually leads to an increase in encryption time. Here RSA encryption algorithm which is an asymmetric encryption algorithm and is slower due to its mathematical computation. Figure 3 illustrates the data size vs decryption time graph i.e. how the decryption time changes with respect to the size of the input data. The graph has a downward trend, it suggests that decryption time decreases as the input size decreases. However, the slight upward trend indicates that larger input sizes increase the decryption time. Figure 4 illustrates the encryption and decryption time taken for each stage of the encryption and decryption process. RSA decryption stage takes the maximum time to decrypt the data. There are three sample texts given to the model and Figure 5 is the Token distribution across different text samples.

```
Text samples   1. "Short text"
               2. "a bit longer text"
               3. "this is a much longer text"
```

From Figure 5 the frequency of Token IDs does not directly affect the length of the encrypted text when using RSA encryption as RSA produces a fixed-size output based on the key size, regardless of the input size. However, the token frequency distribution potentially impacts the time taken to encrypt the text.

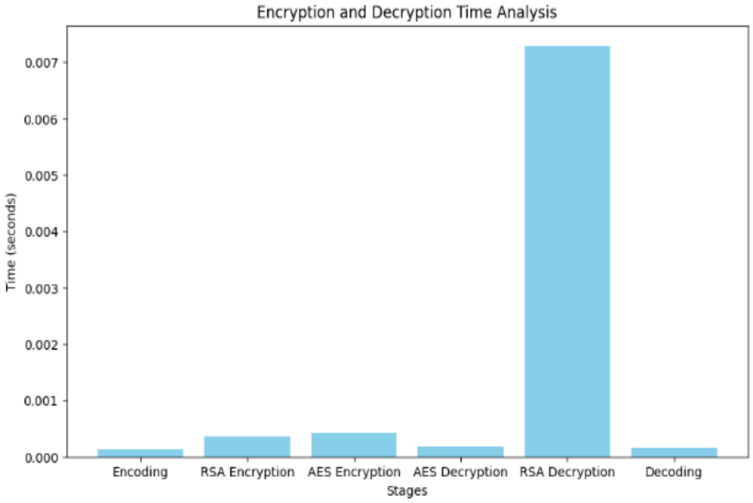

Figure 2. Data size vs encryption time

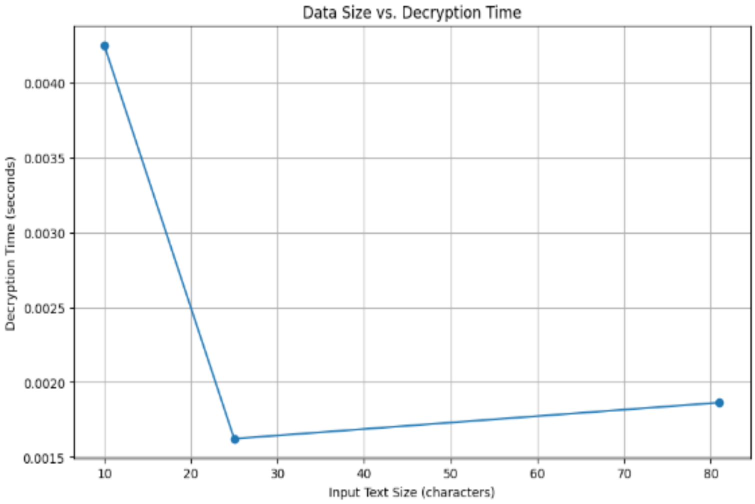

Figure 3. Data size vs decryption time

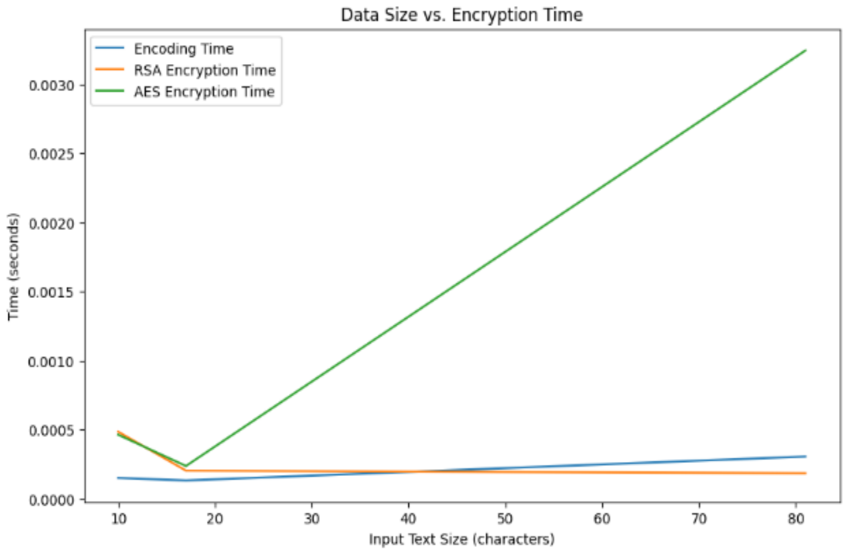

Figure 4. Encryption and decryption time analysis

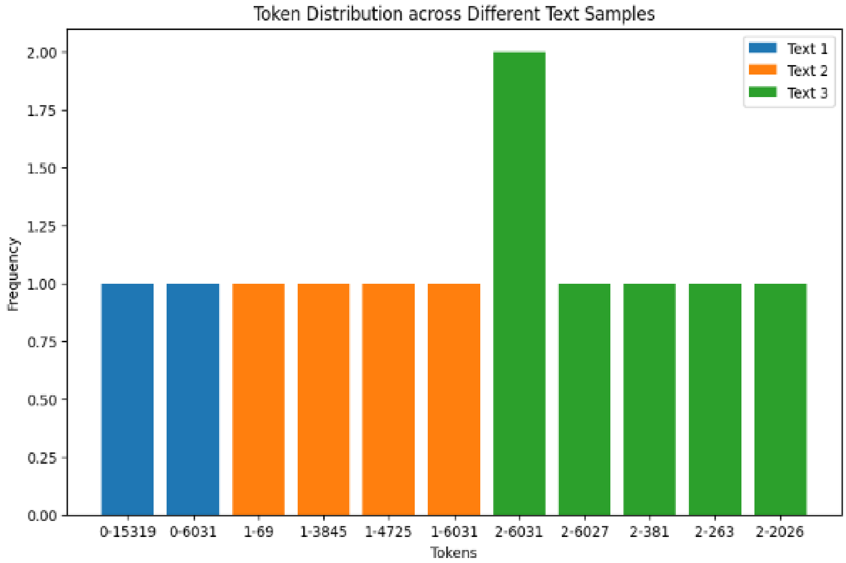

Figure 5. Token distribution across different text samples

5. CONCLUSION

The integration of deep learning with encryption algorithms signifies a groundbreaking advancement in digital security, offering a promising pathway to fortify cybersecurity measures against the complexity of contemporary cyber threats. This research elucidates the potent synergy between the predictive capabilities of deep learning and the robust security offered by advanced encryption techniques. Through our methodological approach, we demonstrate a dual-layer encryption system that not only enhances data security but also maintains efficiency in data transfer and computational resource allocation. This model presents a significant leap forward in securing sensitive textual data across vulnerable digital channels. Our findings indicate a substantial enhancement in cybersecurity, suggesting a pivotal shift towards more dynamic and resilient security architectures. This research contributes to both the academic sphere and practical field, laying the groundwork for future innovations in cybersecurity strategies. It underscores the critical role of emerging technologies in crafting a secure digital ecosystem, capable of withstanding the evolving landscape of cyber threats. In light of this study's implications, future research should explore the integration of additional machine learning techniques to further refine encryption processes, the adaptation of this model to various data types beyond textual information, and the evaluation of its effectiveness in real-world cybersecurity applications. The potential for expanding this model to incorporate quantum-resistant algorithms also presents a valuable avenue for research, ensuring its relevance and efficacy in the face of advancing computational capabilities.

REFERENCES

[1] B. Kaushik, V. Malik, and V. Saroha, "A review paper on data encryption and decryption," *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 4, pp. 1986–1992, Apr. 2023, doi: 10.22214/ijraset.2023.50101.

[2] E. Milanov, "The RSA algorithm," University of Washington, 2009. Accessed: Feb 8, 2024. [Online], Available: https://sites.math.washington.edu/~morrow/336_09/papers/Yevgeny.pdf

[3] A. Muhammad Abdullah, "Advanced encryption standard (AES) algorithm to encrypt and decrypt data," *Cryptography and Network Security*, Jun. 2017, [Online]. Available: https://www.researchgate.net/publication/317615794.

[4] S. Roy, A. R. Shovon, and M. Whaiduzzaman, "Combined approach of tokenization and mining to secure and optimize big data in cloud storage," in *5th IEEE Region 10 Humanitarian Technology Conference 2017, R10-HTC 2017*, Dec. 2018, vol. 2018-January, pp. 83–88, doi: 10.1109/R10-HTC.2017.8288912.

[5] Z. Vahdati, S. M. D. Yasin, A. Ghasempour, and M. Salehi, "Comparison of ECC and RSA algorithms in IoT devices," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 16, pp. 4293–4308, Aug. 2019.

[6] F. O. Mojisola, S. Misra, C. Falayi Febisola, O. Abayomi-Alli, and G. Sengul, "An improved random bit-stuffing technique with a modified RSA algorithm for resisting attacks in information security (RBMRSA)," *Egyptian Informatics Journal*, vol. 23, no. 2, pp. 291–301, Jul. 2022, doi: 10.1016/j.eij.2022.02.001.

[7] 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 Engineering Journal*, vol. 84, pp. 275–284, Dec. 2023, doi: 10.1016/j.aej.2023.10.054.

[8] M. R. Alfani, M. Furqan, and Y. R. Nasution, "Securing text data using digital signature algorithm (DSA) and advanced encryption standard (AES)," *Journal of Science and Social Research*, vol. 7, no. 1, pp. 301–306, 2024, doi: 10.54314/jssr.v7i1.1686.

[9] M. M. Alani, "Applications of machine learning in cryptography: A survey," in *ACM International Conference Proceeding Series*, Jan. 2019, pp. 23–27, doi: 10.1145/3309074.3309092.

[10] B. Kumar Pandey *et al.*, "Encryption and steganography-based text extraction in IoT using the EWCTS optimizer," *Imaging Science Journal*, vol. 69, no. 1–4, pp. 38–56, May 2021, doi: 10.1080/13682199.2022.2146885.

[11] P. Kuppuswamy, S. Q. Y. A. K. Al-Maliki, R. John, M. Haseebuddin, and A. A. S. Meeran, "A hybrid encryption system for communication and financial transactions using RSA and a novel symmetric key algorithm," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 12, no. 2, pp. 1148–1158, Apr. 2023, doi: 10.11591/eei.v12i2.4967.

[12] R. Sood and H. Kaur, "A literature review on RSA, DES and AES encryption algorithms," in *Emerging Trends in Engineering and Management*, Soft Computing Research Society, 2023, pp. 57–63.

[13] R. Akter, M. A. R. Khan, F. Rahman, S. J. Soheli, and N. J. Suha, "RSA and AES based hybrid encryption technique for enhancing data security in cloud computing," *International Journal of Computational and Applied Mathematics & Computer Science*, vol. 3, pp. 60–71, Oct. 2023, doi: 10.37394/232028.2023.3.8.

[14] N. Sangwan, "Text encryption with Huffman compression," *International Journal of Computer Applications*, vol. 54, no. 6, pp. 29–32, 2012.

[15] P. Wang and X. Li, "TEDL: A text encryption method based on deep learning," *Applied Sciences (Switzerland)*, vol. 11, no. 4, pp. 1–30, Feb. 2021, doi: 10.3390/app11041781.

[16] S. Jagadeesh, S. M. Ali, S. P. G. Selvan, M. Aljanabi, M. Gopianand, and J. P. J. Hephzipah, "Hybrid AES-modified ECC algorithm for improved data security over cloud storage," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 32, no. 1, pp. 46–56, Aug. 2023, doi: 10.37934/ARASET.32.1.4656.

[17] S. A. Chakkaravarthy, "A novel hybrid algorithm for an efficient data security in hospital," M.G.R. Educational and Research Institute University, 2023.

[18] S. Samanth, Prema K V, and M. Balachandra, "CLEA-256-based text and image encryption algorithm for security in IOD networks," *Cogent Engineering*, vol. 10, no. 1, Jul. 2023, doi: 10.1080/23311916.2023.2234123.

[19] U. Singh and U. Garg, "An ASCII value based text data encryption System," *International Journal of Scientific and Research Publications*, vol. 3, no. 11, 2013.

[20] Kalvikkarasi and Saraswathi A, "An empirical study of hybrid cryptographic algorithms," *International Journal of Information Technology, Research and Applications*, vol. 2, no. 1, pp. 22–32, Mar. 2023, doi: 10.59461/ijitra.v2i1.50.

[21] A. Srikumar and S. D. Pande, "Comparative analysis of various evolutionary algorithms: past three decades," *ICST Transactions on Scalable Information Systems*, Nov. 2023, doi: 10.4108/eetsis.4356.

[22] V. P. Sriram *et al.*, "Enhancing cybersecurity through blockchain technology," in *Handbook of Research on Cybersecurity Issues and Challenges for Business and FinTech Applications*, IGI Global, 2022, pp. 208–224.

[23] Z. A. Mohammed, H. Q. Gheni, Z. J. Hussein, and A. K. M. Al-Qurabat, "Advancing cloud image security via AES algorithm enhancement techniques," *Engineering, Technology and Applied Science Research*, vol. 14, no. 1, pp. 12694–12701, Feb. 2024, doi: 10.48084/etasr.6601.

[24] S. M. Radhi and R. Ogla, "In-depth assessment of cryptographic algorithms namely DES, 3DES, AES, RSA, and blowfish," *Iraqi Journal of Computer, Communication, Control and System Engineering*, pp. 125–138, Sep. 2023, doi: 10.33103/uot.ijccce.23.3.11.

[25] S. D. Pande, G. Singh, D. M. S. Zekrifa, S. P. Kodgire, S. A. Patel, and V. T. Le, "Threshold public key-sharing technique in block chain," in *Artificial Intelligence, Blockchain, Computing and Security - Proceedings of the International Conference on Artificial Intelligence, Blockchain, Computing and Security, ICABCS 2023*, vol. 1, CRC Press, 2024, pp. 630–639.

BIOGRAPHIES OF AUTHORS

Renuka Shone Durge received a B.E degree in information technology from Pune University, India in 2008, M.E degree in computer science and engineering from Sant Gadge Baba University, Amaravati. Maharashtra, India in 2015. She is currently a lecturer at the Government College of Engineering Amravati. India. Areas of specialization of the author is in the domain cloud computing and doing research work on cloud security system. She can be contacted at email: renuka434@gmail.com.
.

Dr. Vaishali M. Deshmukh holds a PhD degree in computer science and engineering and done B.E, M.E from Sant Gadge Baba University, Amaravati, Maharashtra India. She is currently an associate professor in the Department of Computer Science and Engineering Prof. Ram Meghe Institute of Technology and Research, Badnera-Amravati, India. Areas of specialization is in the domain of algorithmics, system software and have published more than 64 international papers. She can be contacted at email: vmdeshmukh@mitra.ac.in.