Efficient Data Chucking Approach To Optimizing Information Leakage In Cloud Computing

Geetinder saini[1], **Dr. Navdeep kaur**[2]
[1]Research Scholar, Department of CSE, Sri Guru Granth Sahib World University, Fatehgarh Sahib
[2]Dean Research, Sri Guru Granth Sahib World University, Fatehgarh Sahib
Corresponding author: First A. Author (e-mail: contact2geetinder@gmail.com).

Abstract: *This paper presents the Rapid Asymmetric Maximum Algorithm, an optimized version of the AE (Asymmetric Extremum) algorithm, designed to reduce computational load and enhance resistance to byte-shifting in data chunking. In the Rapid Asymmetric Maximum Algorithm (AE), for example, two types of windows—fixed and variable-sized—are used, but they are arranged differently. The byte containing the maximum value is placed at the start of a chunk, followed by a variable-sized window and a fixed-sized window. After determining which byte in the fixed-sized window has the highest value, the algorithm looks to see if the next byte has a greater value. If a higher value is found, it becomes the new maximum, determining the cut-point. The Rapid Asymmetric Maximum Algorithm increases processing speed by scanning only bytes that are equal to or more than the current maximum value, in contrast to AE. Since bytes are more likely to be smaller than the maximum, fewer checks are needed, reducing overhead. The algorithm's sliding window method uses a hash to identify the pattern and begins at the beginning of the chunk and moves leftward until it is identified. This method shares similarities with Rabin-based chunking, as it uses fixed windows and interspersed bytes to determine the cut-point.*

Keywords: *Cloud Computing, information leakage, Chunking, Data Duplication, Data Storage, Asymmetric Extremum*

I. INTRODUCTION

The introduction of cloud computing has altered how people and companies store, process, and manage data. Cloud computing enables flexible, scalable, and economical IT solutions through products like Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Yet, given the intensifying usage of cloud environments, concerns about data security have grown to be more prominent and have manifested in a clear way [1][2]. Among the most critical issues endangering cloud security is information leakage, the improper release of secrets to unauthorized parties. In cloud computing, information leakage may occur due to diverse factors, such as misconfigured cloud storage, insecure APIs, weak authentication mechanisms, vulnerabilities in multi-tenant environments, and insider threats. The vulnerabilities are exploited by cybercriminals to steal the personal and sensitive information of individuals, which can be extremely harmful. Not only can victims suffer financial losses, but their public image can also be damaged, and they may be fined by legal authorities [3][4]. At the same time, the shared nature of cloud infrastructure worsens the situation because an attack can affect not just one organization, but the entire infrastructure. Therefore, both service providers and clients must take securing the infrastructure and the data they use very seriously. The issue of information leakage has been worsened by the complicated nature of cloud services, which, when coupled with the lack of visibility over the location and processing of data, has added further challenges. In general, a cloud environment contains several layers of abstraction, and with many of them being shared, managing the security of data becomes difficult for organizations [5][6]. Data leakage can occur at several points during the transmission of data over the network, while data is stored in the cloud, and even through sophisticated attacks that exploit weaknesses in hardware, such as CPU caches or memory. For the most part, when poorly encrypted information is used for instance, due to outdated algorithms or oversights in the encryption of sensitive data, it becomes much easier for cybercriminals to monitor and misuse the information. Similarly, unauthorized access and mismanagement of user accounts provide additional pathways for attackers [7][8]. The risk of an attack increases when a third party introduces insecure services. In such cases, sensitive information can be exposed without immediate detection. In cloud environments, encryption technology remains one of the most effective tools for securing data against leaks. It works by making data indecipherable to anyone who does not possess the correct decryption keys, even in the event of unauthorized access. In the context of cloud computing, the approaches used in data security can be divided in several categories according to the purposes for which they are intended. A set of methods is targeted at the prevention of data leakage through the adoption of encryption algorithms that operate on plaintext, thus covertly converting the information into ciphertext, while another set of measures is employed for the identification and elimination of such incidents if they occur [9][10]. The major techniques, among others, are encryption, access control, differential privacy, watermarking, and probabilistic models, which respectively cover various data security aspects. Cryptography is the method mostly used in the cloud to prevent the leakage of data in direct and indirect forms. It encodes the original data into garbled data using already approved algorithms in such a way that even if, during an attack, the attackers get the access to storage and backup data, they fail to get the information without the decryption key. To secure data in cloud environments, a variety of cryptographic techniques must be used. The most well-known of these algorithms is the Advanced Encryption Standard (AES), a symmetric encryption technique that allows key sizes of 128, 192, or 256 bits and is frequently applied to fixed block sizes (usually 128 bits). AES continues to be highly effective for encrypting massive data even when it is at rest because of its robust security and quick processing speed. A public key is used for encryption, and a private key is used for decryption, in the Rivest-Shamir-Adleman (RSA) algorithm, which falls under the category of asymmetric encryption. RSA is quite effective in the data security department particularly for secure transmission of data, secure key sharing, and the protection of confidential communications between clients and cloud servers [11][12]. The most popular technique among them is Elliptic Curve Cryptography (ECC) which is equivalent to RSA in terms of security but with much smaller key sizes. This makes ECC the best option for cloud applications that have performance, storage, and bandwidth efficiency as the driving factor. ECC is extremely beneficial for instances such as mobiles and IoT devices connecting to the cloud, where computational resources are limited. Furthermore, hashing techniques like SHA-256 (Secure Hash Algorithm 256-bit) are frequently employed to protect the integrity of data. Rather than encrypting data for decryption later, hashing transforms data into a fixed-size digest, ensuring that any alteration to the original input can be easily detected. Hashing is fundamental to securing authentication mechanisms, digital signatures, and blockchain technologies used within many modern cloud services. Together, these cryptographic methods form a robust framework that protects cloud data from unauthorized access, tampering, and leakage [13][14]. Access control mechanisms are designed to ensure that sensitive data can be viewed or altered only by those people, programs, or services that are authorized to do so. Access control in the cloud is accomplished by Identity and Access Management (IAM) systems, which give administrators the ability to establish multi-factor authentication (MFA) and specify comprehensive policies (role-, attribute-, or policy-based access control). Differential privacy is a quantitative approach designed to determine whether someone's data was part of the training set or not, without actually revealing the identification of the individual user and instead relying on information from the entire available group. The noise is mathematically controlled and carefully added to the query results or the output of the machine learning model, preventing an adversary from learning about the presence of any individual's information in the original data. When combined with cloud machine learning services, such as federated learning with differential privacy (DP) guarantees, it allows organizations to extract trends that can be used for framing problems, making decisions, and providing recommendations from large datasets without exposing the details of any single user [15][16]. Watermarking is a process of embedding a slight trace into files so that they remain visually unaltered, yet their detection and authentication are possible in cases of unauthorized disclosure or spread. For example, text, images, or structured datasets could be watermarked, allowing the breach source such as a certain user or tenant, to be located if necessary. In cloud environments, watermarking can range from applying at the database row or file level to serving as a means of enabling verification and traceability. While the visual quality of digital pictures is not significantly affected, it could be noticeably altered when prints are made. Even with the threat of both metadata removal and format changes, the inserted signal through robust watermarking can withstand the common alterations typically encountered. Probabilistic data leakage detection models employ statistical and machine learning methods to establish "usual" behaviour, as well as to recognize irregularities that might result from exfiltration. Examples of these include Bayesian networks, hidden Markov models, and clustering algorithms that monitor various metrics such as the frequency of requests, the amount of data transferred, the time of access, and the complexity of searches. When system usage deviates significantly from the predicted behaviour, an alert is generated and the problematic session can potentially be cut off in real time, thus enabling the detection of unguarded data exits.

II. LITERATURE REVIEW

The Y. Zhong et al. (2025) introduced a novel network security technique that combines mutual information neural networks and Gaussian denoising to protect network data while maintaining high data utility [17]. The technique created a privacy protection mechanism using Gaussian noise and K-dimensional perturbation trees, which was enhanced by an intrusion detection method based on Bayesian networks. To assess and optimize the protection scheme's parameters, mutual information was used. The experimental findings demonstrated a data utility retention rate of 85% and a maximum of three privacy violations. Over time, through continuous optimization, breaches were reduced to zero. The technique worked well to preserve data utility during transmission and storage while enhancing data security and privacy in cloud computing settings.

Jose et al. (2024) presented a new technique called CybS-CC-SACGAN COA to enhance cloud computing security by fusing the Crayfish Optimization Algorithm (COA) with SACGAN (Self-Attention Conditional Generative Adversarial Networks) [18]. Data termination and missing values were addressed by pre-processing data from the NSL-KDD database using a Reformed Phase Conserving Vibrant Range Compression filter. Utilizing the MRFOA (Manta Ray Foraging Optimization Algorithm), feature selection was optimized. SACGAN categorized cloud data into numerous attack kinds, such as R2L and DoS, as well as normal. SACGAN's performance was enhanced by the COA, increasing the accuracy of anomaly detection. Comparing the Python implementation to other methods, it demonstrated greater AUC, faster computation, lower error rates, and higher detection accuracy.

H. Liu, et al. (2024) explored data protection problems in cloud computing, especially the role of symmetric and asymmetric encryption in improving network security, with a focus on public security systems [19]. This study provided both theoretical insights and practical recommendations for strengthening network security. Experimental results showed that RSA encryption for a 20MB file took around 6400 milliseconds for encryption and 6200 milliseconds for decryption. These findings highlighted that while asymmetric encryption improved security, it impacted performance with large data, necessitating a balance between security and efficiency.

M. Saleem et al. (2023) proposed a novel zero-trust security approach that integrates machine learning for multimedia data analytics to enhance insight into service operations and hazards for confirming confidence in SaaSm [20]. The model leveraged Federated Learning to extract features from data offered by many cloud users, processed using AI. Rich models were used for large-scale multimedia data extraction to monitor cloud service behavior. The analysis of data features was done using an Ensemble Classifier, and decisions were made by majority vote. The model's efficacy in tracking cloud service behaviors, confirming SaaS authenticity, and identifying trust violations was demonstrated through experiments on a standard dataset.

S. Mohammed et al. (2023) suggested C-DSS (Cloud-based Data Security System) with a five-tiered trust model for cloud-edge data-sharing frameworks [21] Before sharing CTI (Cyber Threat Information) for analysis, it gave data owners the option to choose a trust level and sanitization technique. Depending on the degree of trust, either the cloud service provider or an end device could carry out the sanitization procedure. In order to satisfy various requirements for transmitting secret CTI, the study covered the trust architecture, cloud setup, and installation techniques in depth. In comparison to popular cloud encryption systems, test results demonstrated enhanced data protection, quicker cipher processing, and superior security services. An overview of pilot applications that verified the design was presented at the end of the study.

H. M. Alshahrani, et al. (2022) proposed new method, Chaotic Chimp Optimization with Machine Learning-enabled Information Security (CCOML-IS) to improve cloud network security by detecting anomalies and intrusions [22]. The method began by normalizing network data using data transformation and min-max scaling. It then applied a CCOA (Chaotic Chimp Optimization Algorithm) for optimal feature selection, which enhanced detection accuracy. KRR (Kernel Ridge Regression) was used as the classifier to determine security threats. The integration of CCOA helped refine features and boosted overall classification performance. Extensive experiments on benchmark datasets showed that CCOML-IS had outperformed existing techniques in many performance metrics, proving its effectiveness in cloud-based security monitoring.

F. Thabit et al. (2022) introduced a homomorphic cryptographic method that is both efficient and lightweight with a dual-layer encryption architecture [23]. To increase data security in cloud environments, the first layer employed a novel lightweight encryption technique, and the second layer used multiplicative homomorphic encryption. Both symmetric and asymmetric cryptography's properties were merged in this hybrid paradigm. Numerous criteria were used to evaluate the algorithm's performance, including entropy variation, memory usage, computation time, key sensitivity, statistical analysis, and histogram distribution. Experimental results showed strong security, along with significant improvements in encryption speed, resource utilization and overall throughput, outperforming conventional cryptographic techniques previously used in cloud computing.

F. Thabit et al. (2021) created a novel cryptographic strategy to increase cloud computing security by using a dual-layer encryption methodology [24]. By splitting the plaintext and key into equal portions, the first layer applied Shannon's concepts of diffusion and confusion through the use of logical operations like XOR, XNOR, and bit shifting. In order to improve the complexity of the encryption, the second layer was inspired by molecular biology, simulating genetic processes by converting binary data into DNA sequences, transcribed into mRNA, and finally translated into protein structures. Experimental evaluations showed improved data protection, showcasing the algorithm’s better performance in terms of cipher strength, processing speed, and encryption efficiency compared to existing cloud security methods.

F. Thabit et al. (2021) introduced a novel lightweight cryptographic technique to improve data security for cloud-based applications [25]. Feistel structures and substitution-permutation networks served as inspiration for the invention of this 128-bit block cipher, which required a 128-bit key in order to boost encryption complexity. It fulfilled Shannon's diffusion and confusion principles by using logical operations such as bit shifting, swapping, XOR, and XNOR. The algorithm also supported customizable key lengths and encryption rounds, providing flexibility. Experimental evaluations showed robust security performance, with improved encryption speed and improved protection compared to conventional cryptographic methods that were commonly used in cloud computing.

III. RESEARCH METHODOLOGY

This study introduces the Rapid Asymmetric Maximum Algorithm, a faster variant of the Asymmetric Extremum designed to achieve low computing overhead and resilience to byte shifts. This approach uses both fixed and variable-sized windows, which is comparable to AE. These windows' placement, however, deviates from the AE method. The byte with the greatest value and the variable-sized window are positioned after the fixed-sized window, which is at the start of the chunk in this approach. According to this method, the byte with the highest value is placed at the end of the chunk, as seen in Figure 1.

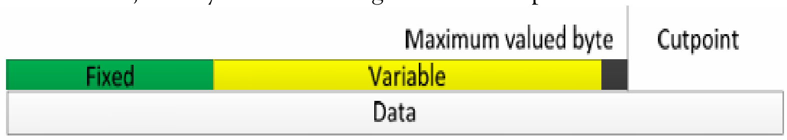
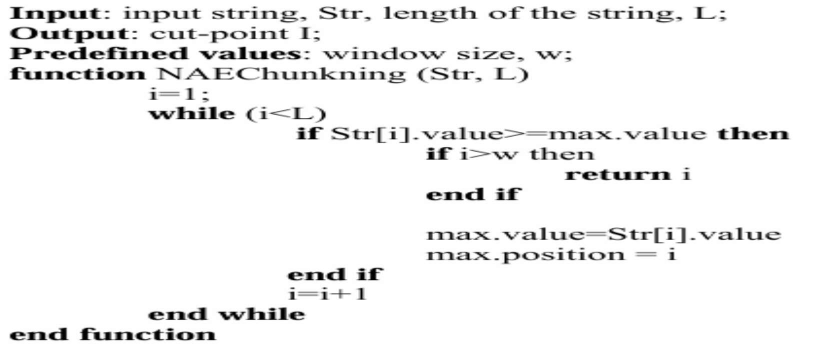


Figure 1. Rapid Asymmetric Maximum algorithm’s windows configuration

The algorithm searches the window of a fixed size for the byte with the highest value. The cut-point is established and the byte next to the fixed-sized window is chosen as the maximum-valued byte if its value is greater than the value of the window. The pseudo code in Figure 2 illustrates how the algorithm proceeds to the next byte if it doesn't find a larger byte. The algorithm's minimum chunk size is therefore 1, which is the same as the fixed-sized window's size.

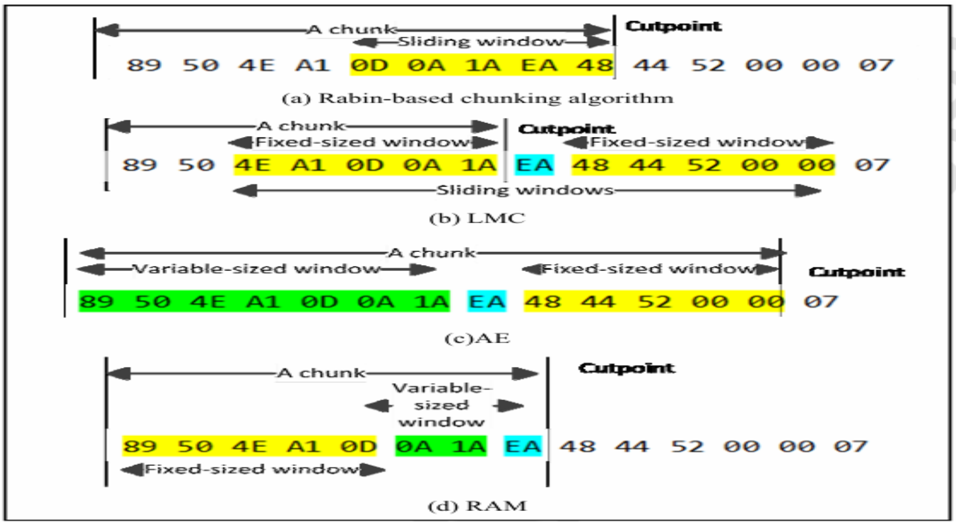


```
Input: input string, Str, length of the string, L;
Output: cut-point I;
Predefined values: window size, w;
function NAEChunkning (Str, L)
        i=1;
        while (i<L)
                if Str[i].value>=max.value then
                        if i>w then
                                return i
                        end if

                        max.value=Str[i].value
                        max.position = i
                end if
                i=i+1
        end while
end function
```

Figure 2. Pseudo Code of algorithm procedure

The chunking strategy employed in the method that is being discussed is demonstrated by the algorithm in Figure 2. AE examines all bytes that are smaller than or equal to the maximum-valued byte, whereas the proposed approach only searches for bytes that are equal to or more than the current maximum value, saving processing time. As the probability of the next byte being smaller than the current maximum value is greater than the possibility of it being larger, the suggested method is less likely than AE to encounter the first condition. As a result, the suggested method's overhead is decreased.



(a) Rabin-based chunking algorithm

(b) LMC

(c)AE

(d) RAM

Figure 3. An illustration of the algorithms' operation using a 14-byte string

Figure 3 presents examples of many existing chunking techniques using a window size of 5 bytes. The examples show how each algorithm functions with the same byte string. The Rabin-based chunking algorithm employs a sliding window that starts at the beginning of the chunk, as depicted in Figure 3(a). Initially, the sliding window contains 0x89594EA10D. Since the window's hash doesn't match the expected pattern, it advances to the next byte by removing the most significant byte, shifting the window left, and adding the new byte. The hash is recalculated as the window slides further until the hash value matches the target pattern, in this case 0x0D0A1AEA48. The sliding window is included in this instance, but it can also be removed from it. LMC is similar to the Rabin-based chunking technique since it uses sliding windows, as Figure 3(b) illustrates. A byte separates each of the two fixed-size windows used by LMC. The cut-point in the example is reached when the byte between the two fixed-size windows is more than the sum of the bytes of the two windows. In Figure 3(c), the AE approach is shown. In contrast to LMC, AE determines the cut-point by examining every byte. The fixed-size window always shows to the right of the scanned byte, and AE compares each byte's value to all the other bytes in the window. A cut-point is identified when a byte's value exceeds the values in the fixed-size window. When the algorithm reached 0xEA, which was greater than any byte in the fixed-size window, it identified the cut-point. In this case, the method scanned from 0x89 to 0xEA. The fixed-sized window's right side is where the cut-point is located. The suggested technique works similarly to AE in terms of byte scanning, going through each byte and comparing its value to the window's maximum value. Finding the greatest value in the fixed-size window—in this example, 0xA1—is the first step in the method. Following the fixed-size window scan, it compares every byte to this upper limit. A cut-point is set when a byte goes over the maximum value. Since 0xEA is greater than 0xA1, it becomes the cut-point, as seen in Figure 3(d). The final scanned byte is included in the chunk.

IV. RESULT AND DISCUSSION

The mathematics toolbox is a set of MATLAB-based mathematical tools that help carry out this research effort. In addition to enabling users to generate data visualizations, this software does the required numerical calculations. It is becoming a vital tool in many scientific and engineering domains due to its programming capabilities. Topographic data, gigapixel resolution, embedded ICC profiles, high dynamic range, and support for several picture formats are all included in the picture Processing Toolbox. Three datasets are employed in this study: text, images, and video. The total number of chunks, chunking time, average chunk size, throughput, deduplication ratio, and processing time are some of the variables used to assess the suggested algorithm's performance.

Table 1. Simulation Parameters

Number of cloudlets	10
Number of virtual machines	7
Operating system	Linux
Architecture	64 bit
Dataset	Home, Media, Linux

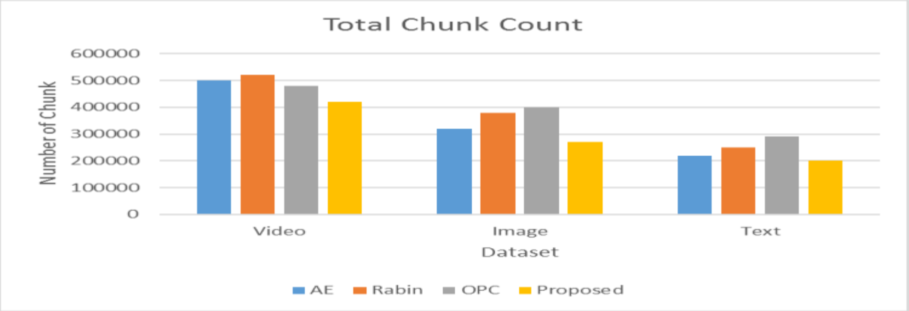


Figure 4. Total chuck Count

Figure 4 shows how the performance of the proposed algorithm is assessed based on the number of chunks. The proposed algorithm outperforms AE, Rabin, and OPC in terms of performance.

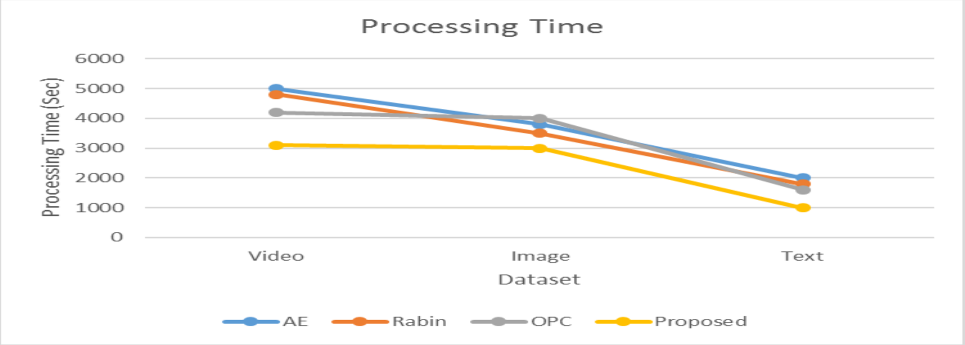


Figure 5. Total chuck Count

As depicted in Figure 5, processing time is used to gauge how well the suggested method performs. It is found that, in comparison to AE, Rabin, and OPC, the suggested method produces the longest processing time.

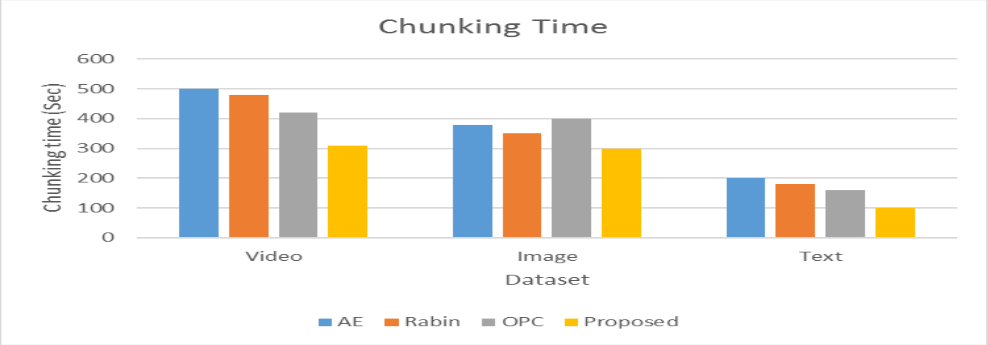


Figure 6. Chunking Time

As presented in Figure 6, the chunking time of the proposed algorithm is compared with that of other algorithms including AE, Rabin and OPC. It is observed that the proposed algorithm achieves the shortest chunking time in comparison to the other algorithms.

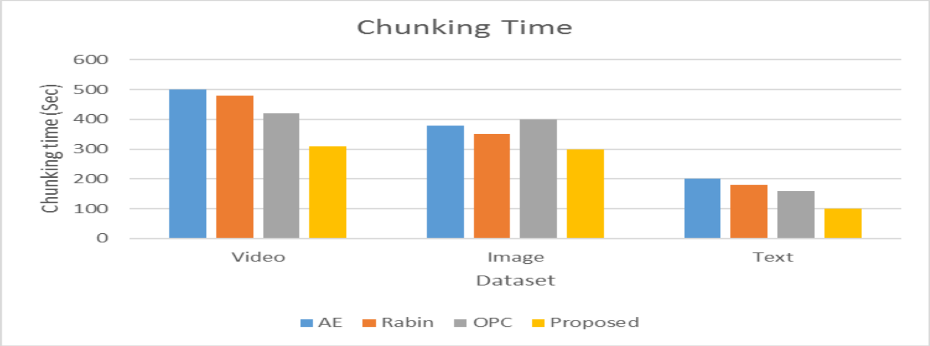


Figure 7. Average Chunk Size

Figure 7. illustrates how the performance of the proposed algorithm is examined using the average chunk size. Analysis shows that the proposed algorithm outperforms OPC, Rabin, and AE.

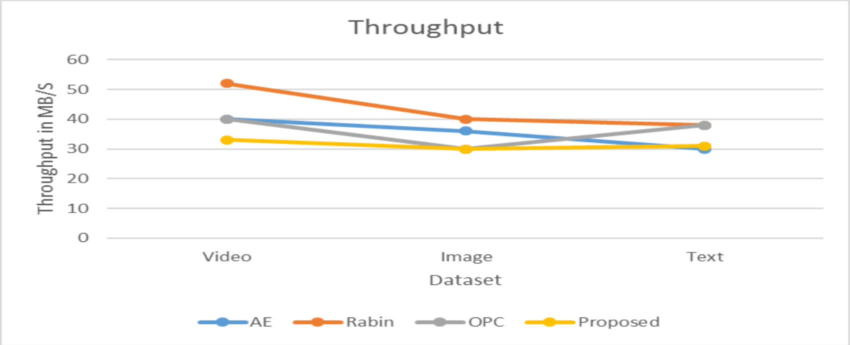


Figure 8. Throughput Analysis

Figure 8 compares the throughput of the proposed method with that of AE, Rabin, and OPC on different kinds of datasets. For all data forms, including text, graphics, and video, the recommended method consistently maintains its stability and attains the highest throughput.

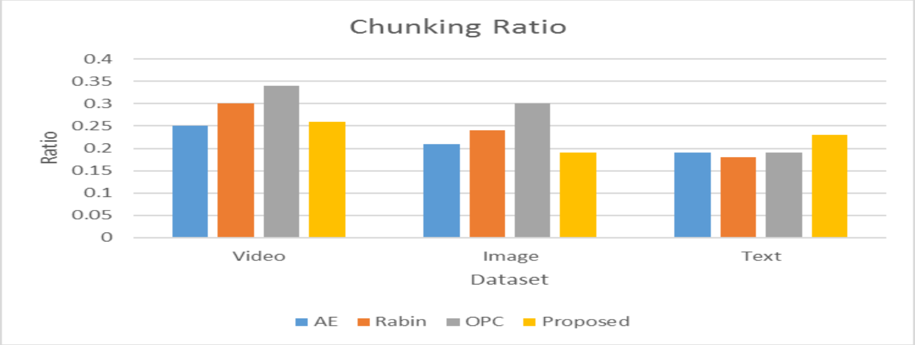


Figure 9. Chunking Ratio

Figure 9 shows how the chunking ratio of the proposed algorithm compares to other algorithms, such as OPC, Rabin, and AE. It is discovered that the recommended strategy achieves the lowest chunking ratio when compared to the other methods.

V. CONCLUSION

This paper introduces the Rapid Asymmetric Maximum Algorithm, a more efficient alternative to the AE (Asymmetric Extremum) algorithm, designed to reduce computational costs and improve resistance to byte-shifting in data chunking. The Rapid Asymmetric Maximum Algorithm, like AE, employs both fixed and variable-sized windows, but they are structured differently: a byte with the maximum value is placed at the beginning of the chunk, followed by a variable-sized window and a fixed-size window. The sliding window in the Rapid Asymmetric Maximum Algorithm starts at the beginning of the chunk and moves leftward, comparing hashed values until it finds a matching pattern. This method, similar to Rabin-based chunking, uses fixed windows interspersed with bytes to more efficiently identify cut-points. Ultimately, the Rapid Asymmetric Maximum Algorithm optimizes data chunking by focusing solely on relevant bytes, striking a balance between speed and accuracy while reducing computational overhead. This structured method represents a significant improvement over traditional AE chunking, offering efficiency gains that can benefit data-heavy applications

REFERENCES

[1] M. Dhinakaran, M. Sundhari, S. Ambika, V. Balaji and R. T. Rajasekaran, "Advanced Machine Learning Techniques for Enhancing Data Security in Cloud Computing Systems," 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India, 2024, pp. 1598-1602, Doi: 10.1109/IC2PCT60090.2024.10486559.

[2] Z. Zou, "Research on User Information Security based on Cloud Computing," 2023 IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2023, pp. 35-39, Doi: 10.1109/ITOEC57671.2023.10291704.

[3] C. Zhao, L. Zhao, C. Zhao and X. Sun, "Data Security Framework and Privacy Protection Strategies in Cloud Computing Environment," 2023 5th International Conference on Frontiers Technology of Information and Computer (ICFTIC), Qingdao, China, 2023, pp. 51-54, Doi: 10.1109/ICFTIC59930.2023.10455863.

[4] Q. Wei, "Analysis of the role of computer big data and cloud computing in information security," 2023 International Conference on Networking, Informatics and Computing (ICNETIC), Palermo, Italy, 2023, pp. 119-123, Doi: 10.1109/ICNETIC59568.2023.00031.

[5] J. Peng, "Research on E-government Information Security Based on Cloud Computing," 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 2022, pp. 312-316, Doi: 10.1109/ITAIC54216.2022.9836548.

[6] W. Ye, "Application of Cloud Computing Technology in Student Information System Security Processing," 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 2022, pp. 264-267, Doi: 10.1109/ITAIC54216.2022.9836769.

[7] T. Chen and H. Liu, "Discussion on Network Information Security Based on Cloud Computing Environment," 2022 8th Annual International Conference on Network and Information Systems for Computers (ICNISC), Hangzhou, China, 2022, pp. 65-68, Doi: 10.1109/ICNISC57059.2022.00023.

[8] H. Yao, "Data Storage Security System based on Cloud Computing," 2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China, 2022, pp. 1220-1223, Doi: 10.1109/ICETCI55101.2022.9832390.

[9] H. Liang, H. Liu, F. Dang, L. Yan and D. Li, "Information System Security Protection Based on SDN Technology in Cloud Computing Environment," 2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China, 2021, pp. 432-435, Doi: 10.1109/AEECA52519.2021.9574276.

[10] M. Huang, "Design of basic process of information security risk assessment in cloud computing environment," 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE), Guangzhou, China, 2021, pp. 494-499, Doi: 10.1109/ICCECE51280.2021.9342156.

[11] D. Li, L. Yan, Y. Song, S. Li and H. Liang, "Network computer security and protection measures based on information security risk in cloud computing environment," 2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China, 2021, pp. 424-427, doi: 10.1109/AEECA52519.2021.9574194.

[12] Y. Ma, H. -j. Ni and Y. Li, "Information Security Practice of Intelligent Knowledge Ecological Communities with Cloud Computing," 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE), Guangzhou, China, 2021, pp. 242-245, Doi: 10.1109/ICCECE51280.2021.9342141.

[13] B. Shi, "Relative Analysis of Network Information Security Technology against the Background of "Cloud Computing"," 2021 7th Annual International Conference on Network and Information Systems for Computers (ICNISC), Guiyang, China, 2021, pp. 40-43, Doi: 10.1109/ICNISC54316.2021.00015.

[14] N. Tutubala and T. E. Mathonsi, "A Hybrid Framework to Improve Data Security in Cloud Computing," 2021 Big Data, Knowledge and Control Systems Engineering (BdKCSE), Sofia, Bulgaria, 2021, pp. 1-5, Doi: 10.1109/BdKCSE53180.2021.9627294.

[15] S. R. Botirov and D. R. Kh, "Analysis of Information Security Evaluation Models in the Cloud Computing Environment," 2020 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2020, pp. 1-5, Doi: 10.1109/ICISCT50599.2020.9351427.

[16] Z. Tang, "A Preliminary Study on Data Security Technology in Big Data Cloud Computing Environment," 2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), Bangkok, Thailand, 2020, pp. 27-30, Doi: 10.1109/ICBASE51474.2020.00013.

[17] Y. Zhong and X. Li, "Network information security protection method based on additive Gaussian noise and mutual information neural network in cloud computing background," Egyptian Informatics Journal, vol. 30, pp. 100673–100673, Apr. 2025, Doi: https://doi.org/10.1016/j.eij.2025.100673.

[18] Jose., G. Sugitha, Ayshwarya Lakshmi. S, and Preethi Bangalore Chaluvaraj, "Self-Attention Conditional Generative Adversarial Network optimised with Crayfish Optimization Algorithm for Improving Cyber Security in Cloud Computing," Computers & security, vol. 140, pp. 103773–103773, May 2024, Doi: https://doi.org/10.1016/j.cose.2024.103773.

[19] H. Liu, "Data Protection in Cloud Computing Environment: New Dimension of Network Information Security," 2024 First International Conference on Software, Systems and Information Technology (SSITCON), Tumkur, India, 2024, pp. 1-5, Doi: 10.1109/SSITCON62437.2024.10796150.

[20] M. Saleem, M. R. Warsi, and S. Islam, "Secure information processing for multimedia forensics using zero-trust security model for large scale data analytics in SaaS cloud computing environment," Journal of Information Security and Applications, vol. 72, p. 103389, Feb. 2023, Doi: https://doi.org/10.1016/j.jisa.2022.103389.
[21] S. Mohammed, S. Nanthini, N. Bala Krishna, I. V. Srinivas, M. Rajagopal, and M. Ashok Kumar, "A new lightweight data security system for data security in the cloud computing," Measurement: Sensors, vol. 29, p. 100856, Oct. 2023, Doi: https://doi.org/10.1016/j.measen.2023.100856.
[22] H. M. Alshahrani et al., "Metaheuristics with Machine Learning Enabled Information Security on cloud Environment," Computers, materials & continua/Computers, materials & continua (Print), vol. 73, no. 1, pp. 1557–1570, Jan. 2022, Doi: https://doi.org/10.32604/cmc.2022.027135.
[23] F. Thabit, O. Can, S. Alhomdy, G. H. Al-Gaphari, and S. Jagtap, "A Novel Effective Lightweight Homomorphic Cryptographic Algorithm for data security in cloud computing," International Journal of Intelligent Networks, vol. 3, pp. 16–30, 2022, Doi: https://doi.org/10.1016/j.ijin.2022.04.001.
[24] F. Thabit, S. Alhomdy, and S. Jagtap, "A new data security algorithm for the cloud computing based on genetics techniques and logical-mathematical functions," International Journal of Intelligent Networks, vol. 2, pp. 18–33, 2021, Doi: https://doi.org/10.1016/j.ijin.2021.03.001..
[25] F. Thabit, A. P. S. Alhomdy, A. H. A. Al-Ahdal, and P. D. S. Jagtap, "A new lightweight cryptographic algorithm for enhancing data security in cloud computing," Global Transitions Proceedings, vol. 2, no. 1, pp. 91–99, Jun. 2021, Doi: https://doi.org/10.1016/j.gltp.2021.01.013.