INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGIES
BULGARIAN ACADEMY OF SCIENCES
CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **26**, No 1
Sofia • 2026 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2026-0003

Hybrid Techniques for Shared File Security in Cloud Environment Using SHA-256

Alaa AlKhazaleh[1], *Feras E. AbuAladas*[2], *Saad M. Ismail*[2], *Mamoun Abu Helou*[3], *Waheeb Abu-ulbeh*[3], *Rizik M. H. Al-Sayyed*[1]

[1]*King Abdullah II School for Information Technology, The University of Jordan*
[2]*College of Information Technology, Amman Arab University, Jordan*
[3]*Faculty of Administrative Sciences and Informatics, Al-Istiqlal University*
E-mails: alaaakh1993.ak@gmail.com F.abualadas@aau.edu.jo S.ismail@aau.edu.jo
mabuhelou@pass.ps w.abuulbeh@pass.ps r.alsayyed@ju.edu.jo (corresponding author)

***Abstract*:** *This research examines file protection in cloud environments using the SHA-256 algorithm. It evaluates secure hash variants in file transfer systems, aiming to improve throughput while meeting security requirements. Data analysis involved gathering and cleaning datasets, measuring encryption times, and applying statistical and graphical methods. Results show SHA-256 as an effective base for encryption, with parallel processing increasing efficiency when real-time speed is essential. The study highlights the value of transformational techniques to boost performance and recommends hybrid systems that combine SHA-256 with other algorithms for stronger security. These outcomes support further work on cryptographic methods to strengthen the safety and reliability of cloud infrastructures.*

***Keywords*:** *Secure Hash Algorithm* (*SHA), SHA-256, Hashing cryptography, File sharing, Cloud-based communication.*

1. Introduction

The growing power of cloud computing has added an extra dimension to file security vulnerabilities and cloud security challenges [1, 2]. Some of the key aspects of security threats include the constant fear of integration of malicious insiders with strong Identity and Access Management (IAM) systems including multi-factor authentication being employed to reduce it; requirements for secure HTTPS and SSl/TLS protocols to shield against content interception during transport sessions [19], as well as data marring, which can be approached through verification devices based on Secure Hash Algorithm-256 (SHA-256) [14]. Also, in cloud computing, a decrease in the level of control over the data does indicate the type of security requirements specifications that must be integrated within the Service Level Agreements [19]. Standards such as GDPR must be complied with, and prevention against new sophisticated cyber-attacks, even more basic intrusion detection systems [9], are fundamental. Data backup and recovery systems, which

are always proactive in nature, are essential for eliminating any chances of data vulnerability to catastrophes [4]. Likewise, data leaks from potential insiders are addressed through close monitoring, training, and restricted access [22]. API security is equally important as it may introduce vulnerability to the host cloud infrastructures [33]. In conclusion, robust data protection strategies, including encryption, power density controls, and physical access management, are recommended to safeguard sensitive files in the cloud without compromising its potential uses.

The protection of shared files stored on the cloud is critical because those documents are likely to be stolen or even destroyed by malicious attacks. To guard against such threats, greater threshold integration is critical, and this could be achieved using different mechanisms and strategies to ensure that data remains secure. One such approach uses the SHA-256 algorithm [7].

This research focuses on enhancing information security by improving the protection of shared files in cloud environments through advanced encryption techniques, specifically using the SHA-256 algorithm for backend processes. Key objectives include increasing data security in the cloud and creating a framework for organizations to strengthen their data protection and comply with regulations. The study also evaluates SHA-256's efficiency compared to other algorithms, addresses encryption overheads through parallel programming, and fosters confidence in cloud technology by promoting robust security measures.

This study investigated several approaches and tools to analyze the security of file sharing in cloud systems implemented with the SHA-256 algorithm. The chosen Integrated Development Environment (IDE) was Visual Studio due to its power and versatility. The .NET Framework 4.8 was also selected due to its reliability and versatility. The malware samples were obtained by means of the Malware Bazaar API, and the collection of data was executed through a console application. The programming language Python was used along with such libraries as Pandas, NumPy, Scikit-learn, and matplotlib.pyplot, Seaborn, scipy.stats, and statsmodels.stats.multicomp. Encryption methods included SHA-256, SHA-512, SHA-1, and MD5. All actions were performed securely within the .NET environment using the Cryptography Library. Data compilation was through Microsoft Excel while more complicated analyses required working papers to be consolidated. Azure DevOps served to develop cloud and simulation in conjunction with pipelines aimed at automating tasks execution, overseeing their performance, as well as tracing them in real-time. These facilities provided accurate and efficient results, and therefore, the safety of the data contained in the clouds was improved.

The research aims to provide an insight into the performance of different encryption algorithms in the context of cloud computing environments by improving security and effectiveness, in this case, through the SHA-256 algorithm. It has been shown that employing parallel technologies in this case solves the problem of performance thus increasing the throughput of encryption without compromising security assurances, making it possible to use them for the protection of cloud data effectively and conveniently [20, 21]. This work opens opportunities

for further exploration and advancement in the field with the view of enhancing the productivity and security of processes in cloud systems.

The rest of the paper is structured as follows: Section 2 introduces the importance of file security in cloud settings, the SHA-256 algorithm, and the performance efficiency of adopting parallel techniques for the hashing algorithms. Section 3 overviews the adopted methodology and introduces the experimental setting including datasets and simulation environment. Reporting the results and discussions are presented in Sections 4 and 5, respectively. Section 6 provides the conclusions and future directions.

2. Backgrounds and related works

2.1. File security in the cloud

The complexity of securing the files hosted in the cloud has been a matter of focus, with numerous approaches that are aimed at enhancing security in the system being proposed. Early work in [11] investigated the problem of how to secure data stored in cloud computing environments. They stressed the necessity to take security measures for access control and the safeguarding of information from breaches. Such research tackled solutions for current and future security vulnerabilities within cloud computing, and their relevance for practical application in cryptography. In [40], the authors state that oversharing online can expose proprietary know-how and open the door to spear-phishing, identity theft, cyberstalking, and undue influence on career decisions. Since digital traces persist and are hard to erase, refrain from posting work tasks or any sensitive professional details.

In [15] they proposed a less complex scheme for the protection of information within the clouds. Their research aims at creating a security framework with minimal consumption of resources, more importantly, human effort. They also characterize their scheme as suitable in climates with low computation ability, as it requires easy set-ups that would not go to the extreme of compromising on performance. The approach is quite helpful in fulfilling the requirement of security mechanisms that do not compromise performance benchmarks. Their study adds to other research by providing ways in which the level of security in cloud resources is improved but the level of security is reasonable enough to be practical.

In [28], they presented a cloud storage and file-sharing security model. Their research was concentrated on improving the level of security and privacy of files and folders which are kept and/or shared in cloud settings. The authors present a model that combines several security techniques, such as encryption of data, policies for access and sharing of information, and secure protocols exclusively designed for the prevention of secure data leaks in the era of cloud computing. As a cloud storage security umbrella, the model resolved certain issues, such as safeguarding information on a network and in databases from “prying eyes”, and safeguarding the entire infrastructure from unauthorized data access. The authors give adequate attention to the integration of security techniques that enable a holistic approach to the issue of secure file storage and sharing in the clouds.

An interesting direction in this regard is the work [35], which addressed issues of security of sensitive information within the cloud by proposing a variety of techniques to address the challenge. They presented a different technique, which though claimed to be revolutionary in its approach, still maintained the application of asymmetric encryption schemes, though with some improvements. The combination of the two asymmetric encryptions for data and bulk encryption of data with symmetric encryption for efficient transfer of data as well as for the distribution of the key provided a drastic reduction in cost, security, and performance compared to the conventional approaches adopted.

Aspects such as security techniques [35], security measures [11], a general all-in-one security solution [28], and simplified crypto techniques [15], highlight the power of employing different crypto and analytic techniques to overcome the numerous challenges posed by the security of data in the cloud. This also addresses vexing security issues where their results would be the starting point in looking for other potential features of cryptographic approaches and the direction for maintenance work in this field [29].

2.2. SHA-256's algorithm

A widely recognized hashing technique, SHA-256 is a member of the SHA-2 family, which was created by the NSA and approved by NIST in 2001. It is essential for securing and validating data, such as during password setup or the use of digital signatures. SHA-256 can deal with data sets of any size and computes them into a 256-bit (32-byte) hash, which simplifies things such as data verification. Some of its characteristics are fixed fragmentation, since all output is a hash of the same size no matter how large the input is all the time; uniqueness, because of the definitive different output that is generated based on the different input making it impossible to counterfeit; irreversibility in the sense that the hash cannot be converted back to get the initial information, and it is very sensitive to changes in that there is a huge difference in the hash even when there is a very tiny difference in the input. Collectively, these characteristics do indeed make the SHA-256 a trusted and strong selection in data security and integrity [12, 13, 14, 26].

This set of characteristics makes the SHA-256 algorithm a great assurance of security, mainly because of its features, which include uniqueness, one-way, and sensitivity to alterations. Practically, this is a strong mechanic that secures information and verifies users in data storage through cloud facilities, increasing the level of protection of customers' personal data from malicious attacks. The SHA-256 algorithm is employed in a range of operations that require security and integrity of information:

- Digital Data Signatures: To embed digital signatures that help validate and protect data, SHA-256 algorithm is applied [30].
- Encryption of Passwords: All passwords used in social media accounts, email accounts as well as digital wallets are hashed using SHA-256 before being added on databases assuring that even in case the database leaks, sensitive information would be kept saved [25].

- File Integrity Check: Once the new file is created, the network can also utilize the SHA-256 algorithm to ensure that the new file is consistent with the old one by measuring the newly created hash against the original [31].
- Security of Electronic Payments: SHA-256 is also used to ensure the security of electronic payments such as money transfers and bitcoins [17].

Several studies have looked at the adoption of the SHA-256 algorithm, including this study, usage, and security evaluation in relation to boosting file safety in the cloud.

Early work presented in [12] focused on the security aspects of the SHA-256 and its algorithms. Their paper discusses both the pros and cons of the algorithm which can inform development and use in various applications. In [13] they performed a performance analysis of a modification of the basic SHA-256 algorithm. Their research was looking at improvements made in the algorithm with regard to its use in the handling of data. In the year 2017, Suhaili Binti and Watanabe presented a high-performance hardware implementation of the SHA-256 hash function based on FPGA. Their work illustrates the practicality of deploying the SHA256 algorithm in hardware to result in higher throughput and efficiency. In [26] they investigated the SHA Algorithm in the spectrum of cryptography. Their research has investigated the efficacy of the algorithm in dealing with various real-life examples.

In [34] they focused on the design and development of a high-efficiency multimem SHA-256 accelerator that supports Society 5.0. The authors highlight how substantial improvement in the efficacy of the SHA-256 has been witnessed due to its hardware acceleration. In [36] they also considered the processes of creating the SHA-256 compression function implemented using pure chaos S-box.

S r i k a n t h et al. [32] developed a CNN-based method for the design of high throughput controlled enhanced SHA-256 RTL to GDSII using Verilog HDL. They also demonstrated improvements in verbosity and efficiency of the SHA-256 algorithm, making it ideal for demanding applications.

The works on hybrid encryption schemes [35], the security protocols advocated by [11], the all-encompassing security architecture [28, and the lightweight sensitive key management techniques elaborated emphasize the benefits of using different methods and perspectives to solve the problems of security in cloud storage facilities [15]. Additionally, works involving the SHA-256 algorithm enhance the application and optimization of this technique within the context of secure cloud systems. Such works help seek means of enhancing security using cryptography, and they equally underscore the need for more such works.

The combination of different cryptographic techniques, as exemplified by the above-mentioned researchers emphasize the need to use different methods of cryptographic and analytic techniques in combination. Such projects are beneficial for the investigation of encryption techniques and highlight the relevance of further development, which is crucial in relation to the issues concerning cloud storage security [27, 29].

2.3. Parallel techniques to improve encryption performance

Wu et al. [38] observed that by adding parallelization to the SHA-256 algorithm the rate and extent at which it could be computed increased dramatically. What this study helped elucidate was the role of hardware parallelization techniques in the cloud in speeding up algorithms.

One of the programming techniques that have been researched and applied for improvement in the performance of encryption in a cloud environment, which is worth mentioning [23]. In particular, they studied algorithms and their performance for cloud computing systems. Their work explained the advantages gained from employing processing techniques to improve the efficiency of encryption algorithms. They proved that breaking up encryption work into smaller tasks and executing them simultaneously would help reduce the time taken to perform that task and thus the overall effectiveness of the security measures is increased.

In [37] they explained the procedure of parallel traversing of a Merkle tree to achieve quantum Leighton-Micali signatures on GPU platforms. Their analysis showed that such techniques can enhance the operational performance of cloud systems, making them more applicable to high-security usage. The study further noted that GPU-oriented parallel computing can overcome the problems that are inherent in post-quantum cryptographic algorithms.

Bezerra et al. [8] applied a parallel image encryption algorithm that aims to enhance image encryption and decryption systems based on kernel maps. The research indicated the potential of applying such encryption techniques to improve the performance of systems, making them suitable for use in cloud-based imaging processing systems. Similar works of image encryption have been conducted in [3, 10].

In [24] they developed a cipher named PRC6 targeted at enhancing the security of data in a cloud environment in a parallelized architecture. Their studies made observations on the characteristics of PRC6, noting that cloud technology has complex security needs, which PRC6 can fulfill, but at a low processing cost. The study called for more of such lightweight encryption algorithms to be developed while making good use of processing capabilities to meet the security requirements of a cloud infrastructure.

The results of these studies emphasize the importance of encryption techniques and parallel processing in enhancing both the security and the performance of cloud computing systems. Using hybrid encryption schemes, applying different techniques, or utilizing the resources of parallel processing can greatly enhance and improve the security mechanisms of clouds. Looking ahead, researchers should consider investigating encryption techniques and parallel methods to mitigate the changing and challenging security issues characteristic of cloud environments [21].

3. Research methodology

This research adopted a mixed-methods approach to investigate the enhancement of file-sharing security in cloud environments using the SHA-256 algorithm. Data collection involved the acquisition of 4451 malware samples from the Malware

Bazaar platform, representing files encrypted using SHA-256, MD5, SHA-1, and SHA-512 algorithms. These samples were processed using the .NET framework and its Cryptography Library for encryption, with Python (Pandas and NumPy) used for subsequent data analysis and visualization. The core of the study focused on evaluating the performance of SHA-256, both sequentially and using parallel processing techniques. Descriptive statistics (mean, standard deviation, percentiles) were calculated for encryption times, and ANOVA and Tukey's HSD tests were applied to compare algorithm performance. Microsoft Azure DevOps facilitated real-world cloud simulations, allowing for the practical evaluation of SHA-256's effectiveness in a cloud environment.

3.1. Research design

The general objective of this study is to strengthen file-sharing security in cyberspace, with the emphasis being placed on high-end encryption, particularly the SHA-256 algorithm. The research investigates why hybrid encryption, more specifically the SHA family, is more optimal at achieving data integrity and confidentiality. With its impressive system throughput and security characteristics, cloud applications are best complemented by SHA-256. A comparative analysis of SHA-256, SHA-512, SHA-1 and MD5 further clarifies its superiority in terms of safety and efficiency as demonstrated in Table 1.

Table 1. Comparison between algorithms

Feature/Algorithm	SHA-256	SHA-512	SHA-1	MD5
Collision resistance	High	High	Moderate	Low
Key length	256 bits	512 bits	160 bits	128 bits
Speed	Moderate	Slower due to longer bit length	Fast	Very fast
Known vulnerabilities	Few known; robust against attacks	Few known; robust against attacks	Vulnerable to collision attacks	Highly vulnerable; compromised integrity
Recommended usage	Highly recommended for sensitive data	Suitable for various applications	Not advised for critical security	Not suitable for security-dependent contexts

Malware uploading and importation alongside the integration of the Malware Bazaar were conducted in Microsoft Azure DevOps to allow for cloud simulation, which tested file security measures. Other steps included the integration of files and data logging during sequential and parallel encryption. The Application injected allowed for the consecutive transmission of files and storage of data while monitoring the files. During the experiment, there was documentation that was later put together into an Excel format, focusing on the report about the files input. Lastly, one had continuous observation of file performance until they were archived, which made it easy to obtain insights later regarding the cloud simulation of security systems. Fig. 1 describes the phases of the simulation.

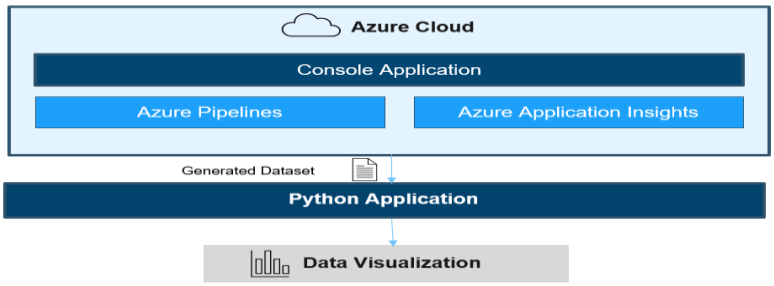

Fig. 1. Simulation in the cloud

3.2. Dataset overview

The malware samples were obtained from the Malware Bazaar, a platform created by abuse.ch for sharing malware. This platform has been used in previous studies [5, 6, 18]. By joining the platform, we obtained an API key that provided access to malware samples better suited for our specific research goals [39].

With the development environment (.NET Framework 4.8) in place, a system to automatically gather and analyze malware samples sourced from Malware Bazaar was devised. The Console application created to dispatch POST requests to the API was implemented as part of the system, where each request could bring, on average, a thousand file samples. Every sample's SHA-256 hash was collected from the file for retrieval and was kept in custom-specified directories. Aggressive error prevention techniques were also implemented to improve process workflows when the system encountered errors.

The final set of information involves 4451 samples, where each sample represents a file that has been processed using either SHA-256, MD5, SHA-1, or SHA-512 hash algorithms. The dataset contains critical information, which includes the file name, file size in bytes, the cryptographic fragmentation values, and the time taken to encrypt in milliseconds. In Table 2, the dataset column descriptions are presented.

Table 2. Dataset columns description

Column name	Description
File name	Specifies each file uniquely
File size (Bytes)	Represents the file size in bytes, providing insight into the volume of data processed
Sequential hash SHA-256	Includes the SHA-256 hash value of the file
Sequential SHA-256 encryption time (ms)	Time taken to calculate the SHA-256 hash (ms)
Sequential hash MD5	Includes the MD5 hash value of the file
Sequential MD5 encryption time (ms)	Time taken to calculate the MD5 hash (ms)
Sequential hash SHA-1	Time taken to calculate the SHA-1 hash (ms)
Sequential SHA1 encryption time (ms)	Time taken to calculate the SHA-1 hash (ms)
Sequential hash SHA-512	Includes the SHA-512 hash value of the file
Sequential SHA-512 encryption time (ms)	Time taken to calculate the SHA-512 hash (ms)

All files can be distinguished from one another based on their names. The amount of power utilized is expressed in particular time of the sampling interval. MAE, as well as the mean squared error, are extensively used as instruments of measure of effectiveness. Several SHA security variations are stored, including SHA-256, MD5, SHA-1, and SHA-512. The time lapses measured were each round of hashing secured and measured in milliseconds.

Confirmed data integrity indicates that there are no missing values reported in any sample. This steadiness simplifies handling and guarantees the precision of the analyses. This attention to detail in data collection and how it has been handled gave us a data set enabling us to study file sharing security in cloud computing environments. The histograms were generated to depict the distribution of encryption times per algorithm. Encryption times are presented in the distribution form in Fig. 2.

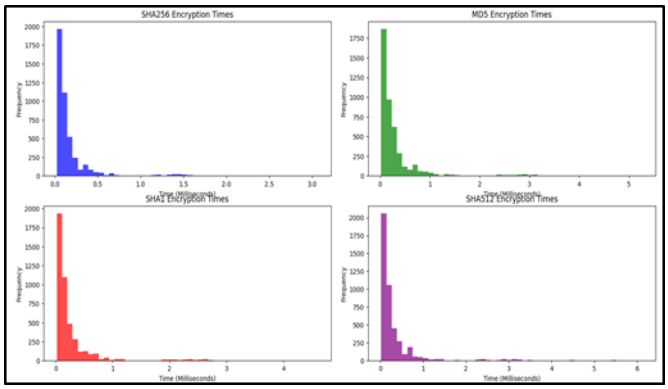

Fig. 2. Bifurcation: Plot of local maxima of x with damping a decreasing

Initially, the data was cleaned up using Python to ensure data integrity. Cleaning involved verifying the lack of missing values and confirming that no imputation was required. All missing value counts are zero, which guarantees the completeness and integrity of the dataset. This confirms the reliability of the data before proceeding with performance evaluation. The encryption programming was done in the NET Framework, which guaranteed the data's safety. Python (Pandas and NumPy) was utilized in statistical computations (means and standard deviations), and graphical representations.

4. Experiment results

4.1. Review stage

This performance analysis evaluated the encryption time of the four algorithms, which were SHA-256, MD5, SHA-1, and SHA-512, based on the 4461 operations that each algorithm performed. The following statistics provide insight into their efficiency and their usefulness throughout the system; summary statistics for encryption times are reported in Table 3.

Table 3. Summary statistics of encryption times

Measure	Encryption time			
	SHA-256	MD5	SHA-1	SHA-512
Count	4461	4461	4461	4461
Mean	0.176366	0.327912	0.266109	0.329803
Std	0.267132	0.551195	0.435188	0.568309
Min	0.0241	0.0217	0.0201	0.0232
25%	0.0558	0.0791	0.0728	0.0778
50%	0.1021	0.1748	0.1352	0.1742
75%	0.1716	0.319	0.2637	0.3272
Max	3.0661	5.2696	4.6051	6.1195

Count: The count depicts the total number of encryption operations done for each algorithm. Since each algorithm thoroughly completed 4461 operations, this uniform count ensured balance.

Mean: Mean is the average time of every single algorithm in performing an encryption operation.

SHA-256: The average encryption time for SHA-256 was found to be 0.176366 s, which suggests that the algorithm is comparatively efficient.

MD5: With a higher mean score of 0.327912 s, it can be foreseen that Average MD5 is greater than the mean SHA-256.

SHA-1: Mean encryption time for the SHA-1 algorithm was 0.266109 s, which places it slightly above the two mean times for both SHA-256 and MD5 in terms of performance.

SHA-512: The algorithm had a mean of 0.329803 s, suggesting that only an average of MD5 and SHA-1 is faster than it; it can be inferred that it is the slowest one of the four.

Standard Deviation (std):

This suggests the amount of variability that exists within the time taken to encrypt the algorithms. A high standard deviation suggests higher variability.

SHA-256: The standard deviation came to 0.267132, which indicates that there is a medium amount of variation.

MD5: Calculated value of standard deviation for MD5 is 0.551195, which denotes high variability, which indicates that the mean of encryption time can be substantially skewed away from the mean.

SHA-1: The standard deviation of 0.435188 value suggests that there is significant variability in the case of SHA-1, also, but it is less than the one faced in the case of MD5.

SHA-512: Computed standard deviation of 0.568309, not only is the highest amongst the range of algorithms, but also indicates that SHA-512 consistently shows extreme deviation in encryption times.

Minimum:

The minimum reflects the quickest recorded encryption time.

SHA-256: The encryption time ranged from 0.0241 s, which is the lowest recorded.

MD5: the lowest recorded encryption time on this algorithm stands at 0.0217 s.

SHA-1: In the best envisaged conditions, it was able to perform an encryption in 0.0201 s, which is the least of all.

SHA-512: This algorithm performed considerably well with average encryption time ranging between 0.0232 and 0.0249 s, which is equal to the instruments above it.

25th Percentile (25%):

From this percentage, it can be observed that 25% of the operations are completed quite a lot faster than others.

SHA-256: Samples were taken at 0.0558 s, which is on the lower end of the gram, indicating that there is a rate of 25% of the samples performed faster.

MD5: 0.0791 s, at which MD5 may be carried out.

SHA-1: 0.0728 s, this also suggests that there are fewer samples carrying out its operations faster than its pod standard.

SHA-512: This delay was measured to be around 0.0778 s for the same operation, but much slower.

50th Percentile (Median):

Medians stand for the average encrypted time, which is greater than half of the encryption done, while the other half is the one that did not reach that time.

SHA-256: The average time on 0.1021 s of the previous sample was met with a majority of the encryption, while 103 sample files took longer.

MD5: 0.1748 s, the longitudinal encrypted median range was much lower.

SHA-1: The median, however, did reach 0.1352 s, which left enough space for a sample that interfered with Esper.

SHA-512: The average time interval is 0.1742 s, which is alright. Also, it confirms that SHA-512 is not too fast, like MD5.

75th Percentile (75%):

As per this percentile, 75% of encryption algorithms did better than this value.

SHA-256: 0.1716 s, this is quite close to the mean, therefore showing how consistent it is.

MD5: 0.319 s, this time is quite high in comparison, which indicates that three-fourths of the operations were performed at a faster pace.

SHA-1: 0.2637 s, which is longer than SHA-256.

SHA-512: 0.3272 s, as always, manager of the four.

Maximum:

The maximum value is the worst-case scenario in reality, as it is the slowest encryption time ever recorded.

SHA-256: Max time for encryption is 3.0661 s, so modus and median have some outliers to suggest the process took an unusual amount of time.

MD5: Maximum time taken is 5.2696 s, making it the most hybrid and, if I dare say so, the slowest at certain events.

SHA-1: 4.6051 s, which shows a performance that is at the lowest battlegrounds.

SHA-512: The rest time of 6.1195 s is the longest, making it one of the indicators of inefficiency for some conditions.

The analysis shows that out of most, SHA-256 was quickest and shafting but MD5 and SHA-512 out of all were slow and showed a tilt, and quite unfortunate to use in cloud encryption.

4.2. Performance analysis

During this phase, an analysis was carried out with the aim of assessing the performance of the algorithms. In particular, the ANOVA test and Tukey's HSD test aimed at investigating and comparing the time taken to accomplish the cryptographic tasks for SHA-256, MD5, SHA-1, and SHA-512 algorithms. The same test was applied to determine whether there are differences in the times taken to encrypt among the algorithms. Results showed an F-statistic value of 104.62 and a p-Value of 3.87×10^{-67}, which is quite small. The low p-Value (<0.05) is suggestive of vast differences in the mean values of encryption times for various algorithms on average. After identifying differences among the cases using the ANOVA test, the HSD test is performed to determine which encryption times differ between pairs of algorithms. Results snapshots are depicted in Fig. 3.

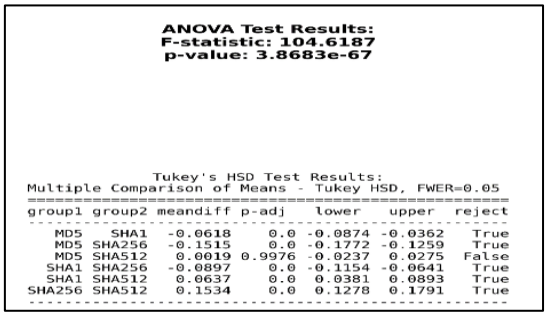

Fig. 3. ANOVA test and Tukey's HSD result

4.3. Parallel performance analysis

The efficiency was examined with regard to the volume of relationships and the time taken for encryption with reference to each algorithm. The time required to encipher all files sequentially in .NET and the parallel enciphering times were evaluated as reported in Fig. 4.

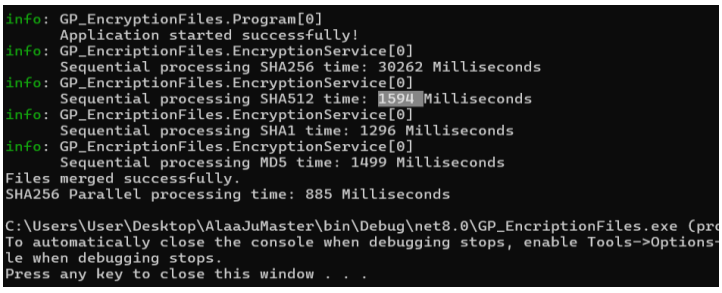

Fig. 4. Encryption time taken for each algorithm

5. Discussion

In the previous section, cloud-based computing security has been analyzed in the context of different encryption algorithms. Data procured from the Malware Bazaar site was worked on with software tools and statistical methods.

At the beginning, the average, range, from the minimum to maximum, and standard deviation, all such statistics were calculated for each of the algorithms' encryption times. The results are as follows:

SHA-256: The estimation of encryption speed and security using the SHA-256 algorithm appears to be moderate in nature with respect to the average value and standard deviation.

MD5: The MD5 algorithm was the fastest; however, its encryption times have been unstable, therefore, performance for the MD5 algorithm shows a higher degree of variation.

SHA-1: Its speed was a little less than MD5, but its performance measures were more reliable than those of MD5.

SHA-512: Encryption times based on this algorithm were the longest, owing to the fact of having an improved level of security but a lower speed of processing.

To provide a better understanding of the results, graphs were created that represented the spread of encryption times for every technique. From the charts, it was easier to comprehend the pattern of distribution of encryption times:

SHA-256: A majority of the data was on the average side, with a few outlier values observed.

MD5: Appeared to have a distribution that was skewed to more rapid times.

SHA-1: The distribution was noted to be similar to that of SHA-256.

SHA-512: The trend in encryption times was more pronounced, but it also meant slowness when in use, which was indicative of improved security performance.

The performance of different algorithms concerning encryption times was statistically analyzed using ANalysis Of VAriance (ANOVA) and Tukey's Honestly Significantly Different test (HSD). Findings confirmed that there were some notable differences between the two algorithms, where MD5 was faster than the other. A comparison of speed between SHA-256 and SHA-512 showed results that were comparatively better for SHA-256 than with SHA-512 that was slower. MD5 and SHA-512 had no significant differences in time taken for hashing. SHA-256 has been proven to be very efficient, as it can withstand collisions as well as make use of data for integrity checking, making it ideal for secured zones.

Security is a critical concern, which is why both MD5 and SHA-1 should not be used, as they are vulnerable to attacks. Despite fast hashing algorithms, they are inherently insecure and biased in terms of security. A better option would be SHA-512, which is not only safe from attacks but is also efficient. Flaws in MD5 and SHA-1 render them unusable for security applications, so it is better not to use them. For applications where a longer time is taken while encrypting, SHA-512 is the algorithm of choice.

Parallel processing technology has been incorporated to improve the SHA-256 algorithm encryption performance. The analysis yielded the following results (Fig. 4).

Sequential encryption time for SHA-256: 30,262 ms.

Parallel encryption time for SHA-256: 855 ms.

That is, by implementing parallel technologies, the performance of SHA-256 increased by nearly 97.07%.

While evaluating the findings of our study in relation to past studies, it is apparent that a deep congruence exists in terms of the performance and security of the SHA-256 algorithm. All studies articulate the strong security fundamentals offered by SHA-256 and the usefulness of parallel techniques for performance improvement. Our findings indicate that parallelism is not only an effective strategy but also one that increases encryption efficiency, making it possible for the claims made in earlier studies concerning the role of parallelism in improving encryption algorithms to be justified.

The study confirms the efficiency and reliability of SHA-256 in securing the files stored in cloud environments. The analysis stresses the necessity of applying transformational techniques so as to boost effectiveness, as well as recommend development of hybrid encryption systems by combining SHA-256 with other algorithms in order to enhance security. These findings justify further exploration and improvement of cryptographic methods and techniques to enhance the safety and reliability of cloud infrastructures.

5.1. Impact of results on file security in the cloud

This research paper addresses file security concerns in cloud settings through the detailed analysis of the performance of encryption algorithms with a focus on the SHA-256 method. This subsection will describe how our research introduces improvement and enhancement over other file security systems in cloud environments in terms of practicality, security, and performance.

1. Enhanced output:

The results of the study showed that the application of parallel processing techniques is likely to greatly enhance SHA-256 encryption. As a result of employing parallel computing, the encryption time decreased by 97.07%. Such a performance improvement improves the capacity of processing large amounts of data at high speed and quickly, which is essential in cloud scenarios where large data distribution and quick feedback are needed.

2. Improving security

Because of its high collision resistance, guaranteeing data integrity and protection from tampering, the use of the SHA-256 algorithm is appropriate. This improves the safeguarding of data transferred or stored in the cloud, for instance, from online attacks. In addition, the application of parallel computing techniques offers a faster and more efficient method to achieve the same degree of security without weakening it.

3. Practical application in cloud environments

The findings were verified using Microsoft Azure DevOps real-world simulations, which confirmed that SHA-256 is a suitable cloud algorithm. Protection of the cloud environment has proven to be as simple as effective with the incorporation of encryption, thus providing a means of data security in the cloud. This implies that our results are not only of a theoretical nature but also contribute to the data security augmentation in practical aspects.

4. Parallel computing and enhancing system performance

The need for encryption parallelism in the process of cloud system encryption is fundamental. Multiple co-processors can be used to support the encryption and decryption processes concurrently, which can drastically cut down the time taken to complete these operations. This not only improves the efficiency of the system but also lowers the costs by optimizing the available resources.

6. Conclusion

The SHA-256 algorithm has been utilized to cope with cloud-based file encryption challenges, and it has been shown to perform well with the incorporation of parallel processing concepts. This research also evaluated some contemporary practices used in encryption and emphasized alternatives with better performance and security. The conclusion drawn here proposes that indeed the SHA-256 standard is a good starting point in the development of the encryption system for data, but its parallel processing variant is much more efficient when speed of encryption is a priority due to its ability to compute in real time, which has broader applications in cloud-based computing solutions. This brings to light how industries today are seeking better trade-offs between security and efficiency without sacrificing either when it comes to advanced encryption methods when dealing with confidential information.

The results of the research have been insightful in broadening the scope of cloud data security recommendations for growth across multiple dimensions. These include the enabling of new forms of encryption that will cope with quantum computing, which poses a more serious threat to information systems' security. In addition, for performance reasons, the researchers propose modifying the methods of encryption by merging strategies of two or more encryption methods in one function, for example, HMAC-SHA-256 and AES. The use of better parallelism techniques integration is often another opportunity because these techniques can help in further improving encryption performance. Furthermore, addressing data protection between multiple clouds and developing modern key management systems is crucial for enhancing security within cloud applications.

The resulting research recommends incorporating advanced technological means such as Artificial Intelligence (AI) and machine learning to address future societal threats. AI-equipped applications that offer real-time analysis could also better secure cloud information, especially if the cloud interactions are more efficient in terms of security. These trends will not only enhance the success of the executive processes but also improve the journey of the users by providing security

which is blend with convenience. This research serves as a springboard for subsequent research in these areas with the central goal of developing secure, effective, and intuitive cloud systems that can withstand the test of time and address new cyber threats.

References

1. Al-Qtiemat, E. M. A. N., Z. E. Y. A. D. Al-Odat. Examining Cloud Security: Identifying Risks and the Implemented Mitigation Strategies. – Journal of Theoretical and Applied Information Technology, Vol. **102**, 2024, No 7.
2. Amajuoyi, C. P., L. K. Nwobodo, M. D. Adegbola. Transforming Business Scalability and Operational Flexibility with Advanced Cloud Computing Technologies. – Computer Science & IT Research Journal, Vol. **5**, 2024, No 6, pp. 1469-1487.
3. Anusha, P., R. Maruthi, D. M. Kumar, H. P. Begum. Double Encryption Technique for Sharing and Storing Images in the Cloud Environment. – In: Proc. of 4th International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT'24), IEEE, 2024, pp. 1-4.
4. Arogundade, O. R. Strategic Security Risk Management in Cloud Computing: A Comprehensive Examination and Application of the Risk Management Framework. – International Advanced Research Journal in Science, Engineering and Technology, Vol. **11**, 2024, No 1.
5. Okazaki, N., S. Usuzaki, T. Waki, H. Kawagoe, M. Park, H. Yamaba, K. Aburada. Optimal Weighted Voting-Based Collaborative Malware Detection for Zero-Day Malware: A Case Study on VirusTotal and MalwareBazaar. – Future Internet, Vol. **16**, 2024, No 8, 259.
6. Haq, M. Y. M., A. Abhishta, S. Zeijlemaker, A. Chau, M. Siegel, L. J. Nieuwenhuis. Measuring Malware Detection Capability for Security Decision Making. – In: Proc. of IEEE European Symposium on Security and Privacy Workshops (EuroS&PW'24), IEEE, 2024, pp. 342-351.
7. Aydin, Y., A. M. Garipcan, F. Özkaynak. A Novel Secure S-box Design Methodology Based on FPGA and SHA-256 Hash Algorithm for Block Cipher Algorithms. – Arabian Journal for Science and Engineering, 2024, pp. 1-14.
8. Bezerra, J. I. M., A. Molter, G. Machado, R. I. Soares, V. V. D. A. Camargo. A Novel Single Kernel Parallel Image Encryption Scheme Based on a Chaotic Map. – Journal of Real-Time Image Processing, Vol. **21**, 2024, No 4, pp. 1-13.
9. Chauhan, R. Hybrid Approaches for Improving Cybersecurity and Network Intrusion Systems. – In: Hybrid Information Systems: Non-Linear Optimization Strategies with Artificial Intelligence, 2024, p. 153.
10. Chen, A. C. Evaluation of Advanced Encryption Standard Algorithms for Image Encryption. – In: Proc. of International Conference on Smart Systems for Applications in Electrical Sciences (ICSSES'24), IEEE, 2024, pp. 1-6.
11. Deshmukh, P. M., A. S. Gughane, P. L. Hasija, S. P. Katpale. Maintaining File Storage Security in Cloud Computing. – International Journal of Emerging Technology and Advanced Engineering, Vol. **2**, 2012, No 10, pp. 2250-2459.
12. Gilbert, H., H. Handschuh. Security Analysis of SHA-256 and Sisters. – In: Proc. of International Workshop on Selected Areas in Cryptography, Berlin, Heidelberg, Springer, 2003, pp. 175-193.
13. Gowthaman, A., M. Sumathi. Performance Study of Enhanced SHA-256 Algorithm. – International Journal of Applied Engineering Research, Vol. **10**, 2015, No 4, pp. 10921-10932.
14. Gupta, A., N. Banakar, C. Kumar, M. Aryan, U. Purushotham. Design and Implementation of an Efficient Fingerprint Authentication Algorithm Using SHA-512. – In: Proc. of 3rd International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE'24), IEEE, 2024, pp. 1-7.

15. G u p t a, S., P. K u m a r, A. S a r d a n a, A. A b r a h a m. A Secure and Lightweight Approach for Critical Data Security in Cloud. – In: Proc. of 4th International Conference on Computational Aspects of Social Networks (CASoN'12), IEEE, 2012, pp. 315-320.
16. H o s s a i n, M. A., M. B. H o s s a i n, M. S. U d d i n, S. M. I m t i a z. Performance Analysis of Different Cryptography Algorithms. – International Journal of Advanced Research in Computer Science and Software Engineering, Vol. **6**, 2016, No 3.
17. H u s s a i n, M. Z., M. Z. H a s a n, A. N. Q u r e s h i, G. M u s t a f a. Implementation of a Blockchain-Based Secure Cloud Computing Mechanism for Transactions. – In: Blockchain-Based Internet of Things, Chapman and Hall/CRC, 2024, pp. 146-166.
18. J a i n, S., S. T h a s e e n. Revolutionizing Malware Detection: Feature-Based Approach for Targeting Diverse Malware Categories. – In: Proc. of IEEE International Carnahan Conference on Security Technology (ICCST'23), IEEE, 2023, pp. 1-5.
19. J a i n, N., P. S i n g h a l. Secure Cloud Data Storage and Sharing. – Journal of Informatics Electrical and Electronics Engineering (JIEEE), Vol. **5**, 2024, No 1, pp. 1-12.
20. K i s h o r e, N., P. R a i n a. Parallel Cryptographic Hashing: Developments in the Last 25 Years. – Cryptologia, Vol. **43**, 2019, No 6, pp. 504-535.
21. L a n k e, R., A. M. Z. R a h m a n, R. B h a r d w a j, D. S. R e d d y, P. J a i n, T. R. M a h e s h. Cloud Cryptography: Mechanism of Different Encryption Standards. – In: Proc. of 11th International Conference on Computing for Sustainable Global Development (INDIACom'24), IEEE, 2024, pp. 356-360.
22. M a r q u i s, Y. A. From Theory to Practice: Implementing Effective Role-Based Access Control Strategies to Mitigate Insider Risks in Diverse Organizational Contexts. – Journal of Engineering Research and Reports, Vol. **26**, 2024, No 5, pp. 138-154.
23. M o h a m m e d, A. S., M. A l f u d a i l, A. A l k h u l a i w i, A. B a s h a r. Performance Analysis of Cryptographic Algorithms in Cloud Computing Systems. – In: Proc. of International Conference on Inventive Computation Technologies (ICICT'24), IEEE, 2024, pp. 1399-1405.
24. M o h a m m e d, Z. A., K. A. H u s s e i n. PRC6: Hybrid Lightweight Cipher for Enhanced Cloud Data Security in Parallel Environment. – Security and Privacy, 2024, e413.
25. M u s t a f a, N. A. A. Analysis of Attackers' Methods with Hashing Secure Password Using CSPRNG and PBKDF2. – Wasit Journal of Engineering Sciences, Vol. **12**, 2024, No 2, pp. 60-70.
26. M y i n t, S. M., M. M. M y i n t, A. A. C h o. A Study of SHA Algorithm in Cryptography. – International Journal of Trend in Scientific Research and Development, Vol. **3**, 2019, pp. 1453-1454.
27. O l a n r e w a j u, R. F., K. A b d u l l a h, H. D a r w i s. Enhancing Cloud Data Security Using a Hybrid of Advanced Encryption Standard and Blowfish Encryption Algorithms. – In: Proc. of 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT'18), IEEE, 2018, pp. 18-23.
28. R a w a l, B. S., S. S. V i v e k. Secure Cloud Storage and File Sharing. – In: Proc. of IEEE International Conference on Smart Cloud (SmartCloud'17), IEEE, 2017, pp. 78-83.
29. S a s i k u m a r, K., S. N a g a r a j a n. Comprehensive Review and Analysis of Cryptography Techniques in Cloud Computing. – IEEE Access, 2024.
30. S a t t a i a h, K., K. C h i n n a i a h. Providing Security in Genesis and Other Blocks of Blockchain Technology Using SHA-256 Algorithm. – In: Proc. of 3rd International Conference for Innovation in Technology (INOCON'24), IEEE, 2024, pp. 1-6.
31. S h a r m a, M. S. A Study on Integrated Crypto-Biometric System to Protect Against Unauthorized Access of Data. – Journal of Interdisciplinary and Multidisciplinary Research (JIMR), Vol. **19**, 2024, No 4.
32. S r i k a n t h, B., J. V. R. R a v i n d r a, G. A. E. S a t i s h, P. K u m a r, F. S h a i k. Implementation of an Improved High-Speed SHA-256 Algorithm from RTL to GDSII Using Verilog HDL. – In: Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough. Vol. **4**. Cham, Springer International Publishing, 2024, pp. 1-17.

33. S u, B., Z. H. U. Y o n g c o n g, J. P a n g, C. W a n g, J. L i, X. L i u. Design of Security Protection Framework for Power Grid Cloud Application API Based on Zero Trust. – In: Proc. of 8th International Conference on Energy System, Electricity, and Power (ESEP'2023), SPIE, Vol. **13159**, 2024, pp. 2035-2043.
34. T r a n, T. H., H. L. P h a m, Y. N a k a s h i m a. A High-Performance Multimem SHA-256 Accelerator for Society 5.0. – IEEE Access, Vol. **9**, 2021, pp. 39182-39192.
35. W a g h, A., S. Y a d a v, P. P a t i l, S. M a g d u m, S. S h i r a v a l e. Enhanced File Storage on Cloud Using Hybrid Cryptography Algorithm. – In: Proc. of 3rd International Conference on Power Electronics and IoT Applications in Renewable Energy and its Control (PARC'24), IEEE, 2024, pp. 398-404.
36. W a n g, J., G. L i u, Y. C h e n, S. W a n g. Construction and Analysis of SHA-256 Compression Function Based on Chaos S-box. – IEEE Access, Vol. **9**, 2021, pp. 61768-61777.
37. W a n g, Z., X. D o n g, Y. K a n g, H. C h e n, Q. W a n g. An Example of Parallel Merkle Tree Traversal: Post-Quantum Leighton-Micali Signature on the GPU. – ACM Transactions on Architecture and Code Optimization, 2024.
38. W u, R., X. Z h a n g, M. W a n g, L. W a n g. A High-Performance Parallel Hardware Architecture of SHA-256 Hash in ASIC. – In: Proc. of 22nd International Conference on Advanced Communication Technology (ICACT'20), IEEE, 2020, pp. 1242-1247.
39. Malware Bazaar (Online).
https://bazaar.abuse.ch/
40. K e t i p o v, R., R. S c h n a l l e, L. D o u k o v s k a, D. D e h e z. Managing Cybersecurity: Digital Footprint Threats. – Cybernetics and Information Technologies, Vol. **24**, 2024, No 3, pp. 151-162.

Received: 07.10.2025, Accepted: 28.11.2025