Methodology:

I fetched 4451 files from Malware bazar with difference filetype and i made different criteria to

Get best result.

The below following steps :

1-fetch 4451 files from malware bazar with difference data types and have malware

2-the t make encryption techniques separately and calculate time taken for each algorithm

Sequential processing SHA256 time: 81967 Milliseconds

Sequential processing SHA512 time: 12871 Milliseconds

Sequential processing SHA1 time: 13591 Milliseconds

Sequential processing MD5 time: 10965 Milliseconds

And merge data information in one excel sheet have all information of each algorithm   
  
  
rank these algorithms based on the time taken (performance) and also consider their security levels:

Performance Ranking (from fastest to slowest):

1. MD5: 10965 Milliseconds
2. SHA512: 12871 Milliseconds
3. SHA1: 13591 Milliseconds
4. SHA256: 81967 Milliseconds

Security Level (from most to least secure):

1. SHA256: Considered secure and widely recommended.
2. SHA512: Also secure, with a longer bit length than SHA256, but not always necessary depending on the application.
3. SHA1: Not recommended for security-sensitive purposes as vulnerabilities have been found.
4. MD5: Considered broken and vulnerable; not recommended for security-sensitive applications.

Add paralasim techniques for SHA256 and I got the bellow result

SHA256 Parallel processing time: 16264 Milliseconds

With compare the original processing time with the new time after implementing parallel techniques.

Original SHA256 processing time: 81967 milliseconds

SHA256 processing time with parallel techniques: 16264 milliseconds

With calculate the percentage improvement in processing time due to parallelization:

Percentage Improvement=(Original TimeOriginal Time−Parallel Time​)×100%

parallel processing techniques for the SHA256 algorithm resulted in an approximately 80.16% improvement in processing time.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Aspect** | **Thesis** | **A Comparative Study of Cryptographic Algorithms for Cloud Security (2018)** | **Study of Cloud Computing Security Methods: Cryptography (2019)** | **Security Analysis of Cryptographic Algorithms in Cloud Computing (2023)** | **A Comprehensive Meta-Analysis of Cryptographic Security Mechanisms for Cloud Computing (2016)** | **Analysis of SHA-1 in Encryption Mode (Handschuh et al.)** | **SECURELY - A Golang CLI Tool for Secure File Sharing (Khandagale et al.)** | **SECURELY - A Golang CLI Tool for Secure File Sharing (Khandagale et al.)** | **Security and Usability of Hashing Functions (Galaczová)** | **Security Analysis of SHA-256 and Sisters (Gilbert and Handschuh)** |
| **Primary Goal** | Enhance security and performance for shared files on the cloud using SHA algorithms and parallel processing. | Compare cryptographic algorithms for cloud security. | Explore security methods in cloud computing with a focus on cryptography. | Comprehensive security analysis of cryptographic algorithms in cloud computing. | Meta-analysis of cryptographic security mechanisms in cloud computing. | Analyze SHA-1 in encryption mode. | Introduce a secure file-sharing system using Shamir's Secret Sharing and AES-256 | Introduce a secure file-sharing system using Shamir's Secret Sharing and AES-256. | Discuss the security and usability of MD-5, SHA-1, and SHA-2. | Analyze the security of SHA-256, SHA-384, and SHA-512. |
| **Methods** | SHA algorithms, parallel processing, OTP. | Comparative analysis of cryptographic algorithms. | Detailed examination of cloud computing security methods. | In-depth analysis of cryptographic algorithms' strengths and weaknesses. | Meta-analysis of cryptographic security mechanisms. | Analysis of SHA-1's performance and security in encryption scenarios. | Implementation of Shamir's Secret Sharing with AES-256 and SHA-256. | Implementation of Shamir's Secret Sharing with AES-256 and SHA-256. | Examination of hashing functions' security and usability. | Security analysis of SHA hash functions. |
| **Key Findings** | Significant performance improvement with parallel processing; integrated OTP for extra security. | Identified effective cryptographic algorithms for cloud security. | Highlighted cryptography's importance in cloud security. | Insights into cryptographic algorithms' suitability for cloud computing. | Effectiveness and challenges of cryptographic mechanisms in cloud computing | Insights into SHA-1's applicability and security in encryption. | Superior processing and security for secure file sharing. | Superior processing and security for secure file sharing. | Evaluation of MD-5, SHA-1, and SHA-2's security and usability. | Vulnerabilities in simplified versions of SHA functions. |
| **Novel Contribution** | 80.16% improvement in SHA256 processing time with parallel techniques; added OTP security layer. | Comparative perspective on cryptographic algorithms' efficiency. | Emphasized cryptography's role in cloud security. | Detailed security analysis of algorithms in the cloud computing context. | Comprehensive review of cryptographic mechanisms in the cloud. | Analysis focused on encryption mode for SHA-1. | Confidential file-sharing system using a combination of cryptographic techniques. | Confidential file-sharing system using a combination of cryptographic techniques. | Security and usability analysis of basic hashing functions. | Security assessment of SHA-256 and its variants. |
| **Applicability & Relevance** | Diverse cloud environments needing balanced security and performance. | Suitable for selecting cryptographic algorithms for cloud security. | Understanding the cryptographic landscape in cloud security. | Choosing the right cryptographic algorithms for cloud scenarios. | Understanding cryptographic mechanisms' effectiveness and issues in the cloud. | Applicability in scenarios where SHA-1 is used for encryption. | Applicable for secure file sharing on various platforms. | Applicable for secure file sharing on various platforms. | Relevance to the application and security of hashing algorithms. | Importance of design choices in ensuring SHA functions' security. |