Journal of Global Research in Electronics and Communication
Volume 1, No. 11, November 2025
Available Online at: www.jgrec.info
RESEARCH PAPER

Redundancy of Collisions and Data Repetition: A State-of-the-Art AES Algorithms with SHA-512 to Support a High-Integrity IoT Storage

Chandralok Kumar
M.Tech. Scholar
Bhopal Institute of Technology
Rajiv Gandhi Proudyogiki
Vishwavidyalaya Bhopal
chandralokkr1990@gmail.com

Dr Arvind Kourav
Professor
Bhopal Institute of Technology
Rajiv Gandhi Proudyogiki
Vishwavidyalaya Bhopal
registrarbitsbhopal@gmail.com

Nehul Mathur
Asst. Professor
Bhopal Institute of Technology
Rajiv Gandhi Proudyogiki
Vishwavidyalaya Bhopal
mathurnehul@gmail.com

***Abstract*—The secure data transmission through the internet is a growing field of work that is vital in streamlining service delivery in modern infrastructure. The study destroys the age-old dilemma of optimization of redundancy in the data security across storage servers that serve data to multiple users and ownership systems. Older hashing algorithms such as MD5 that is prone to collision issues and lack of enough key lengths and vulnerability in sharing keys are often found in conventional methodologies. In order to eliminate these constraints, a new and effective algorithmic method is suggested. This model applies an Advanced AES-based encryption system (which is mentioned in the sources as an Asymmetric Encryption System) to guarantee high-level privacy and security of data. At the same time, SHA-512 is built in to calculate hash, allowing secure detection of files at a faster rate than the older system and redundancy detection through chunking of files on a file-level. The proposed methodology was simulated using MATLAB, and it was shown to work better than the old encryption methods in terms of such parameters as block size, hash output security, and time complexity.**

Keywords—Asymmetric Encryption System (AES), Simple Hash Algorithms (SHA), National Security Agency (NSA), Secure Data Transmission, MD5, Block Size, Word Size, Hash Output, Number of Rounds, Finite Field Construction.

I. Introduction

Cryptography is the mainstay of encryption that converts plaintext into ciphertext for preventing the interception of unauthorized targets. The security of the information transmitted is considered the paramount issue in the contemporary digital arena, covering trade secrets up to governmental informational resources. Advanced Encryption Standard (AES) that is a symmetric block cipher chosen by the U.S. government has found its way to be extensively used by other countries of the world as a means of securing sensitive information.

The development of data transfer infrastructure, especially the transition to large-scale, cloud-connected Internet of Things (IoT) systems is a significant challenge in ensuring high data integrity and efficiency. Lightweight and efficient security is required in IoT devices because they have limited storage and processing power.

There is a general lack of resource-constrained, high-volume cryptographic and data management solutions that exist in such an environment. The application of the conventional file-based scheduling is not adequate in preventing data duplication in enormous server data files because it does not trace inside data partitions. Also, the traditional hash functions, especially MD5, are weak because of collision vulnerabilities, which negatively affect their usefulness in content matching and de-duplication. The imperative of good, long, and powerful keys and effective key sharing algorithms also continues to be a thorny issue, particularly in a multi-user/multi-tenant setup.

This study describes the Advanced AES Algorithm Using Dynamic Key in the Internet of Things System, which presents a balanced approach that at the same time presents a strict security level and a high level of redundancy optimization. The suggested stronger AES encryption combined with SHA-512 hashing is aimed at the shortcomings of the old algorithms to create a system that is more secure and fast with increased reliability and much protection against data manipulation and unauthorized access.

II. Literature Review

One of the main research areas has been to integrate the strong security algorithms with the resource limits of the IoT devices. Literature often compares more normal ciphers such as AES with lightweight ciphers.

A. Comparison of AES and Lightweight Ciphers to use in IoT.

Other researchers have been working on the subject of lightweight cryptographic (LWC) algorithms. Featuring a comparison of the SIMON and SPECK algorithms, Rahul Neve et al. (2023) came up with a hybrid LWC algorithm based on SPECK key scheduling and SIMON round function logic. They also noted that they reduced the encryption and decryption time by half of the initial SIMON algorithm, which showed time and energy efficiency improvements.

The Baiq Yuniar Yustiarini et al. (2022) made a comparative method test of securing the IoT devices, comparing AES with Simon-Speck encryptions. The experimental results they obtained indicate that the Speck algorithm is better in communication delay and memory consumption than Simon and AES. According to an average, however, the Simon algorithm had the largest value of avalanche effect. Pejman Panahi et al. (2021) have conducted a detailed performance analysis of ten lightweight algorithms (among which are AES, SIMON, and PRESENT) on transmission through a cloud, focusing their attention on such primary factors as the memory consumption, energy

efficiency, and the timing of execution. Equally, Li Ning et al. (2020) suggested an assessment system to pick lightweight cryptographic ciphers (such as AES-128) in Internet of Health Things (IoHT) systems, based on the requirements of NIST and ISO.

B. *Improved AES and Hybrid Cryptosystems.*

The AES itself, in terms of its security, is still a subject of analysis, especially in the aspect of its implementation weaknesses. AES masking scheme was proposed by Shvartsman and Zhang (2019) based on the construction variation of finite field construction and random masking to ward off attacks of higher order power analysis. Their approach saved about 12 percent of the logic gates that were to be in the previous best design. Hafsa et al. (2019) suggested a new hybrid cryptosystem that is a combination of symmetric (optimized AES) and asymmetric (optimized ECC), which are shown to have lower execution time and power dissipation than earlier solutions.

More hybridizations are observed in the work of Iavich et al. (2018), who introduced a hybrid encryption model on the basis of AES and ElGamal cryptosystems. Although AES turned out to be more rapid in encryption/decryption, the asymmetric ElGamal was more secure. This was in order to enhance complexity and, therefore, enrich security. Kiruba and Vijayalakshmi (2018) used and evaluated data security through different modes of AES in a real-time IoT-based healthcare application.

C. *Resolving Critical Security and Data Integrity.*

The size of keys and integrity checking are important issues. Bhattacharjya et al. (2019) presented the Secure Hybrid RSA (SHRSA) messaging scheme that is based on a 9-layered cipher with a 1024-bit RSA modulus and aims at hiding the weaknesses of RSA and providing security against a variety of attacks. In Ghosh et al. (2019), the researchers developed a secure cloud data message transaction protocol based on AES with a variable-length key in Python, which can be applied in a financial or e-commerce setting.

The literature also points out that cryptographic algorithms are not sufficient to provide secure transmission, but firewalls, authentication, and message partitioning are necessary. This background supports the two-fold nature of this work on data integrity by hashing and confidentiality by improved AES.

III. Proposed Methodology

The overall purpose of this proposed study is to create a safe and reliable algorithm framework to implement high data privacy and integrity as well as optimum data storage redundancy in IoT systems. The strategy is a moderate technique that works on the basis of securing improvement and data duplication prevention.

A. *Advanced Cryptography Selection.*

To overcome the drawbacks that were described in the previous approaches (Section I), the framework adopts two important algorithms:

- **Advanced Encryption Standard (AES):** The system involves the use of an Advanced AES based encryption system. The earlier AES implementations (e.g., AES-128) used 10 rounds; however, the necessary security level requires the application of AES with 256 bits of the key (offering 14 rounds to secure top-secret information). Switching to the AES-256 greatly enhances security over earlier standards such as the DES (56-bit key, 64-bit block size), which gives a 256-bit block size.
- **SHA-512 Hashing:** To overcome the collision weakness of MD5 and make more efficient and secure file detection, the system incorporates SHA-512. SHA-512, part of the SHA-2 family, will produce a 512-bit message digest, with much greater security than either MD5 (128 bits) or SHA-1 (160 bits). The breaking effort is also based on 2512-bit operations, which is much more than the 2128 operations of MD5.

B. *Proposed System Workflow*

The whole system is divided into two parts, which are modularized and work sequentially: duplicate checking and then secure upload.

1) *Section 1: Uploading of files and duplicate checking.*

When a user tries to upload a file to the data server, the system will start a redundancy check, which is content hashing:

The system uses the powerful hash algorithm in SHA-2 (SHA-512) to compute the hash of the file.

This special hash value is at once matched with the current hash values in the internal database of the data server.

In case the computed hash of the file is discovered to be the same as an existing record (the file is a duplicate), the uploading process is terminated, and the user is informed ("This file already exists in the data server 310). This is done through efficient reduction of redundancy based on file-level chunking.

2) *Section 2: Upload and Storage of Encrypted Files.*

In case the file is considered unique, the Advanced AES encryption process is used:

The encryption and decryption of the unique file are done by the encryption function of AES (file, key).

The ensuing encrypted data (ciphertext) and the distinct hash value are then uploaded and stored in the data server.

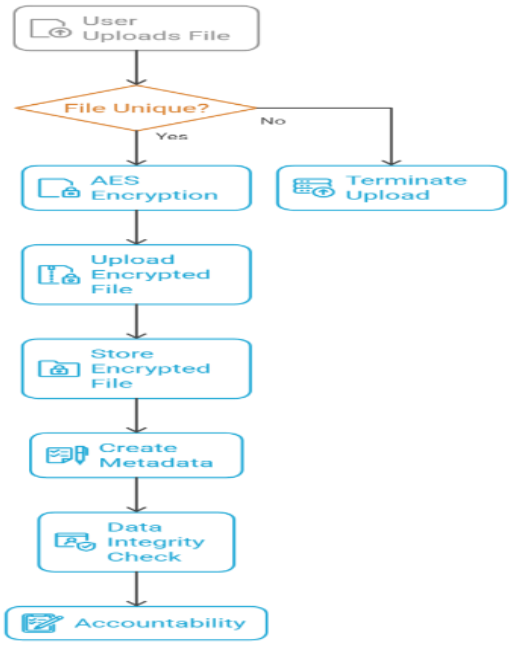

Fig. 1. Secure File Upload and Storage System

Data Integrity and Accountability: To address secure data transmission issues and achieve integrity, the server has a metadata concept. The record contains important data regarding the uploaded file such as; date, time, user ID, hash value and file path. This metadata enables users to confirm their data on the server on a time-to-time basis and its security, as detailed in Figure 1.

C. *AES and SHA-512 Internal Description.*

The AES algorithm uses a column-major 4 x 4 array of bytes (state) to work with. It is a process which is carried out through repetition of four transformations: Sub Bytes, Shift Rows, Mix Columns, and Add Round Key. These transformations are determined by the choice of AES-256 and make 14 rounds.

SHA-256/512 algorithms make use of the Davies-Meyer structure in their one-way compression function. An example of such a hash-based algorithm is SHA-256, which uses 32-bit words in 512 block with 64 independent steps. SHA-512 is used particularly due to the capacity to produce a much longer, and hence more secure, message digest than its predecessors.

IV. Simulation and Results

The algorithm was totally modeled and run in the Matrix Laboratory (MATLAB) software, version R2015a, in a Graphical User Interface (GUI) platform. The simulator was run on a system which was set up with Intel(R) Quad Core (VM) i5-3110 CPU at 2.40 GHz with a RAM amount of 4GB. The aim of the simulation was to compare and quantify the performance metrics of the proposed system, that is, the encryption time and the hash security in comparison to the old methods.

A. *Comparative Algorithms Analysis.*

The choice of AES-256 as an encryption algorithm and the use of SHA-512 as a hashing algorithm is explained by the inherent characteristics of the algorithms in comparison with their predecessors:

1) *AES vs. DES Comparison*

The shift to Advanced AES (AES-256) offers a major increase in security and key/block size over the older Data Encryption Standard (DES) that is now deemed insufficient, as shown in Table I.

TABLE I. Evaluation of DES and AES-256 Encryption

Metric	DES	Proposed AES (AES-256)
Key Length	56 bits	128, 192, or 256 bits
Block Size	64 bits	256 bits
Security	Lack of proven adequacy	Considered secure

2) *Comparison of the security of Hash Functions:*

SHA-512 offers a message digest that takes exponentially more operations to crack down, which has been achieved in ensuring integrity and de-duplication involving the possibility of collision, as shown in Table II and Figure 2

TABLE II. Security Analysis of Hash Functions

Comparison Keys	MD5	SHA-1	SHA-512
Security Level	Less Secure	More Secure	Highly Secure
Message Digest (bits)	128	160	512
Attacks Necessary to Break	2^{128} operations	2^{160} operations	2^{512} operations

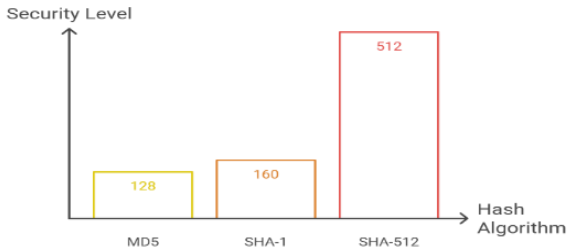

Fig. 2. Security Level of Hash Algorithms

B. *Time, Performance, and efficiency of encryption.*

Another essential indicator of performance in data transmission infrastructure is the Encryption Time H(t) and this is the time obtained to compute the value of the hash of the output. The less time, the more efficient it was in file detection and de-duplication. The simulation was used to compare the proposed SHA-512 algorithm with MD5, SHA-1 and SHA-256 with different file sizes (5 kb to 50 kb), as shown in Table III and Figure 3.

TABLE III. Performance of Encryption Time

File Size	MD5 (ms)	SHA-1 (ms)	SHA-256 (ms)	SHA-512 (ms)
5 KB	2.086	1.854	1.550	0.781
50 KB	2.951	3.348	2.988	1.731

By the findings, it was established that the SHA-512 implementation was more secure and faster than previous methods since the generation time of hashes was the fastest of all the file sizes tried.

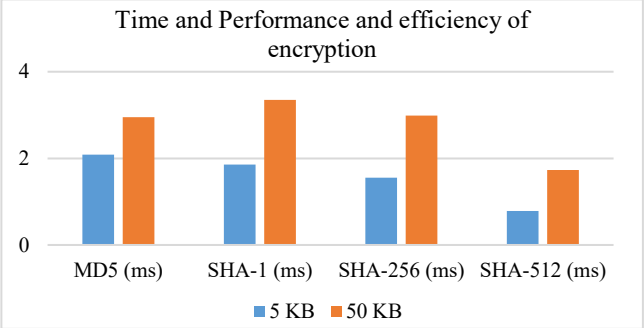

Fig. 3. Time and Performance, and efficiency of encryption

This performance advantage ensures that the duplicate file checking mechanism of the system can work with a low level of latency, which decreases the cost of the computation process and increases its security.

C. *Data Server and Control over the Administration.*

The simulation involved graphical illustrations of the file uploading procedure, computation of the hash and the administrative data server window. It is depicted that all the records will be stored in the data server, such as user login/logout time, and a metadata table of uploaded files. This metadata administration (user ID, hash value, date, time) is essential to trace the activities of users and establish a secure transmission of the data.

V. Conclusion and Scope of the Future.

A. *Conclusion*

The twin issues of data privacy, security, and effective redundancy optimization are effectively resolved in the

implementation of the Advanced AES Algorithm Using Dynamic Key in the Internet of Things System. Through the use of an Advanced AES-based encryption scheme and a combination of the high-security, high-speed hash algorithm fittingly known as the SHA-512, the proposed approach is effective in addressing the weaknesses with MD5 collisions and insufficient key lengths. The framework with the help of file-level chunking and an obvious two-part upload procedure prevents data corruption and avoids duplication effectively. The MATLAB simulation reveals that the optimized system has a better time complexity and cryptographic protection than the former encryption systems.

B. Future Scope

The improved methodology is set on the way to future developments:

- **Real-Time Deployment:** The framework will be created to support the possibilities of real-time deployment with the help of industry standards cloud infrastructure, which should be more reliable and secure than existing solutions.
- **Hashing Optimization:** Future studies can look into ways that hashing can possibly be avoided to make data available faster.
- **System Analysis Diversity:** The systems analysis and performance assessment is expandable into various operations systems and file format values.

REFERENCES

[1] A. L. Ananya et al., "Survey of applications, advantages and comparisons of AES encryption algorithm with other standards," Vol. 2, Issue 02, 22 Mar. 2023.

[2] R. Neve, R. Bansode, and V. Kaul, "Novel lightweight solution to perform cryptography on data security and privacy in IoT mobile devices," ISSN: 2147-6799, 16 Jul. 2023.

[3] B. Y. Yustiarini, F. Dewanta, and H. H. Nuha, "A comparative method to secure Internet of Things (IoT): AES vs Simon-Speck encryptions," 07 Sep. 2022.

[4] L. M. Shamala, G. Zayaraz, K. Vivekanandan, and V. Vijayalakshmi, "Lightweight cryptography algorithms to Internet of Things enabled networks: An overview," 2021.

[5] P. Panahi, C. Bayilmis, U. Cavusoglu, and S. Kacar, "Performance appraisal of lightweight encryption algorithms in IoT-based applications," 13 Jan. 2021.

[6] A. H. A. Al-Ahdal, G. A. Al-Rummana, G. N. Shinde, and N. K. Deshmukh, "NLBSIT: Lightweight block cipher design that is used to militarize data in IoT devices," 31 Oct. 2020.

[7] L. Ning et al., "A hybrid MCDM approach of choosing lightweight cryptographic cipher based on the ISO and the NIST lightweight cryptography security requirement of Internet of Health Things," 2020.

[8] A. Chandel et al., "Comparative analysis of AES and RSA cryptographic techniques," in 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), IEEE, 2019.

[9] A. Hafsa et al., "A new security approach to support the operations of ECC and AES algorithms on FPGA," in 2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), IEEE, 2019.

[10] P. Ghosh et al., "A variable length key based cryptographic approach on cloud data," in 2019 International Conference on Information Technology (ICIT), IEEE, 2019.

[11] B. Sonmez, A. A. Sarikaya, and S. Bahtiyar, "Machine learning-based side channel selection of time-driven cache attacks on AES," in 4th International Conference on Computer Science and Engineering (UBSE), IEEE, 2019.

[12] A. Bhattacharjya, X. Zhong, and X. Li, "A lightweight and efficient secure hybrid RSA (SHRSA) messaging scheme with four-layered authentication stack," IEEE Access, vol. 7, pp. 30487–30506, 2019.

[13] P. Shvartsman and X. Zhang, "Side channel attack resistant AES design using finite field construction variation," in 2019 IEEE International Workshop on Signal Processing Systems (SiPS), IEEE, 2019.

[14] M. Iavich et al., "Hybrid encryption model of ElGamal cryptosystem and AES cryptosystem in flight control systems," in 5th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC), IEEE, 2018.

[15] W. M. Kiruba and M. Vijayalakshmi, "Implementation and analysis of data security in a real-time IoT-based healthcare application," in 2nd International Conference on Trends in Electronics and Informatics (ICOEI), IEEE, 2018.

[16] N. Mekki et al., "A real-time chaotic encryption of multimedia data and application to secure surveillance system in the IoT system," in 2018 International Conference on Advanced Communication Technologies and Networking (CommNet), IEEE, 2018.

[17] E. H. Rachmawanto et al., "Secured PVD video steganography method using AES and linear congruential generator," in 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), IEEE, 2018.

