V10-2 (2023) 99–116
International Journal of Computers and Information
(IJCI)
Available online at *https://ijci.journals.ekb.eg/*

Security Evaluation of Different Hashing Functions with RSA for Digital Signature

Mahmoud Badawy[a,b]

[a] *Computer Science and Information Department, Applied College, Taibah University, Madinah 41461, Saudi Arabia*
[b] *Computers and Control Systems Engineering Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt*

Abstract

The emergence of the Internet of Things (IoT) represents a significant trend, where integrating IP, data, and wireless technologies onto a single network yields substantial advantages that are both essential and appealing. However, the amalgamation of these entities introduces novel susceptibilities and opportunities for infiltrating IoT networks, thereby necessitating the perpetual advancement of integrated security methodologies. This study investigates the feasibility of utilizing the Rivest–Shamir–Adleman (RSA) method, based on the Miller-Rabin technique, as a stream key generator with five distinct hashing functions to attain robust digital signatures. The RSA encryption method underwent a comprehensive battery of tests to assess its validity, and its encryption efficacy was evaluated through mathematical analysis. This research examines the assessment of digital signatures by utilizing five distinct hash functions in conjunction with RSA keys. Each signature file was assessed based on four tests: entropy, floating frequency, autocorrelation, and histogram analysis. The tests were conducted on a document with a size of 256 bytes. In addition, nine hash algorithms were utilized, namely SHA224, SHA256, SHA384, SHA512, BLAKE2B, BLAKE2S, MD5, MD2, and RIPEMD160. Different algorithms were used for varying key sizes and word counts for hashing. The experiment was repeated 100 times to obtain precise measurements of the average time and entropy. The findings indicate that when implemented with an appropriate key length, RSA exhibits both efficiency and sufficient security to be deployed in IoT networks.

Keywords: Digital Signature; Internet of Things; Hashing; RSA; Security.

1. Introduction

Encryption and decryption are two critical procedures in cryptography. Encryption encodes the initial data into a message that cannot be read as the original. In contrast, decryption converts the encoded message back to the initial message. Further, to achieve higher levels of security, data signing verifies the data's sender and typically includes some encryption in the process. Signing sensitive data, emails, and other online documents has become vital since it confirms the sender's identity and assures the data has not been changed during transmission.

Thus, Digital Signature [1] is a cryptographic technique used in signing electronic documents or messages to grant authentication [2]. Digital signatures are widely utilized nowadays within commerce and the financial industry (e.g., money exchange), marking digital contracts and, in numerous other scenarios, marked electronic records trade, marking exchanges within the public blockchains (such as exchange of tokens, coins, or other computerized assets) [3-6].

Digital signatures are unable to determine who has created a specific signature. [7]. This could be handled with digital certificates that link a person's identity to a public key proprietor (organization, individual, web location, or other). While establishing digital signatures, messages are bound to public keys rather than advanced identities. The public-key cryptographic algorithms (e.g., ECC and RSA) and public/private key sets are widely used in digital signature schemes. In Figure 1, the private key is for signing messages, and any signature process is verified using the correlated public key [8-10]. On the other hand, digital signatures are a relatively recent technological development that authenticates the sender's identity concerning sent digital data. The recipient or a neutral party could independently verify this binding [11-12]. Digital signatures are cryptographic values calculated with data and keys known only to those who signed the documents [13].

In practice, a message's recipient must check its authenticity before acting. Therefore, even the sender should not be able to disprove the message's authenticity. This criterion is crucial since the possibility of a data exchange disagreement is significant in commercial applications. Figure 2 presents the full model of the digital signature scheme as public key cryptography.

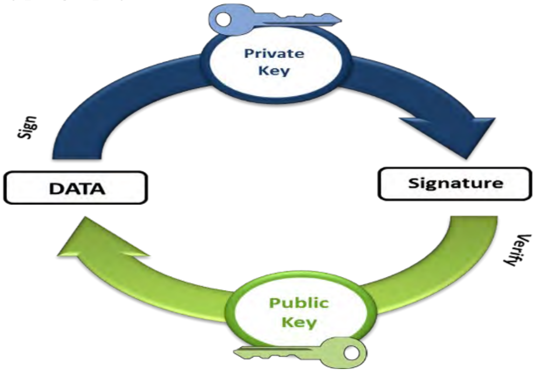

Fig. 1. Digital signature process

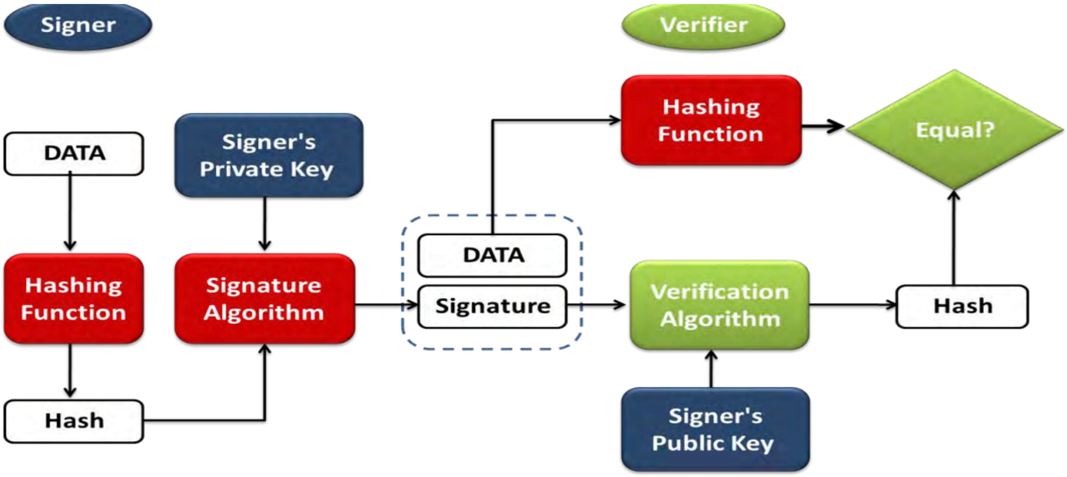

Fig. 2. Digital signatures full model

It ought to be noted that a hash of data is usually produced rather than immediately signing data via signing algorithms. The hash of data can be a unique representation of data. Thus, the hash can be signed in place of the data. The proficiency of schemes is the major fundamental cause for employing hashing techniques rather than data directly for signing. Signing massive amounts of data with modular exponentiation consumes time and computation costs [14]. Furthermore, the hash data may be a small digest of the data; consequently, signing hash values is more effective than fully data-signing [15].

Finally, this work has two-fold contributions: (i) presenting a detailed study about digital signatures and illustrating how digital signatures can improve online data security, integrity, authenticity, and non-repudiation. (ii) Evaluating different digital signature algorithms with hashing functions to secure Information from unauthorized access. This paper is structured and organized as follows: Section 2 discusses a background for digital signatures, followed by digital signature tools in Section 3. The work methodology is depicted in Section

4. Then, in Section 5, performance and security analysis are discussed. Section 6 introduces the security evaluation of nine hashing functions with RSA for digital signature. Finally, the paper is concluded and gives a summary of the future work in the last section.

2. Background

The digital signature techniques with public keys cryptography is a major and effective way out of all cryptographic primitive techniques because it achieves information security. In addition, Digital signatures provide messages with non-repudiation and can provide data integrity and message authentication. Briefly, digital signatures are important for achieving (i) Message authentication, (ii) Data integrity, and (iii) non-repudiation [16-18]. Finally, the cryptosystem provides the four key aspects of security: privacy, integrity, authentication, and non-repudiation, which can be established by combining public-key encryption with any digital signature technique. Digital signatures based on public key cryptography are widely recognized as one of the most important and useful tools for ensuring data integrity. In addition to the non-repudiation of the message, the digital signature also provides data integrity and message authentication. Digital signatures are important nowadays for (i) Data Integrity. (ii) Non-repudiation. (iii) Message authentication.

Furthermore, digital communication systems prefer exchanging encrypted messages to sending confidentiality plaintexts [19]. However, in schemes based on public-key encryption, a sender's public key is openly accessible; as a result, anyone can spoof it and transmit an encrypted message to the receiver [20]. As a result, users manipulating public key cryptography for encryption must acquire digital signatures amongst encrypted data to ensure message authentication and non-repudiation. There are two ways to accomplish this: (i) sign-then-encrypt and (ii) encrypt-then-sign. Although, a receiver can use sign-then-encrypt cryptosystems to impersonate the sender's identity and then send this data to a third party. So, this is why this procedure is not recommended. On the other hand, the encrypt-then-sign technique is extensively used and more dependable, as illustrated in Figure 3.

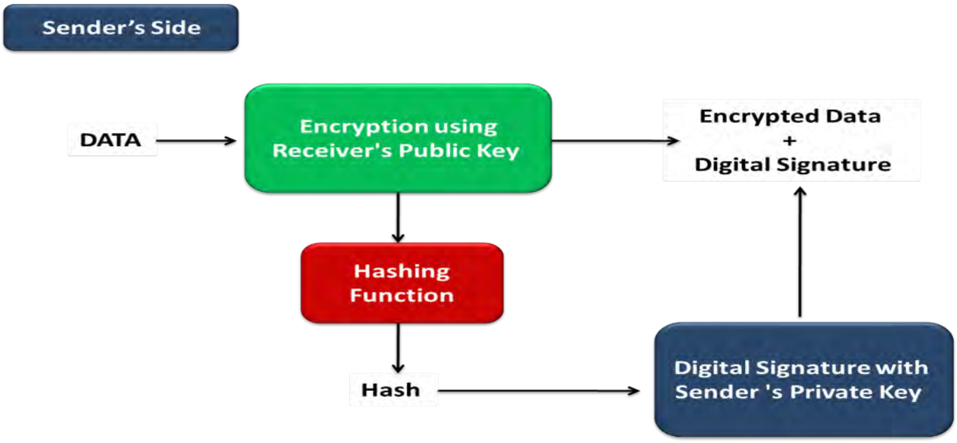
Sender’s Side
DATA
Encryption using Receiver's Public Key
Encrypted Data + Digital Signature
Hashing Function
Hash
Digital Signature with Sender 's Private Key

Fig. 3. The process of encrypt-then-sign technique

Most public-key cryptosystems, including RSA [21], DSA [22], and ECC [23], offer secure digital signature techniques. RSA, DSA, ECDSA [24], EdDSA[25], ElGamal [26], and Schnorr[27] are the most well-known digital signature schemes/algorithms. These signature systems were developed in response to the difficulties of the discrete logarithm problem (DLP) [28] and (ii) the elliptic-curve discrete logarithm problem (ECDLP) [29], both of which are quantum-breakable. Quantum-safe [30] digital signature schemes such as SPHINCS [31], BLISS [32], and XMSS [33] are not widely utilized due to their long key length, slow performance, long signature with the comparison to EdDSA or ECDSA, for example. The most used digital signature methods are those based on RSA, ECDSA, and EdDSA. Figure 4 summarizes the different algorithms for creating digital signatures. Table 1 [34-37] briefs the main characteristics of each digital signature algorithm, while Table 2 compares the main features of each hash function.

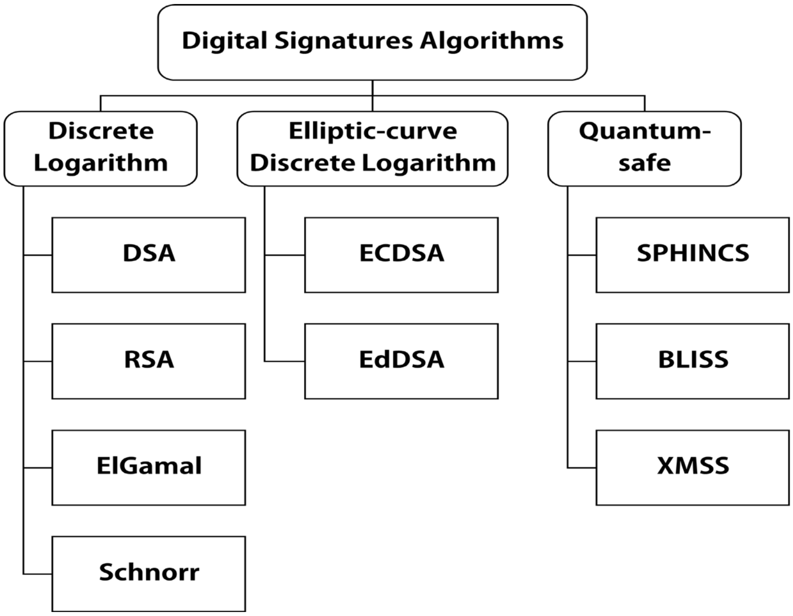

Fig.4. The digital signatures algorithms

Table 1. *Comparison of Digital Signature Schemes and Algorithms*

Algorithm	Type of Algorithm	Security Requirements	Simulation Speed
RSA [21]	Asymmetric	Specialized algorithms are needed to factor integers with specific qualities.	Fast
DSA [22]	Asymmetric	Random generators are required for private keys.	Fast
ECDSA [24]	Asymmetric	Random generators are required for generating keys.	Fast
EdDSA [25]	Asymmetric	A long key length that enhances the insecurities of ECDSA.	Fast
El-Gamal [26]	Asymmetric	Two phases of the key generator compute a single key pair for one user.	Fast

Table 2. *Comparison of the Main Characteristics of each Hash Function [38-43]*

Hashing Function	Msg. Block Size	The Size of Digest (bit)	Collision
MD2	128	128	Yes
MD4	512	128	Almost
MD5	512	128	Yes
SHA-0	512	160	Yes
SHA-1	512	160	Disability
SHA-256/224	512	256/224	No
SHA-512/384	1024	512/384	No
RIPEMD	512	128	Yes
RIPEMD-128/256	512	128/2556	No
RIPEMD-160/320	512	160/320	No
WHIRPOOL	512	512	No

3. Digital Signature Tools

After a detailed comparative study of existing software tools for digital signature with encryption tools, the list of software includes the following:

- Encyro: Send secure email and attachments to any email address without requiring recipient sign-up. Clients may transmit files and secure communications by clicking a single link. Advanced compliance tools let you request electronic signatures, and you may brand and customize the platform with your branding.
- Adobe Acrobat: to generate, transform, and exchange PDF documents, utilize Adobe Acrobat PDF editing software. It can convert documents between Microsoft Office formats and PDFs. In addition, the free Acrobat Reader mobile app may be used to view, annotate, and sign PDFs.
- DeliverySlip: A cloud-based system for email security, file sharing, and electronic signatures called DeliverySlip provide high-grade email encryption, secure file transfer, electronic approvals, online forms, bulk send, and more. The program functions inside Microsoft Outlook, Gmail, and several other programs.
- JotForm: The automatic e-signature process called Jotform Sign is created to make your workflow more efficient. The best method for gathering e-signatures is Jotform Sign, which has field recognition, an easy constructor, and connectors with excellent automation tools.
- eFileCabinet: Rubex is designed for back-office departments in every business, including those in HR, Insurance, and Accounting, to reduce tedious labor by streamlining their repetitive document procedures.
- Secure Exchanges: a technology that enables users to securely communicate, recover, and share massive files of private data through emails, as well as to provide the most secure digital signature available.
- Nitro Sign: Without a printer, paper, or pen, anybody can quickly, effortlessly, and securely sign documents with Nitro Sign. With unrestricted eSigning, corporate capabilities, and connection with the Nitro Productivity Suite, Nitro Sign allows you to stay digital and be productive from anywhere.
- MeSign: offers services for email encryption, digital signatures, and timestamping for your business emails. Automatically configure email certificates and encrypt email messages using the S/MIME standard. Both on-premises and cloud key management are supported.

4. Methodology

After a comparative study of primality testing for Fermat, Solovay Strassen, and Miller-Rabin tests, the Miller-Rabin test is more powerful than other tests [44-45]. So, public keys (e) and private keys (d) have been generated based on the Miller-Rabin test—equations from 1 to 3 present the Miller-Rabin calculations. The value range of the prime numbers (q) and (p) of the Miller-Rabin have been entered independently. The effectiveness of the RSA calculation depends on, among other things, the capacity to arbitrarily select two huge prime numbers, p and q. Table 3 compares the tested hash function's bit length.

$$e = \mathrm{phi}(N) \tag{1}$$

$$\mathrm{phi}(N) = (p-1)(q-1) \tag{2}$$

$$d = e - 1\,.\,\mathrm{mod}(\mathrm{phi}(N)) \tag{3}$$

Table 3. *Comparison of Tested Hash Functions Bit Length*

Index	Hash Function	Key Generation	Bit Length of Hash	Bit Length of N
1	MD2	RSA KEY	128	304
2	MD5	RSA KEY	128	304
3	SHA	RSA KEY	160	304
4	SHA-1	RSA KEY	160	304
5	RIPEMD-160	RSA KEY	160	304

Based on statistical testing and mathematical measurements, appropriate metrics are essential to explore the randomness degree and encryption efficiency for the binary sequences delivered by different digital signature algorithms. Those will be utilized for assembling proof whose yield sequences are random, have qualified encryption, and can be used safely in the applications of converged networks. Each statistical test decides whether these sequences have truly random attributes. The performance evaluations were implemented using Cryptool version 1.4.41 in a Windows 7 (64-bit) operating system with Intel Core i5-3317U @ 1.70GHz and 8 GB RAM. Figure 5 presents the scenario for creating signatures to test and validate the security of MD2, MD5, SHA, SHA-1, and RIPEMD-160 algorithms with RSA keys.

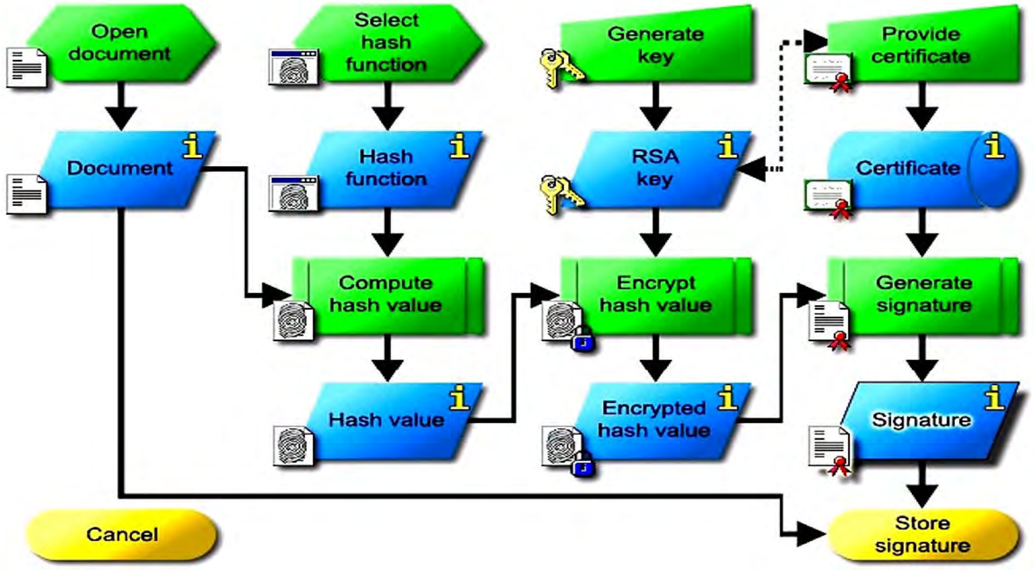

Fig.5. Digital signature for documents scenario

5. Performance Analysis

The next sub-sections illustrate the evaluation of digital signatures using five hash functions with RSA keys. Each signature file has been evaluated due to four primary tests (i) entropy, (ii) floating frequency, (iii) autocorrelation, and (iv) histogram. Executing these tests has been done on a 256-byte document.

5.1. Entropy

Entropy is randomness cryptographic frameworks utilize to produce cryptographic keys [46]. Therefore, great entropy is essential to produce strong keys. Table 4 summarizes the no. of different bytes in the signature document and the entropy of the whole document. Entropy values reflect that current hash algorithms have a moderate attitude but still need continuous development from researchers worldwide to achieve high security.

Table 4. *Entropy Comparison of Different Hashing Algorithms*

Index	Hash Function	No. of different bytes of 256 bytes	The entropy of the whole document	Periodicity	Security Level
1	MD2	85	3.31	Not Found	Weak
2	MD5	89	3.31	Not Found	Weak
3	SHA	100	3.90	Not Found	Medium
4	SHA-1	103	4.02	Not Found	Medium
5	RIPEMD-160	150	5.85	Not Found	Strong

5.2. Floating Frequency Analysis

It is defined as the characteristic of local Information in the document at individual points [47]. The floating frequency indicates the number of distinct characters detected in any 64-character document segment. Figure 6 presents a floating frequency analysis of signature files in MD2. The x-axis presents section offset, while the y-axis shows the number of unique characters in a 64-byte block.

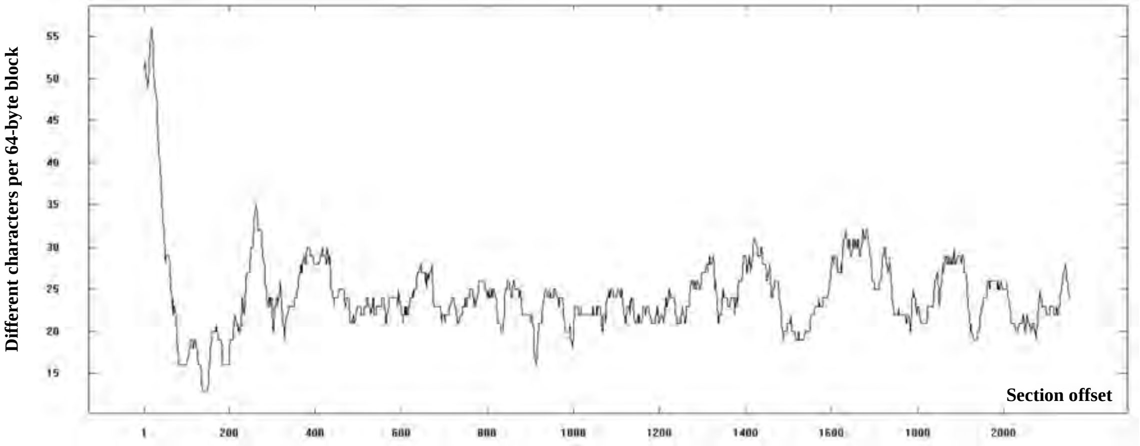

Fig.6. Floating frequency analysis of signature file using MD2

5.3. Autocorrelation Analysis

The autocorrelation of the bit sequence measures the proximity of different sequence parts [48]. For example, the autocorrelation of an encrypted file can sometimes be used to calculate its key length [49]. This experimental test of independence examines the relationship between successive outcomes of the pseudorandom number generator and between the binary arrangements and version of the sequence that additional positions have displaced. Figure 7 presents an autocorrelation analysis of signature files in MD5. The x-axis gives the offset, while the y-axis shows the no. of different matching characters.

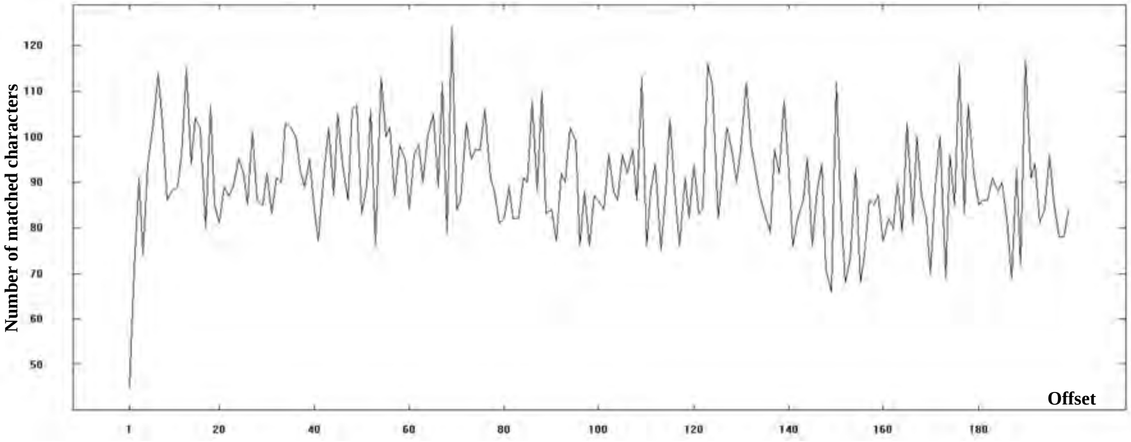

Fig.7. Autocorrelation analysis of signature files using MD5.

5.4. Histogram Analysis

A document's histogram graphically shows the character frequency distribution in a relevant window for a certain document [49-50]. The histogram's x-axis contains all characters within a specific character set: In a content window, the character set. While in a hexadecimal outputs and inputs window, this character set includes the values 0 through 255. Each character's frequency appears (as %) on the vertical axis. Figure 8 shows the histogram of the signature files of different hashing functions is uniform. The x-axis presents the value itself, while the y-axis shows the frequency.

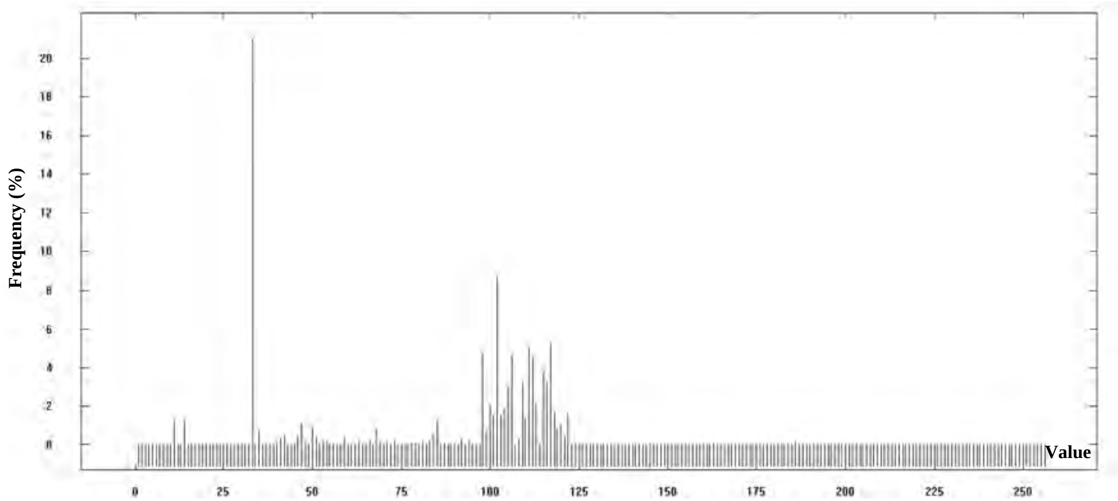

Fig.8. Histogram analysis of signature files of different hashing functions

6. Security Evaluation of nine hashing functions with RSA for digital signature

The digital signature mechanism is shown in Figure 9. Nine different hash algorithms are used for further analysis: SHA224, SHA256, SHA384, SHA512, BLAKE2B, BLAKE2S, MD5, MD2, and RIPEMD160. Each algorithm is applied for different key sizes and many words to hash. Every single experiment is applied 100 times to find the average time and entropy accurately.

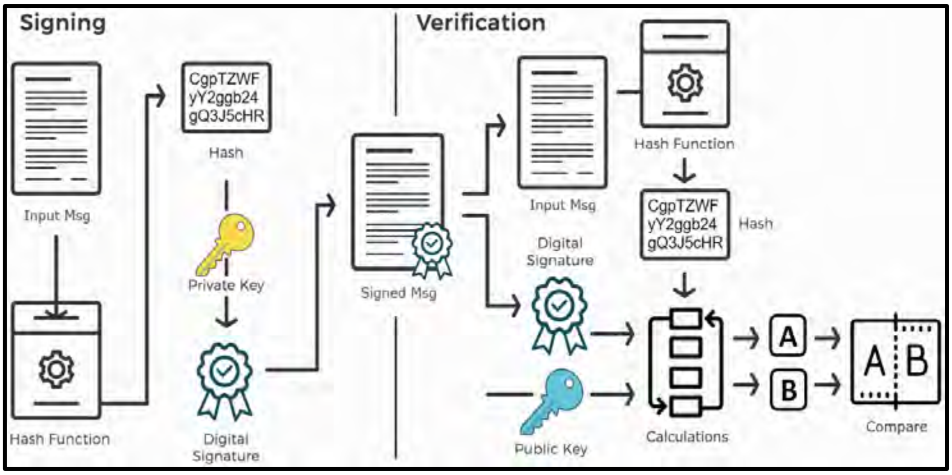

Fig.9 Digital signature mechanism

These experiments are done due to the following scenario:

(i) First step: Key Generation (using equations 4 and 5)

$$\mathrm{N} = \mathrm{p} * \mathrm{q} \tag{4}$$

$$\varphi = (p - 1) * (q - 1) \tag{5}$$

φ: is the Euler parameter. e: is coprime with φ where $\mathrm{e} > 1$ and $\mathrm{e} < \varphi$. The public key is (n, e), and the private key is (n, d).

(ii) Second step: Message Encryption (using equation 6)

$$C = M^{e} \ mod \ n \tag{6}$$

- Where M is the plain text, and C is the cipher text.

(iii) Third step: Message Decryption (using equations 7 and 8)

$$M = C^{d} \ mod \ n \tag{7}$$

$$d = e^{-1} \ mod \ \varphi \tag{8}$$

6.1. Experiments Configurations

The experimental configuration is shown in Table 5. Each specific experiment is repeated 100 times to accurately ascertain the average time and entropy. For instance, in an investigation involving the hash algorithm SHA224, a key size of 512 and a text size of 100,000 are used 100 times. The same methodology is applied to all other combinations.

Table 5. *Experiments Configurations*

Parameter	Configuration
Total number of hash algorithms	9
The hash algorithms employed	SHA224, SHA256, SHA384, SHA512, BLAKE2B, BLAKE2S, MD5, MD2, and RIPEMD160.
Employment	Each algorithm is employed with varying key sizes and a distinct count of words for hashing.
Key sizes	512, 1024, and 2048.
Text sizes (word count)	(100,000; 200,000; 300,000; 400,000; 500,000; ………., 1,000,000.

6.2. Results of Experiments

The hash algorithms utilized in this study include SHA224, SHA256, SHA384, SHA512, BLAKE2B, BLAKE2S, MD5, MD2, and RIPEMD160. These algorithms, with key sizes of 512, 1024, and 2048 bits, were employed on different text sizes 100 times. The aim was to identify (i) relationships between the number of words and the processing time, (ii) relationships between the key sizes and processing time, (iii) relationships between the number of words and the entropy of the hex digest, and (iv) relationships between key sizes and the entropy of the hex digest. Figures 10 through 27 present the experimental results for each algorithm.

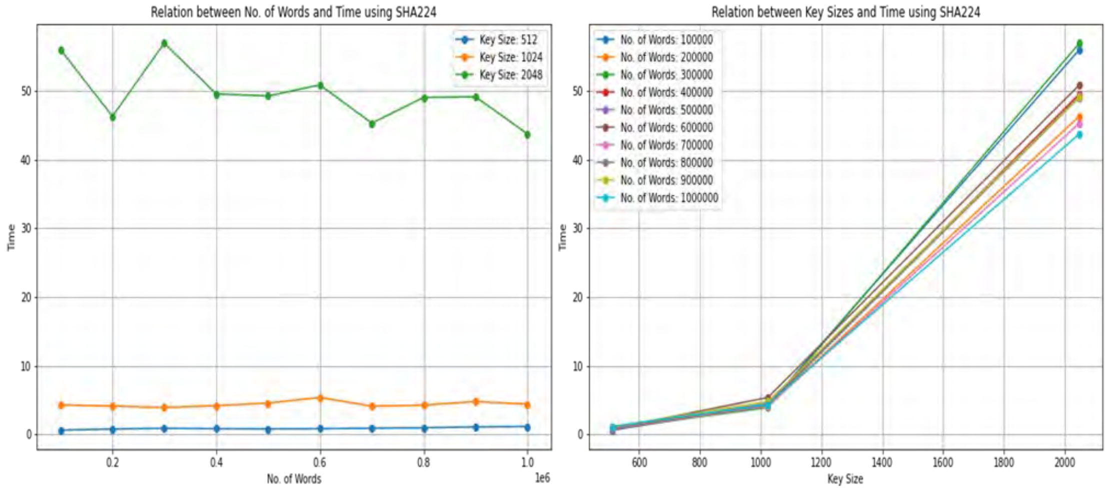

Fig. 10. SHA224: Relations between the number of words and time (left)- SHA224: Relations between key sizes and time (right)

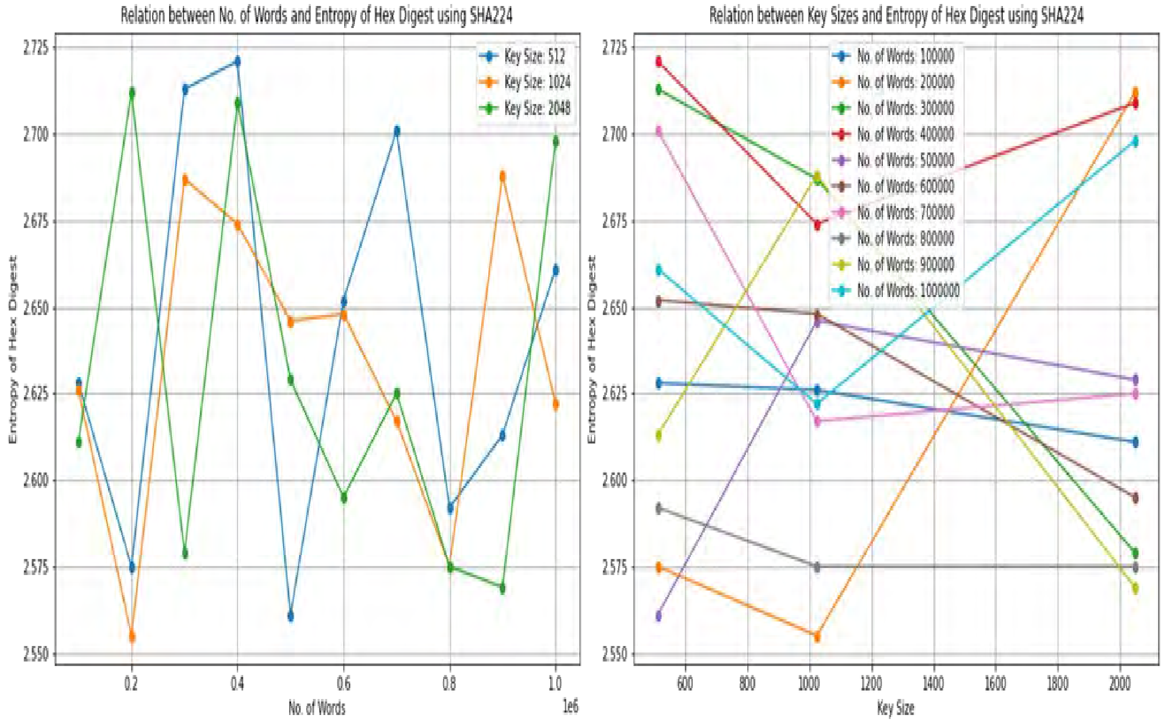

Fig. 11. SHA224: Relations between the number of words and entropy of hex digest (left), SHA224: Relations between key sizes and entropy of hex digest (right)

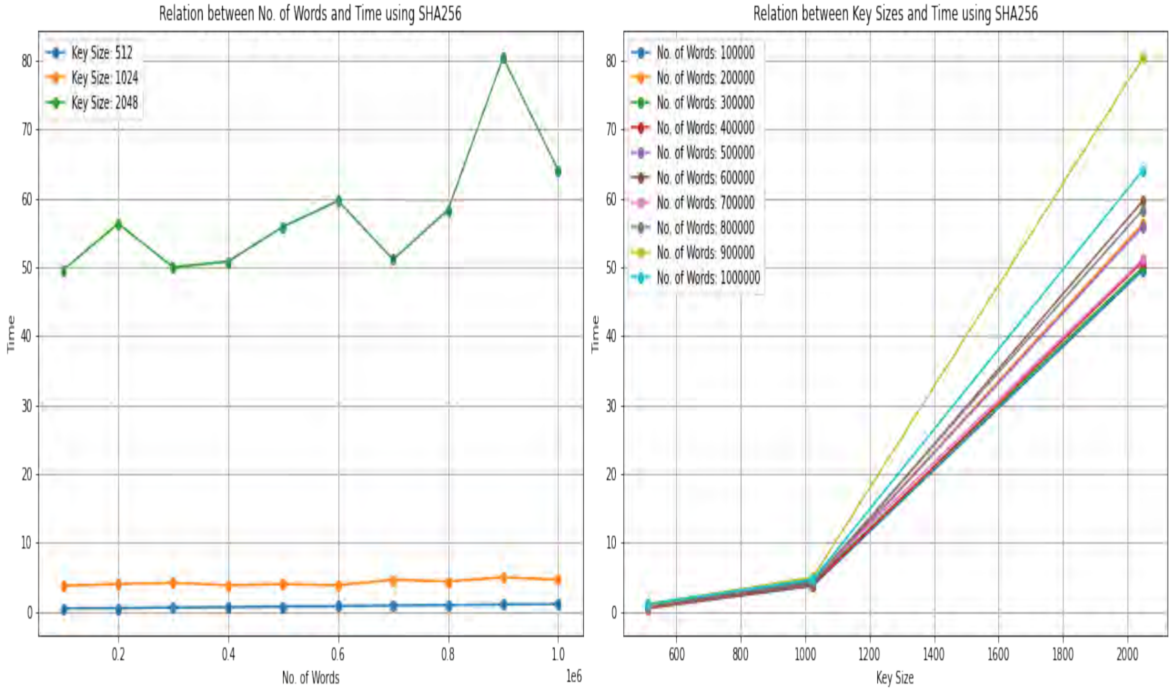

Fig. 12. SHA256: Relations between the number of words and time (left), SHA256: Relations between key sizes and time (right)

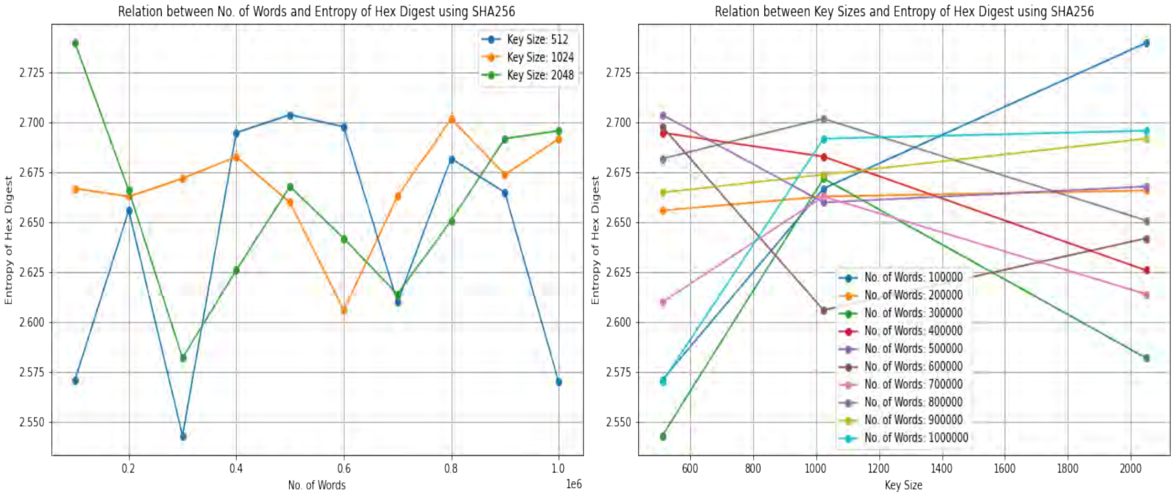

Fig. 13. SHA256: Relations between the number of words and entropy of hex digest (left), SHA256: Relations between key sizes and entropy of hex digest (right)

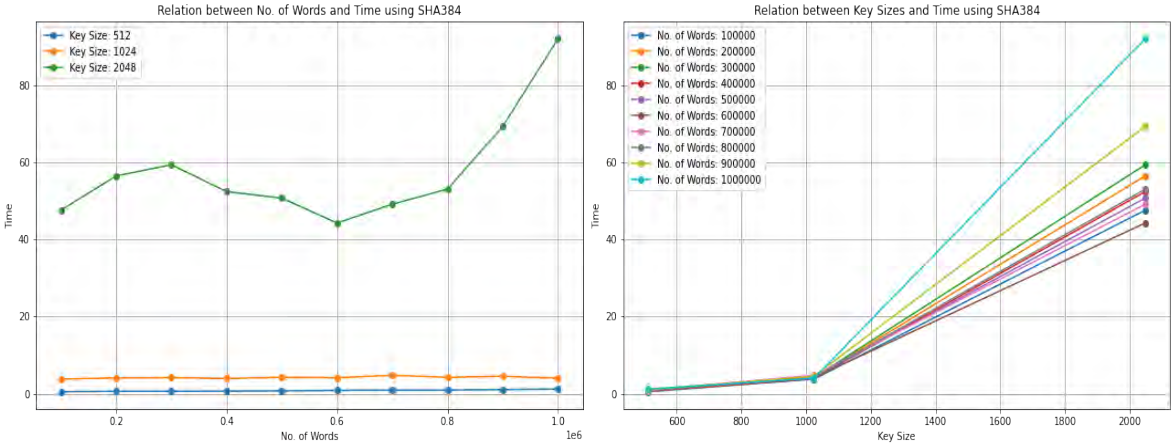

Fig. 14: SHA384: Relations between the number of words and time (left), SHA384: Relations between key sizes and time (right)

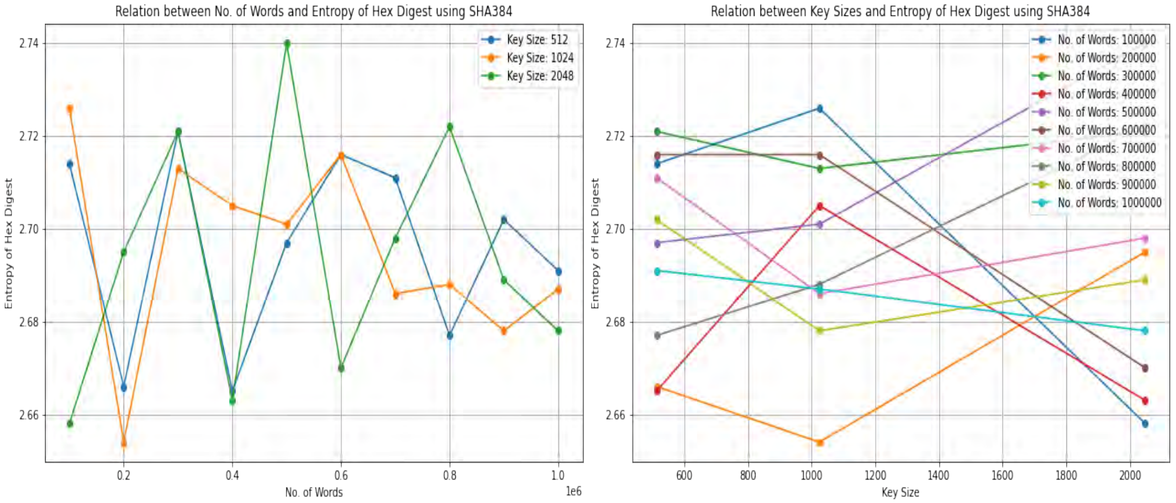

Fig. 15: SHA384: Relations between the number of words and entropy of hex digest (left), SHA384: Relations between key sizes and entropy of hex digest (right)

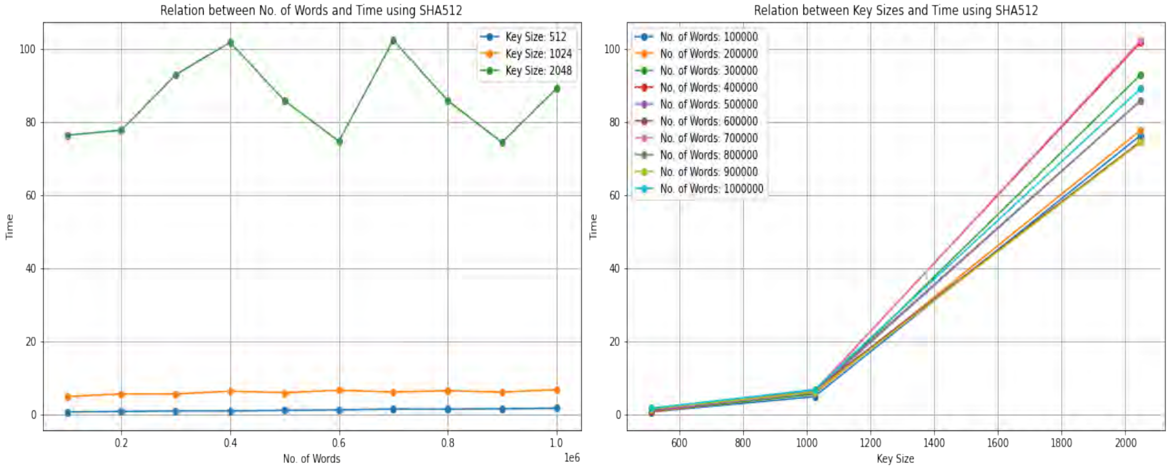

Fig. 16: SHA512: Relations between the number of words and time (left), SHA512: Relations between key sizes and time (right)

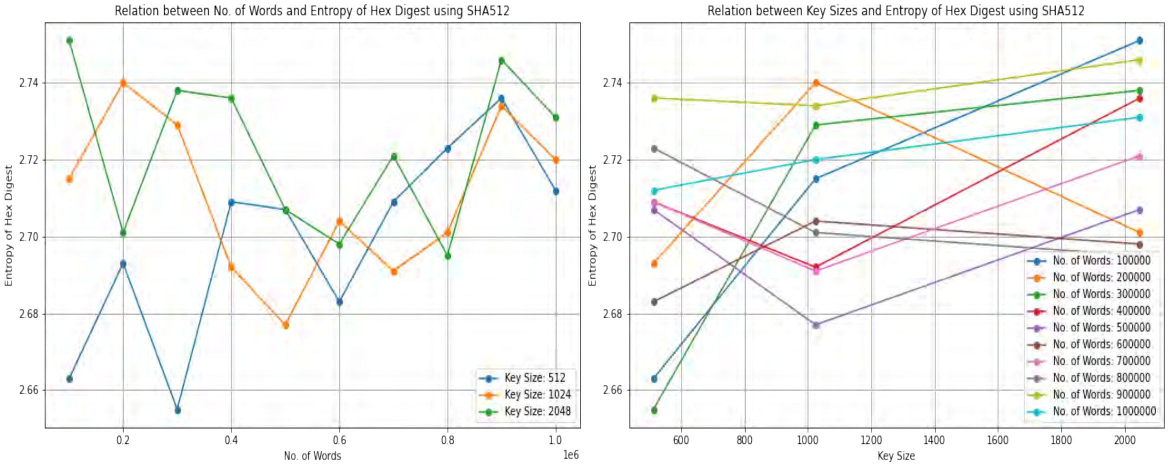

Fig. 17: SHA512: Relations between the number of words and entropy of hex digest (left), SHA512: Relations between key sizes and entropy of hex digest (right)

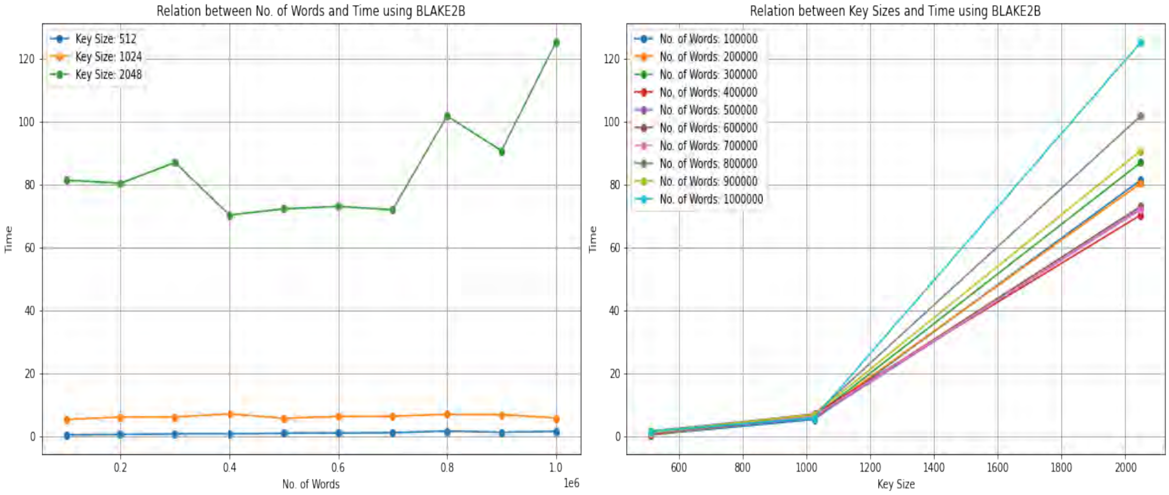

Fig. 18: BLAKE2B: Relations between the number of words and time (left), BLAKE2B: Relations between key sizes and time (right)

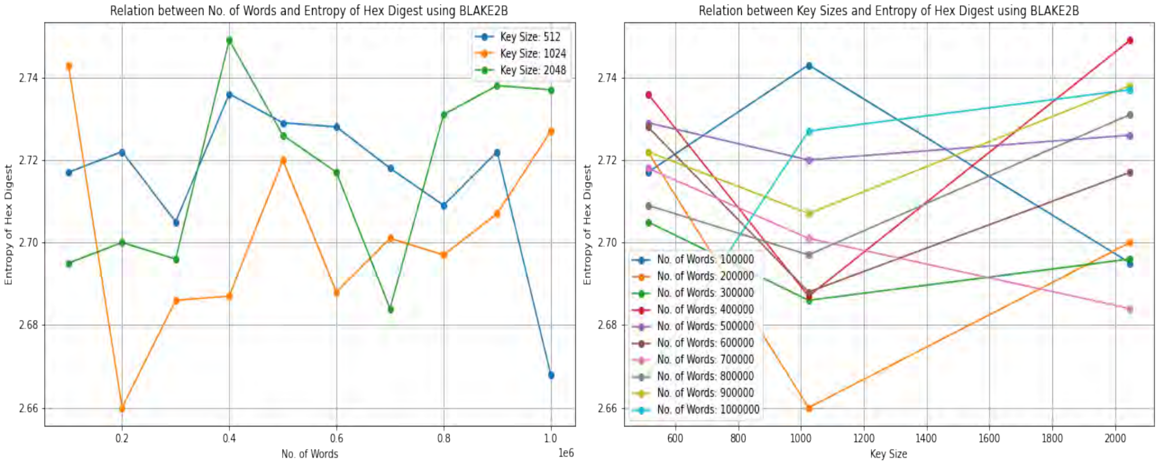

Fig. 19: BLAKE2B: Relations between the number of words and entropy of hex digest (left), BLAKE2B: Relations between key sizes and entropy of hex digest (right)

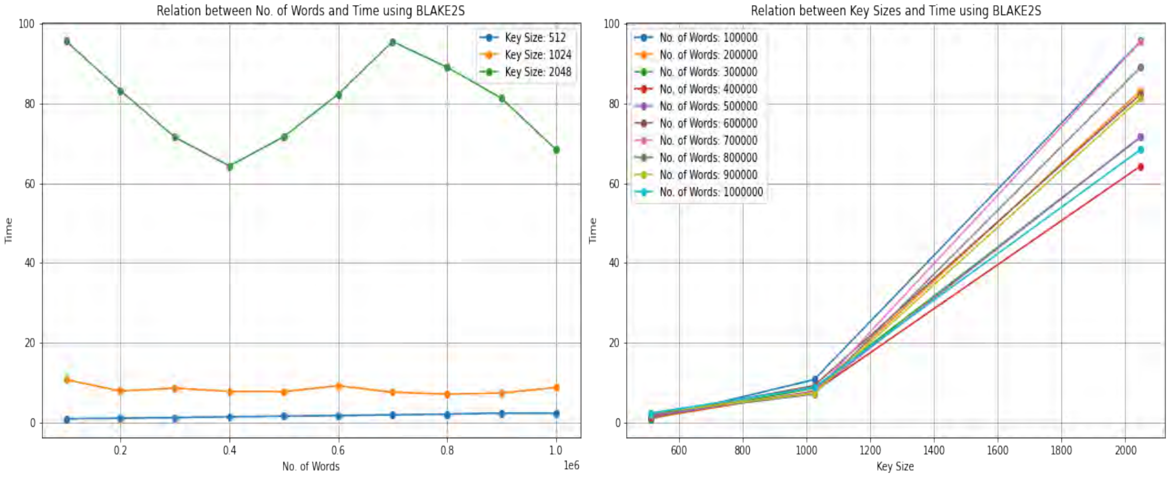

Fig. 20: BLAKE2S: Relations between the number of words and time (left), BLAKE2S: Relations between key sizes and time (right)

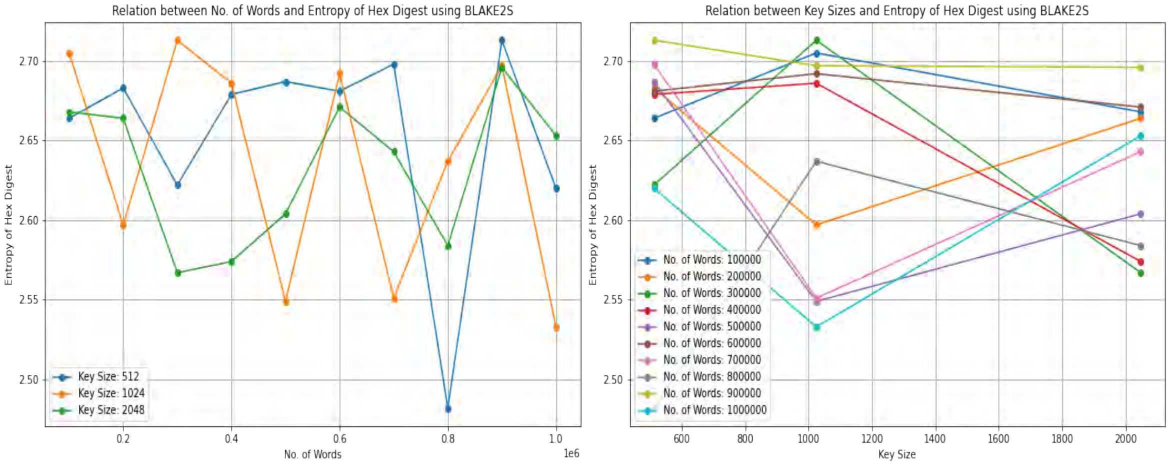

Fig. 21: BLAKE2S: Relations between the number of words and entropy of hex digest (left), BLAKE2S: Relations between key sizes and entropy of hex digest (right)

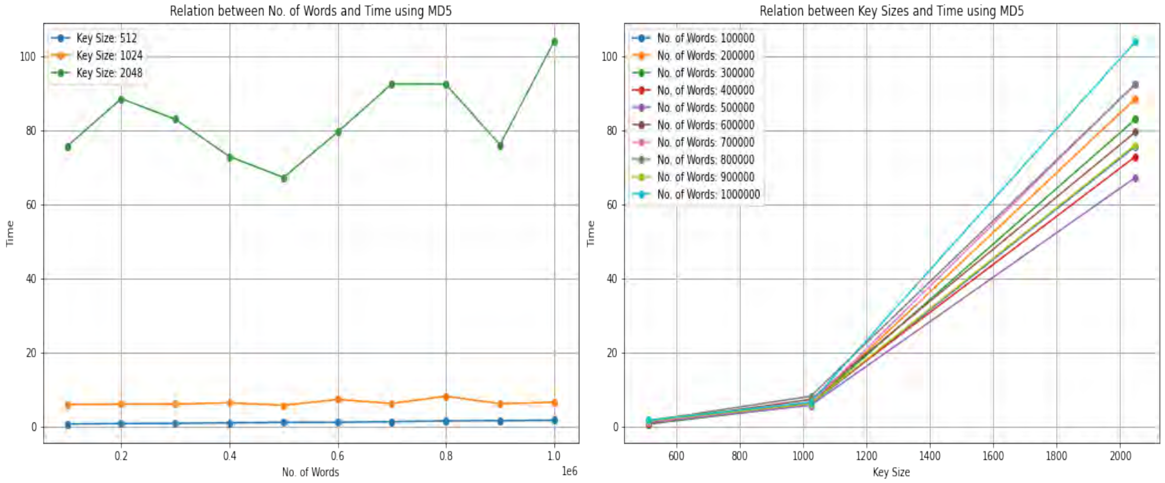

Fig. 22: MD5: Relations between the number of words and time (left), MD5: Relations between key sizes and time (right)

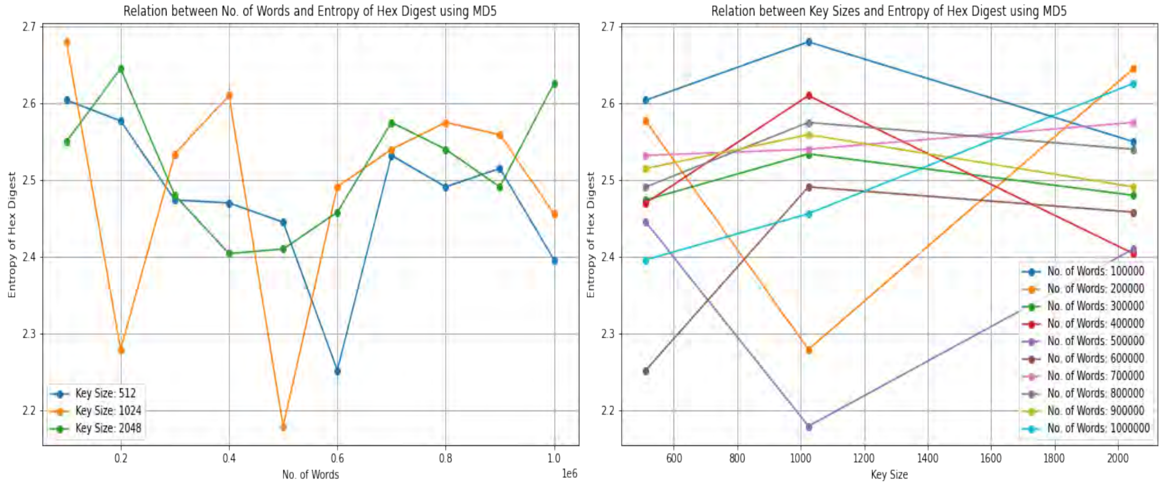

Fig. 23: MD5: Relations between the number of words and entropy of hex digest (left), MD5: Relations between key sizes and entropy of hex digest (right)

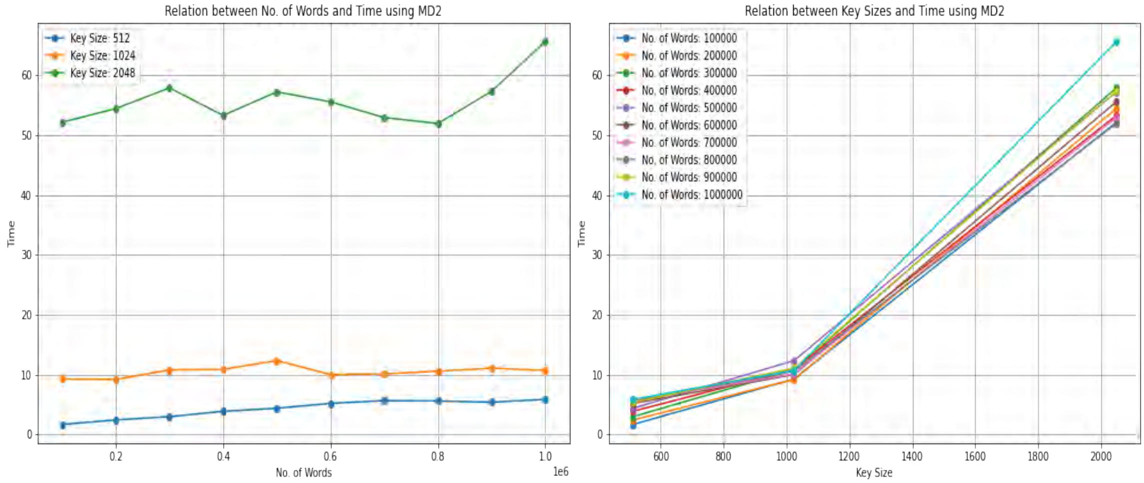

Fig. 24: MD2: Relations between the number of words and time (left), MD2: Relations between key sizes and time (right)

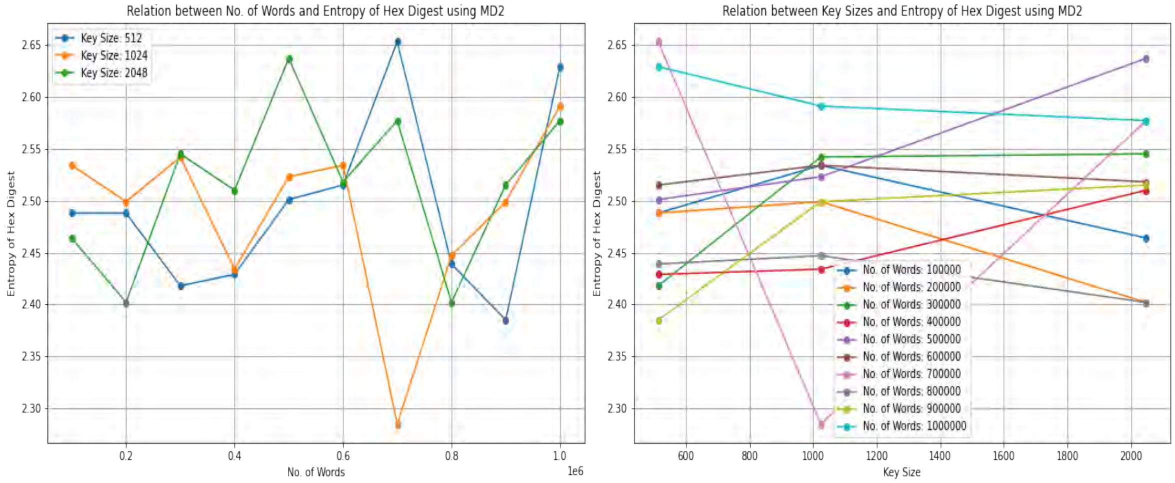

Fig. 25: MD2: Relations between the number of words and entropy of hex digest (left), MD2: Relations between key sizes and entropy of hex digest (right)

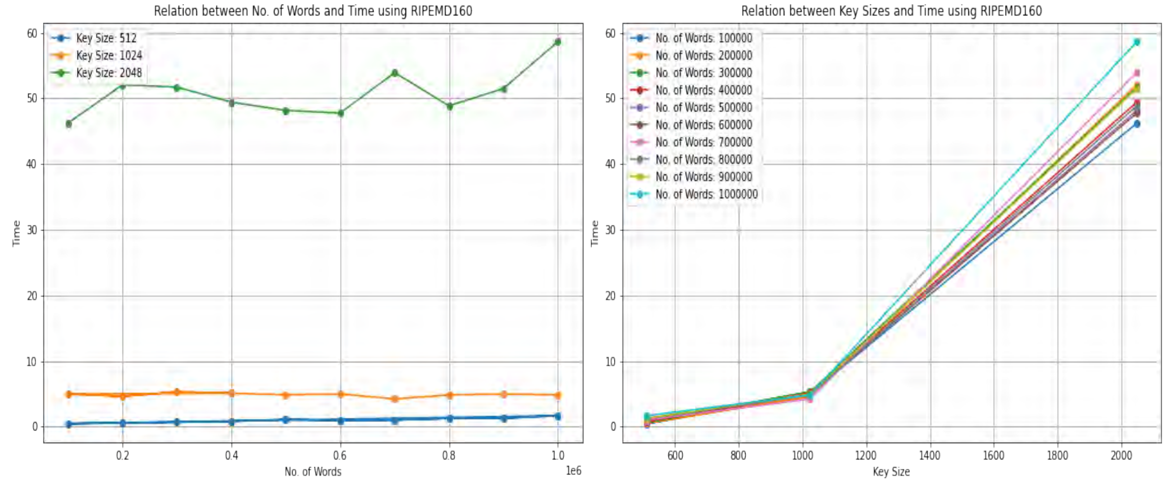

Fig. 26: RIPEMD160: Relations between the number of words and time (left), RIPEMD160: Relations between key sizes and time (right).

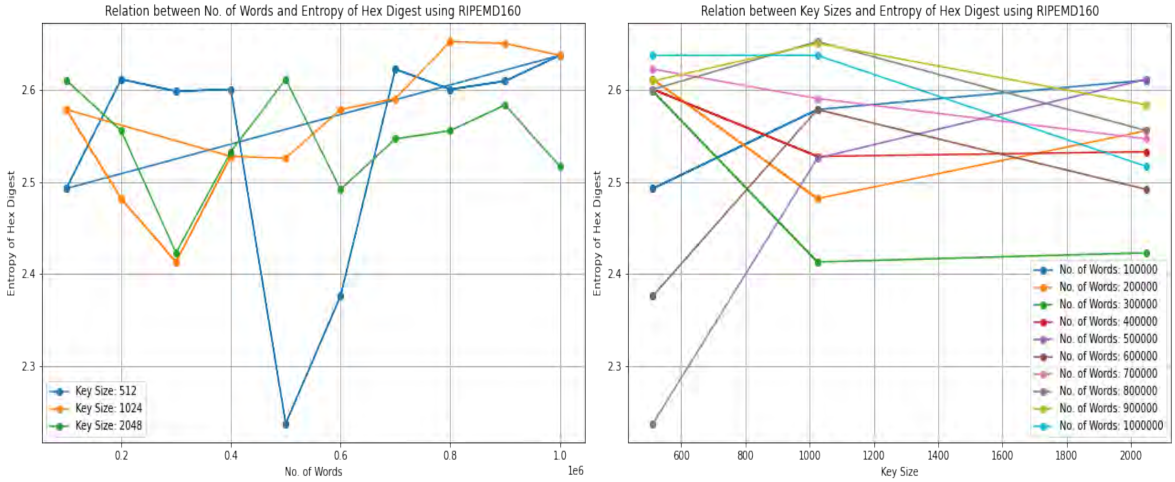

Fig. 27: RIPEMD160: Relations between the number of words and entropy of hex digest (left), RIPEMD160: Relations between key sizes and entropy of hex digest (right).

Based on the data analysis, the key size of 2048 bits displayed the highest entropy values across all the tested hash algorithms. This suggests that elevating the size of the key can contribute to enhanced security levels, even while using the same hash function. However, it is important to consider the directly proportional relationship between key sizes and processing time - as the key size increases, so does the time taken for computation. The 2048-bit key proved significantly more time-consuming in all the hash algorithms tested. Given these findings, a key size of 1024 bits offers an optimal balance between processing time and security level. Ultimately, Table 6 presents a summary of the time cost and entropy levels, reflecting the strength of the utilized hash algorithms: SHA224, SHA256, SHA384, SHA512, BLAKE2B, BLAKE2S, MD5, MD2, and RIPEMD160 with key sizes of 512, 1024, and 2048 bits.

Table 6. *Comparison of Different Hashing Algorithms over 512, 1024, and 2048 key sizes*

Index	Hash Function	Key Size	Time Cost	Entropy Level
1	SHA224	512	Low	Strong
		1024	Low	Strong
		2048	High	Strong
2	SHA256	512	Low	Strong
		1024	Low	Strong
		2048	High	Strong
3	SHA384	512	Low	Medium
		1024	Low	Medium
		2048	High	Medium
4	BLAKE2B	512	Low	Medium
		1024	Low	Medium
		2048	High	Medium
5	BLAKE2S	512	Low	Medium
		1024	Low	Medium
		2048	High	Medium
6	MD5	512	Low	Weak
		1024	Low	Weak
		2048	High	Weak
7	MD2	512	Low	Medium
		1024	Low	Medium
		2048	High	Medium
8	RIPEMD-160	512	Low	Medium
		1024	Low	Medium
		2048	High	Strong

7. Conclusions

In light of the extensive analysis conducted on five distinct hashing encryption techniques, this study yields several valuable insights that can significantly enhance the current understanding of digital signatures in the Internet of Things (IoT) context. The investigation underlines the efficacy of the Rivest–Shamir–Adleman (RSA) encryption method, based on the Miller-Rabin technique, when used with suitable key lengths. Our results demonstrate that RSA manifests superior efficiency and robust security, making it a viable candidate for deployment in IoT networks. Further, our detailed evaluation elucidates a parity in the entropy levels across five hashing functions, namely MD2, MD5, SHA, SHA-1, and RIPEMD-160, providing a uniform measure of uncertainty across these algorithms. It has also been observed that MD2 and MD5 demonstrate superior speed compared to SHA, which, in turn, outperforms RIPEMD-160 and SHA-1. These findings can inform the choice of hashing function based on specific computational speed requirements.

The results from autocorrelation and histogram analyses indicate an average percentage of randomness across the five algorithms. This characteristic is fundamental to the security of the generated digital signatures, given that greater randomness makes them more resistant to cryptographic attacks. Interestingly, this research has also revealed specific relationships within the encryption process. A direct correlation exists between the number of words and time and, similarly, between key sizes and time, offering insights into computational efficiency. Additionally, associations were found between the number of words and entropy of hex digest and between key sizes and entropy of hex digest, elucidating the role of these factors in enhancing security. In conclusion, this research gives users vital insights into various encryption algorithms, enabling them to make an informed selection based on their specific requirements. However, it also underlines the need for future work and potential modifications in these algorithms to achieve a higher entropy percentage, further bolstering security. As the field of IoT continues to evolve, such an understanding and continued advancement in security methods will be pivotal in safely realizing the immense potential of this technology.

References

[1] R. Merkle, "A certified digital signature," in Conference on the Theory and Application of Cryptology", New York: Springer, Aug. 1989, pp. 218–238. Available: https://link.springer.com/content/pdf/10.1007/0-387-34805-0_21.pdf

[2] S. Aggarwal and N. Kumar, "Digital signatures," in Advances in Computers, Elsevier BV, 2021, pp. 95–107. doi: 10.1016/bs.adcom.2020.08.004.

[3] J. K. Winn, "Couriers without luggage: negotiable instruments and digital signatures," in Routledge eBooks, 2022, pp. 245–292.

[4] Nabih, S. , fahmy H. and abdelgaber, S, " An Approach for Data Privacy Management for Banking Using Consortium Blockchain", Preprints 2022, 2022090413. https://doi.org/10.20944/preprints202209.0413.v2

[5] S. Pu and J. S. L. Lam, "The benefits of blockchain for digital certificates: A multiple case study analysis," Technology in Society, vol. 72, p. 102176, Feb. 2023, doi: 10.1016/j.techsoc.2022.102176.

[6] S. Pu and J. S. L. Lam, "The benefits of blockchain for digital certificates: A multiple case study analysis," Technology in Society, vol. 72, p. 102176, Feb. 2023, doi: 10.1016/j.techsoc.2022.102176.

[7] J. -C. Cheng, N. -Y. Lee, C. Chi and Y. -H. Chen, "Blockchain and smart contract for digital certificate," 2018 IEEE International Conference on Applied System Invention (ICASI), Chiba, Japan, 2018, pp. 1046-1051, doi: 10.1109/ICASI.2018.8394455.

[8] Rivest, R. L., Shamir, A., and Adleman, L. M., "A method for obtaining digital signatures and public key cryptosystems," In Secure communications and asymmetric cryptosystems, Routledge, 2019, pp. 217-239, doi: 10.4324/9780429305634.

[9] F. J. Aufa, Endroyono and A. Affandi, "Security System Analysis in Combination Method: RSA Encryption and Digital Signature Algorithm," 2018 4th International Conference on Science and Technology (ICST), Yogyakarta, Indonesia, 2018, pp. 1-5.

[10] R. Karim, L. S. Rumi, Md. A. Islam, A. A. Kobita, T. Tabassum, and Md. S. Hossen, "Digital signature authentication for a bank using asymmetric key cryptography algorithm and token based encryption," in Springer eBooks, 2020, pp. 853–859.

[11] M. Kathiresh, R. Neelaveni, M. A. Benny, and B. Moses, "Vehicle diagnostics over internet protocol and Over-the-Air updates," in EAI/Springer Innovations in Communication and Computing, 2021, pp. 89–100. doi: 10.1007/978-3-030-59897-6_5.

[12] N. A. El-Mawla, M. Badawy, and H. Arafat, "SECURITY AND KEY MANAGEMENT CHALLENGES OVER WSN (A SURVEY)," International Journal of Computer Science & Engineering Survey (IJCSES), vol. 10, no. 01, pp. 15–34, Feb. 2019.

[13] Md. S. Hossen, T. Tabassum, Md. A. Islam, R. Karim, L. S. Rumi, and A. A. Kobita, "Digital Signature Authentication Using Asymmetric Key Cryptography with Different Byte Number," in Springer eBooks, 2020, pp. 845–851.

[14] F. Alharbi, A. Alrawais, A. B. Rabiah, S. Richelson, and N. Abu-Ghazaleh, "CSPROP: Ciphertext and Signature Propagation Low-Overhead Public-Key cryptosystem for IoT environments," USENIX Security Symposium, pp. 609–626, Jan. 2021, [Online]. Available: https://www.usenix.org/system/files/sec21-alharbi.pdf

[15] E. A. Adeniyi, P. B. Falola, M. S. Maashi, M. Aljebreen, and S. Bharany, "Secure Sensitive Data Sharing Using RSA and ElGamal Cryptographic Algorithms with Hash Functions," Information, vol. 13, no. 10, p. 442, Sep. 2022, doi: 10.3390/info13100442.

[16] I. Nurhaida, D. Ramayanti and R. Riesaputra, "Digital Signature & Encryption Implementation for Increasing Authentication Integrity Security and Data Non-Repudiation", International Research Journal of Computer Science (IRJCS), vol. 4, no. 11, pp. 4-14, 2017, [online] Available: http://doi.org/10.26562/IRJCS.2017.NVCS10080.

[17] C. M. Nalayini, Jeevaakatiravan, P. V. Imogen and J. M. Sahana, "A Study on Digital Signature in Blockchain Technology," 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 2023, pp. 398-403.

[18] M. A. Mughal, X. Luo, A. Ullah, S. Ullah and Z. Mahmood, "A Lightweight Digital Signature Based Security Scheme for Human-Centered Internet of Things," in IEEE Access, vol. 6, pp. 31630-31643, 2018, doi: 10.1109/ACCESS.2018.2844406.

[19] M. S. Rathore et al., "A novel trust-based security and privacy model for Internet of Vehicles using encryption and steganography," Computers and Electrical Engineering, vol. 102, no. 108205, p. 108205, Sep. 2022, doi: 10.1016/j.compeleceng.2022.108205.

[20] M. Patel and R. Patel, "Improved identity based encryption system (Iibes): A mechanism for eliminating the key-escrow problem", Emerg. Sci. J., vol. 5, no. 1, pp. 77-84, 2021.

[21] M. J. Wiener, "Cryptanalysis of short RSA secret exponents," in IEEE Transactions on Information Theory, vol. 36, no. 3, pp. 553-558, May 1990, doi: 10.1109/18.54902.

[22] S. Trasatti, "Electrocatalysis: understanding the success of DSA®," Electrochimica Acta, vol. 45, no. 15–16, pp. 2377–2385, May 2000, doi: 10.1016/s0013-4686(00)00338-8.

[23] J. Lopez and R. Dahab, "An Overview of Elliptic Curve Cryptography", technical report Inst. of Computing State Univ. of Campinas, May 2000. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.37.2771&rep=rep1&type=pdf

[24] D. Johnson, A. Menezes, and S. A. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," International Journal of Information Security, vol. 1, no. 1, pp. 36–63, Aug. 2001, doi: 10.1007/s102070100002.

[25] I. Liusvaara and S. Josefsson, “Edwards-Curve Digital Signature Algorithm (EDDSA),” Jan. 2017. doi: 10.17487/rfc8032.

[26] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in IEEE Transactions on Information Theory, vol. 31, no. 4, pp. 469-472, July 1985, doi: 10.1109/TIT.1985.1057074.

[27] C.-P. Schnorr, "Efficient signature generation by smart cards," Journal of Cryptology, vol. 4, no. 3, pp. 161–174, Jan. 1991, doi: 10.1007/bf00196725.

[28] K. S. McCurley, "The discrete logarithm problem", Proc. Symp. Appl. Math., vol. 42, pp. 49-74, 1990.

[29] J. H. Silverman and J. Suzuki, "Elliptic curve discrete logarithms and the index calculus," in Lecture Notes in Computer Science, 1998, pp. 110–125. doi: 10.1007/3-540-49649-1_10.

[30] S. Y. Yan, Cybercryptography: Applicable cryptography for cyberspace security. 2019. doi: 10.1007/978-3-319-72536-9.

[31] S. K. Sood and Pooja, "Quantum Computing Review: A Decade of Research," in IEEE Transactions on Engineering Management, doi: 10.1109/TEM.2023.3284689.

[32] L. L. G. Bruinderink, A. Hülsing, T. Lange, and Y. Yarom, "Flush, Gauss, and Reload – A Cache Attack on the BLISS Lattice-Based Signature Scheme," in Lecture Notes in Computer Science, 2016, pp. 323–345. doi: 10.1007/978-3-662-53140-2_16.

[33] A. Huelsing, D. Butin, S.-L. Gazdag, J. Rijneveld, and A. Mohaisen, “XMSS: EXtended Merkle Signature Scheme,” May 2018. doi: 10.17487/rfc8391.

[34] M. T. Kumar, R. K. Katragadda, V. S. Kolli and S. L. Rahiman, "A Hybrid Approach for Enhancing Security in Internet of Things (IoT)," 2019 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 2019, pp. 110-114.

[35] P. Semwal and M. K. Sharma, "Comparative study of different cryptographic algorithms for data security in cloud computing," 2017 3rd International Conference on Advances in Computing,Communication & Automation (ICACCA) (Fall), Dehradun, India, 2017, pp. 1-7, doi: 10.1109/ICACCAF.2017.8344738.

[36] F. Mallouli, A. Hellal, N. Sharief Saeed and F. Abdulraheem Alzahrani, "A Survey on Cryptography: Comparative Study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms," 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/ 2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Paris, France, 2019, pp. 173-176, doi: 10.1109/CSCloud/EdgeCom.2019.00022.

[37] F. J. Aufa, Endroyono and A. Affandi, "Security System Analysis in Combination Method: RSA Encryption and Digital Signature Algorithm," 2018 4th International Conference on Science and Technology (ICST), Yogyakarta, Indonesia, 2018, pp. 1-5.

[38] S. Long, "A comparative analysis of the application of hashing encryption algorithms for MD5, SHA-1, and SHA-512," Journal of Physics, vol. 1314, no. 1, p. 012210, Oct. 2019, doi: 10.1088/1742-6596/1314/1/012210.

[39] P. P. Pittalia, "A Comparative Study of Hash Algorithms in Cryptography", Int. J. Comput. Sci. Mobile Comput., vol. 8, no. 6, pp. 147-152, Jun. 2019.

[40] G. Tang, B. Pang, L. Chen and Z. Zhang, "Efficient Lattice-Based Threshold Signatures With Functional Interchangeability," in IEEE Transactions on Information Forensics and Security, vol. 18, pp. 4173-4187, 2023, doi: 10.1109/TIFS.2023.3293408.

[41] X. Feng, J. Ma, H. Wang, Y. Miao, X. Liu and Z. Jiang, "An Accessional Signature Scheme With Unmalleable Transaction Implementation to Securely Redeem Cryptocurrencies," in IEEE Transactions on Information Forensics and Security, vol. 18, pp. 4144-4156, 2023, doi: 10.1109/TIFS.2023.3293402.

[42] M. R. Alagheband and A. Mashatan, "Advanced digital signatures for preserving privacy and trust management in hierarchical heterogeneous IoT: Taxonomy, capabilities, and objectives," Internet of Things, vol. 18, p. 100492, May 2022.

[43] P. Gupta, A. Sinha, P. Srivastava, A. Perti, and A. K. Singh, "Security implementations in IoT using Digital Signature," in Lecture notes in electrical engineering, 2020, pp. 523–535. doi: 10.1007/978-981-15-4692-1_40.

[44] A. M. Adeshina, "Evaluation of Elliptic Curve El-Gamal and RSA Public-Key Cryptosystems for Digital Signature", vol. 4, no. 1, pp. 36-49, 2020.

[45] E. Gradini, “Comparison Study Of Fermat, Solovay-Strassen And Miller-Rabin Primality Test Using Mathematica 6.0,” Visipena: Jurnal Komunikasi Pendidikan, Jun. 2012, doi: 10.46244/visipena.v3i1.48.

[46] E. Simion, "Entropy and Randomness: From Analogic to Quantum World," in IEEE Access, vol. 8, pp. 74553-74561, 2020.

[47] A. Sarkar, J. Dey and A. Bhowmik, "Multilayer neural network synchronized secured session key based encryption in wireless communication", Indonesian J. Electr. Eng. Comput. Sci., vol. 14, no. 1, pp. 169, Apr. 2019.

[48] B. Militzer, M. Zamparelli and D. Beule, "Evolutionary search for low autocorrelated binary sequences," in IEEE Transactions on Evolutionary Computation, vol. 2, no. 1, pp. 34-39, April 1998, doi: 10.1109/4235.728212.

[49] A. D. Riad et al., "Security Evaluation and Encryption Efficiency Analysis of Rc4 Stream Cipher for Converged Network Applications", Journal of Electrical Engineering, vol. 64, no. 3, 2013.

[50] R. S. Salman, A. K. Farhan and A. Shakir, "Lightweight Modifications in the Advanced Encryption Standard (AES) for IoT Applications: A Comparative Survey," 2022 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, 2022, pp. 325-330, doi: 10.1109/CSASE51777.2022.9759828.