Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Modify the Speck algorithm-SHA3 to enhance data integrity and authentication in WoT environments

Baraa Mohammed Hasan [1] *, ***Samar Kareem Tuama*** [2]

[1,2] *Computer Department, College of Education for Pure Sciences, Wasit University, Al-Kut, Wasit, Iraq.*

[1]*Email: bhassan@uowasit.edu.iq*

[2] *Email: stohma@uowasit.edu.iq*

ARTICLE INFO

Article history:
Received: 09 /02/2026
Rrevised form: 05 /04/2026
Accepted : 06 /04/2026
Available online: 30 /06/2026

Keywords:

SPECK algorithm, SHA-3, hybrid encryption, WoT integrity, WoT authentication

ABSTRACT

The rapid evolution of the Web of Things (WoT) demands cryptography solutions to ensure a high level of security with minimal usage and to work efficiently. In large scale WoT environments, standard cryptographic primaries can be of great energy and computational cost. The following paper is a proposal of a hybrid cryptographic system that will combine the lightweight SPECK block cipher and the Secure Hash Algorithm-3 (SHA-3) to provide improved data integrity, authentication, and efficiency in operations. The proposed hybrid cryptography can reduce the complexity of computing and still have a high level of security since it adds SPECK to the SHA-3 sponge design. Experimental results show significant gains in execution time and throughput compared to SHA3-512, especially for large data sets. Furthermore, NIST Statistical Test Suite findings show that the proposed hybrid approach has randomness properties comparable to SHA3-512, with all p-values exceeding the acceptable significance threshold. These findings show that the suggested hybrid cryptography is a scalable and efficient security solution for resource-constrained WoT.

https://doi.org/10.29304/jqcsm.2026.18.22637

1. Introduction

With the increased integration of sensors, actuators, embedded devices, and smart electrical appliances into the Internet, the World Wide Web (WWW) emerges as a natural framework for intelligent networking. The rapid growth of the WoT has enabled the seamless integration of physical devices with web technologies, allowing smart things to be accessed, monitored, and controlled via ordinary web protocols, as illustrated in Fig. (1). Despite their benefits, WoT settings are fundamentally limited in terms of compute power, memory capacity, and energy consumption, making the adoption of typical cryptographic techniques difficult. Consequently, there is a need to have light and effective security tools that would ensure data integrity, authentication, and confidentiality without undue heavy load on the devices with limited resources.

∗Corresponding author

Email addresses: *bhassan@uowasit.edu.iq*

Cryptography plays a fundamental role in the achievement of security and reliability of distributed digital ecosystems. Its four essential objectives, including confidentiality, integrity, authentication, and non-repudiation play an important role in protecting data flow and system reliability. Cryptographic tools play a very important role in reducing security issues in an environment that is becoming dramatically more and more difficult to control and manage, where the environment is more and more varied and distributed. The adequate choice of cryptographic primitives will thus have a direct impact on the general hardiness and realism of the system [1].

One of these primitives is cryptographic hash functions, which are necessary in offering authentication techniques and ensuring data integrity [2]. They help recipient devices to ascertain that the information transmitted in the process has not been altered. Nevertheless, a lot of traditional hash algorithms do not scale to low resources platforms since they are expensive both in computation and memory. SHA-3 in this respect has turned out to be an effective and versatile alternative. The sponge-based SHA-3 is a design with flexible security settings and lightweight code executions, as well as strongly resistant to known cryptanalytic attacks [3][4]. Owing to these attributes, it is particularly applicable in the safe processing of data and integrity checks in small environments where high security guarantees have to co-exist with efficiency [5].

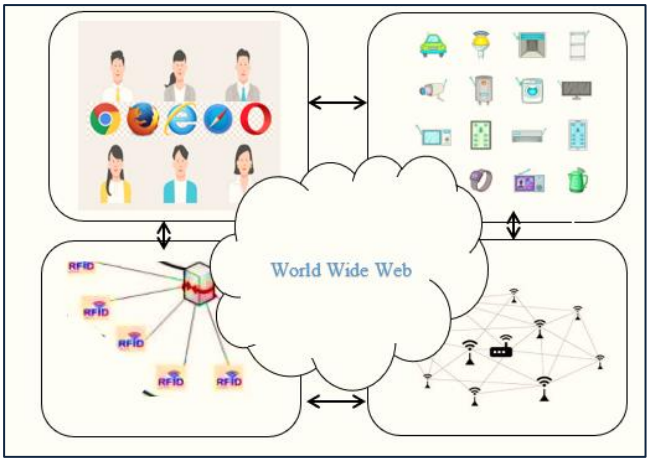

Fig.1- Overview of WoT Environment.

Along with ensuring data integrity, encryption is one of the basic cryptographic tools for protecting the privacy of sensitive information exchanged between devices in the WoT [6]. Lightweight encryption algorithms are very relevant in this sector due to the limitations of computing resources (including power expenditure). One such algorithm is Speck, which was designed for limited hardware settings and has attracted interest from numerous researchers due to its ease of design, size, and ability to work well on both software and hardware [7]. The advantages that Speck possesses are incredibly effective when it comes to the nature of real-time communications that are prevalent in most WoT solutions, since Speck is low latency, as well as energy efficient. Speck can help transmit data with encryption security and efficiency, greatly improving the security, reliability, and practicality of a WiT infrastructure [8][9][10].

In this paper, we suggest using a combination of Speck and Hash Algorithm-3, where one can ensure the safety of their data and at the same time pretend the integrity of their data; the combination is best applicable in large-scale, highly heterogeneous, and resource-constrained environments. A balanced security framework can be achieved for WoT environments.

The structure of this paper consists of the following sections: Section 2 will provide a review/analysis of the literature related to encryption; Section 3 will provide an overview of the architecture used for encryption (and also provide an overview of how each component works); Section 4 will explain in detail what this new encryption approach looks like, including why the authors created it and how it will function; Section 5 discusses the results of implementing this new approach and evaluates how well it has performed in terms of both security and performance. Finally, Section 6 summarises the key findings and outlines prospective research areas.

2. Related Work

SPEAK and SHA-3 have been discussed separately across several studies. The authors, Yi Yang et al., introduced a compact FPGA-based SHA-3 architecture utilizing RAM and FSM control to minimize area. The main contribution was achieving an area reduction of up to 74.7% compared to prior designs. Results confirm suitability for WBSN environments. However, the study lacks a lower throughput compared to high-speed designs; future work includes power optimization [11].

The authors, Young Beom Kim et al., proposed a software chaining optimization methodology that merges θ, ρ, and π steps to reduce memory access. The key contribution was a 26.1% performance improvement on 8-bit AVR microcontrollers. Results demonstrated reduced execution time and better suitability for IoT. However, the study lacks platform dependency; future work targets broader MCU and post-quantum applications [12].

The authors, Jeethu James et al., presented a Verilog-based SHA-3-256 IP core implemented on a Virtex-6 FPGA. The contribution was a complete and reliable hardware characterisation of SHA-3-256. Results demonstrated correct functionality and stable performance. This study lacked low-area and low-power enhancements [13].

The authors, Brian Baldwin et al., proposed a standardised hardware wrapper methodology to fairly evaluate SHA-3 candidates across FPGA/ASIC platforms. The main contribution is integrating communication and padding into a unified interface. Results showed more accurate comparisons of area, timing, and power [14]. The authors, Elena Andreeva et al., adopted a survey and analytical methodology to study the security foundations of cryptographic hash functions, particularly focusing on modular designs based on compression functions, block ciphers, and permutations. The authors identify unresolved theoretical gaps in collision resistance, indifferentiability, and security reductions. The main result was a structured set of open research problems that highlight limitations in current security proofs. A key limitation was that the work was theoretical in nature and did not propose concrete new hash constructions or implementations [15].

The authors Ray Beaulieu et al. proposed two families of lightweight block ciphers designed to operate efficiently across both hardware and software platforms. The methodology was based on the simple round functions and the ability to parameterize it to provide various block and key sizes. It has been demonstrated by performance analysis that SIMON and SPECK are in the midst of numerous current lightweight ciphers based on area, memory consumption, and throughput [16].

His proposal, Speck-R, an improved light-weight cipher, using the SPECK algorithm, by the authors Lama Sleem and Raphael Couturier added a dynamic key-dependent substitution layer to decrease the number of rounds. The authors conducted the scheme on actual IoT hardware and security and randomness tests. The findings demonstrated up to a 77 percent decrease in the execution time and resistance to both differential and linear attacks, relative to the original SPECK. The new approach, however, adds a dynamic layer to the design, and the extra cryptanalytic efforts by third parties are needed to confirm long-term security [17]. Emmanuel Agullo et al. employed an analytic, systematic, and survey-based approach to investigate fault tolerance in exascale computer systems. The authors analyzed the resilience techniques, which encompassed algorithm-based fault tolerance, error-aware algorithms, and redundancy mechanisms. The findings demonstrated that conventional checkpoint-restart techniques are not scalable for exascale systems, necessitating algorithm-level resilience. A limitation was that the paper was largely conceptual and did not provide concrete implementations or quantitative performance evaluations [18].

As well as Abdullah Sevin and Ünal Çavu¸so˘glu, presented a lightweight hash function constructed using the SPECK block cipher to improve efficiency in IoT environments. The methodology integrates SPECK's ARX structure to achieve strong diffusion and confusion with reduced computational cost. Experimental results demonstrated improved execution time and successful resistance to collision and pre-image attacks compared to traditional hash functions. However, the smaller output size may limit collision resistance for high-security applications, indicating a trade-off between efficiency and security [19].

The authors, Alex Biryukov et al., applied an automatic differential trail search methodology to analyse the security of SIMON and SPECK against differential cryptanalysis. The authors present improved differential trails and key-recovery attacks on reduced-round versions. The findings revealed that full-round cases are not under attack, whereas the reduced-round ones are prone to sophisticated attacks. One weakness was that the attacks couldn't be scaled to full-round implementations, and their practical impact was modest [20].

This paper examined quantum attacks on SPECK based on the Grover algorithm and quantum differential cryptanalysis. The authors construct reversible quantum circuits and optimize resource usage in terms of qubits and gate complexity, which proved to be IBM-Q-checked. It was found that the SPECK abundance of quantum-based security is particularly lower. But the experiment presupposes perfect quantum hardware, and quantum constraints make it impossible to do such attacks in practice at present [21].

Last, the reviewed literature confirms that although SHA-3 and SPECK-based primitives can be realised with large efficiency gains to constrained environments, these additional gains are frequently at the cost of area, power, or security margins. This gives rise to a research gap on the holistic cryptographic design that meets the goal of maximizing performance, lightweight, and resilience against classical and quantum attackers.

3. Background

3.1 Speck algorithm

The Speck cipher is a block cipher which is light weight and ensures that it supports a high variety of block and key sizes with a high degree of flexibility in their cryptographic implementations. Even though numerous lightweight block ciphers were suggested, the majority of the lightweight block ciphers are well tuned to a single platform and not well implemented in a wide range of hardware and software platforms [7][22]. In June 2013, the U.S. National Security Agency (NSA) introduced Speck as part of the Simon and Speck family of lightweight encrypted algorithms to provide efficient and secure encryption of the software and hardware platform [16]. Software implementations have had excellent performance by Speck, especially with regards to memory efficiency and less code size whereas its counterpart, the Simon algorithm tends to be more efficient in terms of hardware-implementations. The Speck algorithm is able to be effectively installed on a wide array of platforms and devices due to its flexible design and, thus, it can successfully apply on lightweight devices like embedded systems or Internet of Things (IoT) systems [23].

The Speck cipher takes a variety of block and key sizes with each block being two words, 16, 24, 32, 48 or 64 bits, and the key size is two, three, or four words. Its round operation makes use of circular rotations, modular addition and exclusive-OR (XOR) operations, where the count of the rounds is decided by the set of chosen parameters [9].

The SPECK cipher employs three fundamental operations on n-bit words in each encryption round: bitwise exclusive-OR (XOR) $\oplus$, addition modulo 2^n, and circular left and right shifts by r_2 and r_1 bits, respectively. In the r-th round, the left and right n-bit input words are denoted as $X_{r-1,L}$ and $X_{r-1,R}$, while the corresponding round key is represented by k_r. The output words of the round, $X_{r,L}$ and $X_{r,R}$, are computed by applying these operations sequentially to the input words and the round key as in Fig2 and Equations (1) and (2):

$$X_{r,L} = \left(\left(X_{r-1,L} \gg r_1 \right) \boxplus X_{r-1,L} \oplus k_r \right) \quad (1)$$

$$X_{r,R} = \left(\left(X_{r-1,R} \ll r_2 \right) \oplus X_{r,L} \right) \quad (2)$$

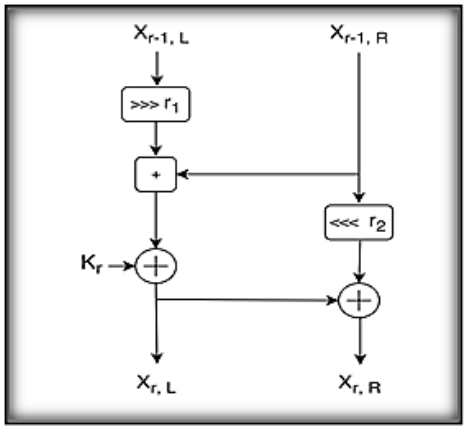

Fig.2-The SPECK round function[24].

Various key sizes are employed across different instances of the SPECK family, and the total number of encryption rounds is determined by the selected key size. The rotation constants $r_1$and $r_2$are defined as $r_1 = 7$and $r_2 = 2$for SPECK32, while for all other variants they are set to $r_1 = 8$and $r_2 = 3$. The parameters corresponding to all SPECK variants are summarized in Table 1[25].

Table.1 The Different Sizes of Blocks and Keys in Speck Cipher[25].

Block Size	Key Size	Rounds
Speck32	64	22
Speck48	72	22
	96	23
Speck64	96	26
	128	27
Speck96	96	28
	144	29
Speck128	128	32
	192	33
	256	34

3.2 Secure Hash Algorithm -3(SHA3)

In 2007, the National Institute of Standards and Technology (NIST) launched an open international cryptographic competition aimed at the development of a next-generation hash function, later standardized as SHA-3. This initiative was motivated by the need to enhance the robustness and diversity of cryptographic hash algorithms within the Federal Information Processing Standards (FIPS) framework, particularly in light of growing cryptanalytic advances targeting existing standards[26].

Under the first submission phase in 2008, NIST was able to receive more than sixty candidate algorithms of researchers all around the world. Fourteen proposals that passed through a thorough evaluation process were chosen in order to proceed to the second round in 2009. In 2010, five algorithms-BLAKE, Grostl, JH, Keccak and Skein- became finalists, after successful selection based on the strength of the algorithms, efficiency of their execution and the flexibility of their implementation. These contestants were then taken through a lengthy trial period of scrutiny and a technical analysis of about one and a half years in the hands of the people. After this intensive evaluation, in 2012 Keccak was finalized as the winning algorithm, owing to its innovative sponge-based

implementation, large security margins and good performance on both hardware and software platforms thus becoming the SHA-3 cryptographic hash standard [27]. Among the major factors that have led to the choice of Keccak as the SHA-3 standard:

Keccak is resistant to cryptanalytic attacks. Numerous security analyses revealed that only reduced-round versions of Keccak were vulnerable to finite types of cryptanalyses like near-collision attacks which maxed out at beauty in no more than five of the twenty-four rounds. Noteworthy, these attacks never managed to go past the initial rounds, which means that Keccak has a large security margin since the sixth round. This large gap between the best-known attacks and the complete round implementation is a solid guarantee of Keccak long-term security and resistance to future attacks on cryptanalyses, so justifying its use as a highly reliable and resilient cryptography hash algorithm.

Keccak has a hardware-oriented architecture with a focus on basic operations at the bit-level, which is radically unlike the architecture used by the SHA-2 family. The design selection facilitates easy parallelism and adaptability in hardware and software implementation, as well as helping in the achievement of better performance and architectural variety in hash function design in cryptography.

The Keccak displays efficient performance on both software and hardware implementations which is an indication of its sound design and comparatively good capability to ensure data security with regard to other hash algorithms.

Keccak is founded on a new hashing model the sponge construction that introduces a flexible and efficient implementation of data processing. This method works over a large internal state and has two major stages, absorption where input bits are added to the internal state and squeezing where output bits are produced. The obtained output is a pseudo-random function of all the previously received inputs, which allows good diffusion and high security. This design provides Keccak with a great level of flexibility, allowing them to produce variable length outputs and provides a broad portfolio of cryptographic use [28].

As it was stated earlier, the SHA-3 algorithm is based on the sponge construction which is a flexible cryptographic model that does not pass input data through the rigid transformation chains but rather applies the input data to the two processing stages: absorption and squeezing. During the absorption stage, the input message is first divided into fixed-size blocks which are repeated with a specific amount of internal state, and then a nonlinear permutation function f is used. This is done so that diffusion of input data within the state is complete. Under the squeeze stage, a block of output is produced by removing data in the writable part of the state, termed as the rate (R) and the rest part also called capacity (C) is obscured and acts as the main source of cryptographic security. This distance between rate and capacity is quite effective in countering cryptanalytic attacks. This working scheme (refer to Fig 3) determines the basic mode of working of the SHA-3 algorithm [12].

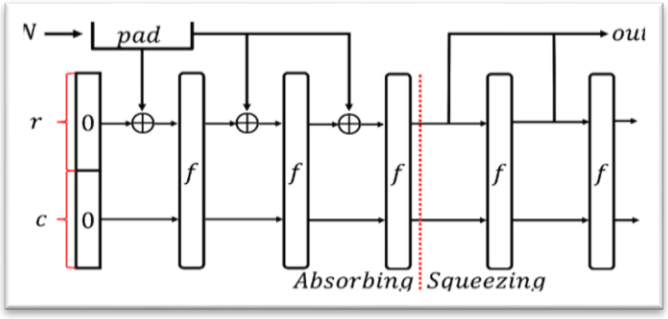

Fig.3-The sponge function's structure[12].

4. Speck algorithm -SHA3 for data integrity and authentication

As already noted, SHA-3 has natural processing inefficiencies in its use, as the suggested algorithm cannot do the processing of a fragmentation of constant length required to be efficient in its use. This can greatly hamper the execution of WoT environments, especially in application where data validation is done with the use of SHA-3. To overcome this obstacle, this research proposes a new hybrid cryptographic architecture that combines SPECK algorithm, well-known with its lightweight architecture, high performance, and excellent security assurance, and the SHA-3. The intended integration will improve the integrity of the data, curb the illegitimate manipulation of the data, and offer quality authentication schemes.

In the suggested design, the SPECK cipher is implemented into the sponge implementation of SHA-3 as a supplementary security measure and is set to use nine rounds to minimize computation costs. This hybrid architecture is also expected to be much more efficient overall system-wide and maintain the degree of cryptographic security in a secure operation of the resource-constrained WoT, as evidenced in Fig.4, demonstrating the hybrid approach to Speck and SHA-3.

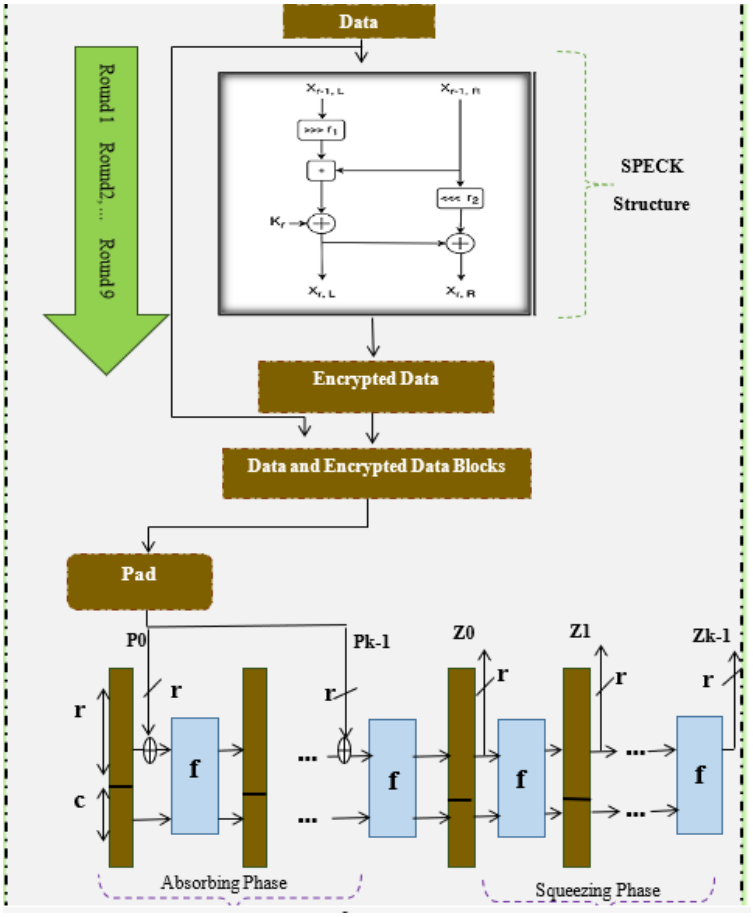

Fig .4-Block Diagram of the Proposed Hybrid Method.

5. Results and discussion

The proposed hybrid solution, a combination of Speck block cipher and the SHA-3 hash algorithm, is tested in this paper to analyze the solution through a series of thorough experimental procedures. After deriving the values of the hybrid method, a critical analysis and statistical analysis of the results produced were performed, and the findings are presented in Table 2. In addition, the NIST statistical test suite was used to test the dataset in order to establish

that it is random, secure, and dependable. Table 3 elaborates the results. All in all, these analyses provide a satisfactory level of empirical evidence, which shows that the proposed hybrid approach is robust, effective, and valid. Table 4 shows the test results obtained after changing a single letter in the input text and compares the outputs provided by the proposed hybrid and the original SHA3-512 algorithms.

Table 2 presents a performance comparison between the proposed hybrid and SHA3-512 algorithms in terms of execution time and throughput across different data sizes. The results show that the proposed hybrid consistently requires less processing time than SHA3-512 for all input sizes. Additionally, the proposed hybrid achieves higher throughput values, particularly for large datasets, where it reaches approximately 96–97 bytes/msec compared to about 73 bytes/msec for SHA3-512. These findings point to the fact that the proposed hybrid is less prone to scalability and offers better performance in terms of computational power and, therefore, more appropriate for the processing of high-speed hashing.

Table 2. A performance comparison to the proposed hybrid—512 and SHA3-512.

Size of Data (bytes)	proposed hybrid -512 (msec) 9 rounds	Throughput of proposed hybrid -512 (bytes/ msec)	SHA3-512 (msec) 24 rounds	Throughput of SHA3-512 (bytes/ msec)
10	0.2652	37.7074	1.0296	9.7125
25	0.5148	48.5625	1.0408	24.01910
70	1.0140	69.0335	1.0764	65.0316
100	1.2636	79.13810	2.1372	46.7902
1000	10.3740	96.3948	14.0400	71.2251
2000	20.6544	96.8317	28.0020	71.4235
10000	102.7418	97.3314	137.4050	72.7776
500000	5310.4311	94.1543	6848.9268	73.0041
1000000	10349.7758	96.6205	13698.4777	73.0008

As Table 3 shows, the proposed hybrid and SHA3-512 both pass the NIST Statistical Test Suite, and all p-values are larger than the value of (alpha = 0.01). The results are largely centered around 9, indicating balanced bit distribution and strong randomness. The proposed hybrid exhibits stable performance across all tests, while SHA3-512 shows a lower p-value in the overlapping template matching test, though it remains within acceptable limits. Overall, the proposed hybrid demonstrates randomness properties comparable to the standard SHA3-512 algorithm.

Table 3. Results of proposed hybrid -512 in NIST Test.

Names of NIST Statistical Tests Results	proposed hybrid -512	SHA3-512
Monobit Test	0.5025	0.4995
Frequency Within Block Test	0.5012	0.4940
Runs Test	0.5009	0.4970
Longest Run Ones in a Block Test	0.4899	0.4925
Binary Matrix Rank Test	0.4821	0.53910
Discrete Fourier Transform (DFT) Test	0.4783	0.4812
Non-Overlapping Template Matching Test	0.4521	0.4457
Overlapping Template Matching Test	0.5458	0.0147
Maurers Universal Test	0.9987	0.9987
Linear Complexity Test	0.4607	0.4084
Serial Test	0.5043	0.5007

Approximate Entropy Test	0.5009	0.4968
Cumulative Sums Test	0.5220	0.5144
Random Excursion Test	0.72410	0.7041
Random Excursion Variant Test	0.7151	0.7240

Table 4 presents example hashing results for plaintext with a 512-bit block size. The proposed hybrid 512 demonstrates improved randomness and security when processing larger files, as increased data size and heterogeneity enhance the randomness of the generated hash values. Unlike SHA-3, which relies on masking techniques, the algorithm primarily depends on input size, making it well-suited for large-scale data processing. Its main objective is to ensure reliable and secure hashing for huge files generated by multiple entities within a WoT environment.

Table 4. Examples of hashing result in a block size 512.

Clear Text	proposed hybrid -512	SHA3-512
00000000000000000	7a6b47e70a9053dbfa5c9227426a96b15a6f48eb4e5f710968b6cc70cad046ba8741ee4e845fb335712973746cc4faffa1e48a83455b823b38d85ac6547c5438	792dba3aeeb8e10aeec0f08db56516ab9ffb1c248b03d1f50d7ac562a21d0d7c6fadd0a179fc891710677ebf4b4b79a4ea102db295e787ebc947f7369e84e8e5
0123456789ABCDEF	8e35c7148001f3443d8ce42fccd8101b790ac9932510dfec77bf27b98c58b212cb33aaaeec06ec98a6b929c5c6db32bc638998bc6394057e8ce74acd1eddda33	97ff3b66641a5a87e90d2d015d9f033db897b629af7840bfe1c6d751f3a2e8d60ae5b03b62813fee5a86ce633f24c4730ba35044d75188249d794cd6dbf2b630
FFFFFFFFFFFFFFFF	fd79b2f7567abf5853998e7ff83cdf191aa86affc262db2187f0d80a0e3084f049a14080940c5871b024e847307ed050aaf6af0eee451df7f1c07a923ab76a7f	bfa195edb2841a82a15275d9c68f8c59916c1e77d885d6648fb74672cca84718c1f66a3f1a8f7346d17ac6f5158a2f22cdee9a3726d451570f2d8ffd097f078a
1111111111111111	f86d23a1b77c981606fb8fa16108e2859ce1f85abae3f80d15bec40da2dda0dd86584be899afd8002db77d6b553eb169dd752e18ddedfd5a94e07cc8b311b363	e26d6f6dcc69ccaee7be74e18079337e0ceef90b1e4f68728b433002c172975db24cc6a2ffe7ca3bc540997a1c7c806945ed33a39ba16e91cd0e830db7508d1b
1010101010101010	bf2993d31521dfb61b44e9a3eee4ad2f70e4354e158d0d332ecf26c4ade9fe9653808fed914d50bc676f81c8651ffb72c2b3392dc0aabd45d05ab99f68e999d0	2b3a3b778370a01997717d6da1d000449cd1bf3d906a22dedb3fc078b0be16f82b87e579c1d15919e1390481a5bda2544d23f537791cab3932df7b30e8177614
0000000011111111	37a99008eda7e1552b5dcd2b1b9eca8b5574051eca23c586bcf21d6b5d4daedb5ead8e320691b53f7a2324955e7b3775d27f76810103f5a306426a62b9e98530	b910749c3b32ab699bda7c61f5ae01edb26db7a42994b3b78e607d605f93bdea5691da55c4c097079faee286b905085b2f4da34ba5288ad97668aedbdb37ae5c

6. Conclusion

Based on a rigorous analysis of the experimental results obtained from the evaluation of the proposed hybrid، a set of conclusions can be drawn regarding the suitability of the applied modifications for the intended operational environment. The paper introduced a hybrid approach that integrates the lightweight SPECK block cipher with the

SHA-3 hashing algorithm, achieving an effective balance between security and computational efficiency. Experimental results demonstrate improved execution time and throughput compared to SHA3-512, particularly for large data sizes, while NIST statistical tests confirm its reliability for data integrity and authentication in WoT environments. Future work will focus on optimization for ultra-low-power devices and evaluation against advanced cryptanalytic and quantum-based attacks.

References

[1] A. Shemshadi, Q. Z. Sheng, and Y. Qin, “The Anatomy of An Intent Based Search and Crawler Engine for the Web of Things,” in Managing the Web of Things: Linking the Real World to the Web, Elsevier Inc., 2017, pp. 37–72. doi: 10.1016/B978-0-12-809764-9.00003-2.

[2] R. Sobti and G. Ganesan, “Cryptographic Hash Functions: A Review,” International Journal of Computer Science Issues (IJCSI, Vol. 9, Issue 2, No 2, March 2012. Available: https://www.researchgate.net/publication/267422045

[3] Dworkin, Morris J, “SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions,” 2015, nist.gov. doi: 10.6028/NIST.FIPS.202.

[4] S. El Moumni, M. Fettach, and A. Tragha, “High throughput implementation of SHA3 hash algorithm on field programmable gate array (FPGA),” Microelectronics J, vol. 93, Nov. 2019, doi: 10.1016/j.mejo.2019.104615.

[5] M. Rao, T. Newe, I. Grout, and A. Mathur, “High Speed Implementation of a SHA-3 Core on Virtex-5 and Virtex-6 FPGAs,” Journal of Circuits, Systems and Computers, vol. 25, no. 7, Jul. 2016, doi: 10.1142/S0218126616500699.

[6] R. Sardar et al., “Challenges in detecting security threats in WoT: a systematic literature review,” Artif Intell Rev, vol. 58, no. 7, Jul. 2025, doi: 10.1007/s10462-025-11176-z.

[7] R. A. F. Lustro, A. M. Sison, and R. P. Medina, “Performance analysis of enhanced speck algorithm,” in ACM International Conference Proceeding Series, Association for Computing Machinery, Oct. 2018, pp. 256–264. doi: 10.1145/3288155.3288196.

[8] A. Fanfakh and N. Abduljalil, “Performance Analysis of The Speck Cryptography Algorithm,” Journal of Innovations in Electronics and Computer Engineering, doi: 10.46649/fjiece.v4.2.3a.20.9.2025.

[9] T. Park, H. Seo, and H. Kim, “Parallel implementations of SIMON and SPECK,” in Proc. 2016 Int. Conf. Platform Technology and Service (PlatCon), 2016, pp. 1–6.

[10] Baraa Mohammed Hassan, “A proposed hybrid cryptography algorithm based on GOST and SPECK for data confidentiality in WoT,” AIP Conf Proc, no. Issue 1, May 2025, doi: https://doi.org/10.1063/5.0257346.

[11] Y. Yang, D. He, N. Kumar, and S. Zeadally, “Compact Hardware Implementation of a SHA-3 Core for Wireless Body Sensor Networks,” IEEE Access, vol. 6, pp. 40128–40136, Jul. 2018, doi: 10.1109/ACCESS.2018.2855408.

[12] Y. B. Kim, T. Y. Youn, and S. C. Seo, “Chaining optimization methodology: A new sha-3 implementation on low-end microcontrollers,” Sustainability (Switzerland), vol. 13, no. 8, Apr. 2021, doi: 10.3390/su13084324.

[13] J. James, R. Karthika, and R. Nandakumar, “Design & Characterization of SHA 3- 256 Bit IP Core,” Procedia Technology, vol. 24, pp. 918–924, 2016, doi: 10.1016/j.protcy.2016.05.184.

[14] B. Baldwin, A. Byrne, L. Lu, M. Hamilton, N. Hanley, M. O'Neill, and W. P. Marnane, "A hardware wrapper for the SHA-3 hash algorithms," in IET Irish Signals and Systems Conference (ISSC 2010), Stevenage, UK, 2010, pp. 1-6.

[15] E. Andreeva, B. Mennink, and B. Preneel, "Open problems in hash function security," Designs, Codes and Cryptography, vol. 77, no. 2, pp. 611-631, 2015.

[16] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, “The Simon and Speck Families of Lightweight Block Ciphers,” 2013.

[17] L. Sleem, R. Couturier, and R. Couturier, “Speck-R: An ultra light-weight cryptographic scheme for Internet of Things.” [Online]. Available: https://hal.science/hal-03359990v1

[18] E. Agullo et al., “RESILIENCY IN NUMERICAL ALGORITHM DESIGN FOR EXTREME SCALE SIMULATIONS,” 2020.

[19] A. Sevin, Ü. Çavu¸so, and Ç. ˘ Glu, “Design and Performance Analysis of a SPECK-Based Lightweight Hash Function,” 2024, doi: 10.3390/electronics.

[20] A. Biryukov, A. Roy, and V. Velichkov, "Differential analysis of block ciphers SIMON and SPECK," in International Workshop on Fast Software Encryption, Berlin, Heidelberg: Springer, 2014.

[21] R. Anand, A. Maitra, and S. Mukhopadhyay, "Evaluation of quantum cryptanalysis on speck," in International Conference on Cryptology in India, Cham: Springer International Publishing, 2020, pp. 395-413.

[22] L. A. Muhalha and I. S. Alshawi, “A hybrid modified lightweight algorithm for achieving data integrity and confidentiality,” International Journal of Electrical and Computer Engineering, vol. 13, no. 1, pp. 833–841, Feb. 2023, doi: 10.11591/ijece.v13i1.pp833-841.

[23] A. G. Sawant, S. Kamthe, Y. Shaha, B. Morajkar, and A. Sakpal, “Implementation of SIMON & SPECK Algorithm,” 2019. [Online]. Available: www.jetir.org

[24] A. Bossert, S. Cooper, and A. Wiesmaier, "A comparison of block ciphers SIMON, SPECK, and KATAN," TU Darmstadt, Tech. Rep., Sep. 2016.

[25] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "Notes on the design and analysis of SIMON and SPECK," Cryptology ePrint Archive, 2017.

[26] M. A. Kale and S. Dhamdhere, "Survey Paper on Different Type of Hashing Algorithm," International Journal of Advance Scientific Research and Engineering Trends, vol. 3, no. 2, Feb. 2018.

[27] D. K. P. Jayanti Sharma, Low Power and Pipelined Secure hashing Algorithm- 3(SHA-3). IEEE, 2016.

[28] S. Chang et al., “Third-Round Report of the SHA-3 Cryptographic Hash Algorithm Competition,” Gaithersburg, MD, Nov. 2012. doi: 10.6028/NIST.IR.7896.