INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2025, VOL. 71, NO. 4, PP. 1-6
Manuscript received September 11, 2025; revised September 2025. doi: 10.24425/ijet.2025.156530

Statistical analysis of enhanced *SDEx* encryption method based on BLAKE3 hash function

Artur Hłobaż, and Maciej Pawlak

***Abstract*—This paper presents a statistical analysis of the enhanced *SDEx* (Secure Data Exchange) encryption method, using a version that incorporates two session keys. This method has not previously been combined with the BLAKE3 hash function. The statistical analysis was conducted using the NIST Statistical Test Suite. Several real-world sample files were encrypted using the proposed method and then subjected to statistical analysis through selected tests from the NIST suite. These tests aimed to determine whether the resulting ciphertexts meet the criteria for pseudorandomness. Additionally, compression tests were performed using WinRAR, which confirmed that the ciphertexts are not compressible.**

***Keywords*—secure communication; data encryption; data security; secure transmission; secure data exchange method; *SDEx* method; end-to-end data security; NIST Statistical Test Suite**

I. Introduction

MODERN digital infrastructure is experiencing unprecedented growth in the volume of transmitted data, which directly impacts the demand for efficient and secure information protection mechanisms. As highlighted in the Cisco Annual Internet Report (2018–2023) White Paper [1], global IP traffic has been growing exponentially—with over 29 billion connected devices by 2023 and an increasing amount of data generated by mobile applications, cloud computing, and the Internet of Things (IoT). Similar trends have been observed by other researchers, who emphasize the rapid expansion of end-user devices and applications that require secure data transmission [2],[3].

In parallel with this growth, the demands for data encryption are also increasing – not only in terms of the level of security provided but also regarding time and energy efficiency. Recent literature emphasizes that data protection – both in transit and at rest – has become one of the most critical challenges facing modern information systems [4],[5]. High entropy of ciphertexts and resistance to cryptographic attacks must go hand in hand with minimal computational overhead, especially in cloud environments and edge computing systems [6].

The performance of cryptographic algorithms is crucial not only from the perspective of end users but also for data centers and large-scale systems that process vast amounts of data. Reducing the number of processor cycles required to perform cryptographic operations directly translates into energy savings, which significantly impacts the operational costs and energy efficiency of data infrastructure [7],[8]. As pointed out in works such as [9],[10], energy savings achieved by optimizing the cryptographic layer can be a critical factor in supporting the sustainable development of information technologies.

In this context, increasing attention is being given to the design of modern, lightweight cryptographic algorithms that combine high attack resistance with computational efficiency. Studies such as [11]-[13] indicate that the development of new hash functions and encryption methods optimized for parallelization and streaming (e.g., BLAKE2, BLAKE3, SipHash) is essential for scalable and future-proof security systems. The implementation of such algorithms can provide measurable benefits both in resource-constrained environments (embedded systems) and in distributed cloud infrastructures serving millions of users in real time.

This paper presents a statistical analysis of the Enhanced Secure Data Exchange (*SDEx*) encryption method, strengthened through the use of the BLAKE3 cryptographic hash function. The statistical quality of the resulting ciphertext – alongside encryption speed and security level – is one of the essential indicators of a cryptographic algorithm's effectiveness. Notably, *SDEx* has the potential to serve as an alternative to the widely used AES algorithm. A core component of *SDEx* is the hash function, which plays a central role in its encryption process.

Previous studies have examined *SDEx* in conjunction with the SHA-256 and SHA-512 hash functions, as documented in the publications "Statistical Analysis of Enhanced *SDEx* Encryption Method Based on SHA-256 Hash Function" [14] and "Statistical Analysis of Enhanced *SDEx* Encryption Method Based on SHA-512 Hash Function" [15]. In both cases, the method passed all statistical tests and was deemed secure for practical use. Additionally, the publication "Analysis of the Possibility of Using Selected Hash Functions Submitted for the SHA-3 Competition in the *SDEx* Encryption Method" [16] explored various SHA-3 candidate algorithms and concluded that BLAKE3 was the only one meeting all necessary criteria for secure integration with *SDEx*.

Given BLAKE3's high security – comparable to that of SHA-256 and SHA-512 – along with its superior performance and

This work is a continuation of research carried out thanks to a grant from the National Science Centre (no. 2020/04/X/ST6/00407).

Artur Hłobaż is with the Faculty of Mathematics and Computer Science University of Lodz, Poland (e-mail: artur.hlobaz@uni.lodz.pl) and with the Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology, Poland (e-mail: artur.hlobaz@p.lodz.pl).

Maciej Pawlak is a graduate of the University of Lodz (e-mail: maciej.pawlak@edu.uni.lodz.pl).

flexibility (e.g., support for data streaming), it stands out as a particularly attractive choice for real-time, high-throughput encryption applications.

II. Enhanced Secure Data Exchange (*SDEx*) method

The *SDEx* (Enhanced Secure Data Exchange) method has already been presented in the author's previous publications [14]-[19].

This method is a block cipher algorithm, and its security and speed are based on the security and speed of the hash function used, which acts as a dynamic pseudorandom bit sequence generator. Below (Figure 1) is a slight modification of the method presented that allows for two-way authentication of both the sender and the recipient. The main session key is created from two subkeys exchanged by the sender and the recipient using asymmetric cryptography. We can describe this modification as an enhanced *SDEx* encryption method with a double session key. It may be used in situations where authentication of the sender and recipient is required, e.g. in instant messaging [19].

The symbols used in the encryption scheme (Figure 1) and in the equations (1-11) are shown below:

- $M_1, M_2, \ldots M_i$ - plaintext blocks,
- $C_1, C_2, \ldots C_i$ - ciphertext blocks,
- $h_1, h_2, \ldots h_k$ - particular iterations of hash computation,
- H_{S1} - hash from the first session key,
- H_{S2} - hash from the second session key,
- H_{S1+S2} - hash from the concatenation of first and second session key,
- $\oplus$ - XOR operation,
- $+\!\!+$ - concatenation of two strings.

The individual steps of the *SDEx* encryption method are described by equations and have the following form:

$$C_1 = M_1 \oplus H_{S1} \oplus H_{S1+S2} \tag{1}$$

$$C_2 = M_2 \oplus H_{S1} \oplus H_{S2} \tag{2}$$

$$C_{2k+1} = M_{2k+1} \oplus h_k \oplus h_{k-1} \qquad k \geq 1 \tag{3}$$

$$C_{2k+2} = M_{2k} \oplus H_{S2} \oplus h_k \qquad k \geq 1 \tag{4}$$

$$h_1 = hash\,(H_{S1+S2}; M_1 +\!\!+ M_2) \tag{5}$$

$$h_2 = hash\,((h_1 \oplus H_{S1+S2}); M_3 +\!\!+ M_4) \tag{6}$$

$$h_k = hash\,((h_{k-1} \oplus h_{k-2}); M_{2k-1} +\!\!+ M_{2k}) \qquad k \geq 3 \tag{7}$$

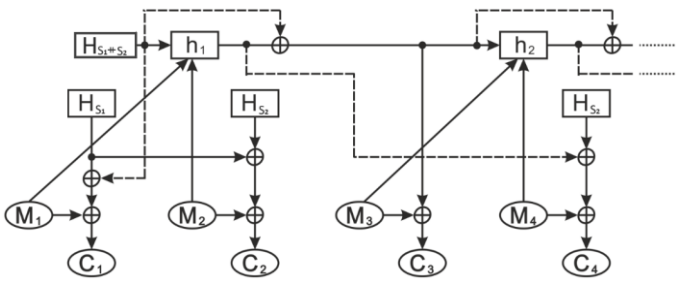


Fig. 1. Enhanced *SDEx* encryption method with a double session key.

The decryption process is described by equations:

$$M_1 = C_1 \oplus H_{S1} \oplus H_{S1+S2}, \tag{8}$$

$$M_2 = C_2 \oplus (H_{S1} \oplus H_{S2}) \tag{9}$$

$$M_{2k+1} = C_{2k+1} \oplus h_k \oplus h_{k-1} \qquad k \geq 1 \tag{10}$$

$$M_{2k+2} = C_{2k} \oplus H_{S2} \oplus h_k \qquad k \geq 1 \tag{11}$$

The above formulas, describing the encryption and decryption process using the *SDEx* method, show that the main factor influencing the efficiency of the algorithm is the efficiency of the hash function used. The publication [16] presents a comparison of the computational efficiency of selected popular hashing algorithms. It shows that the BLAKE 3 hash function offers the highest efficiency per processor thread, as well as a better security margin than hash functions from the Grøstl, JH or Keccak groups.

In the paper [16] it was also proved that the use of the BLAKE hash function in the *SDEx* encryption method, instead of hash functions from the SHA-2 family (SHA-256 and SHA-512), will increase the speed of its operation several or dozen times (depending on the use of the BLAKE2 or BLAKE3 hash function) compared to AES while maintaining similar level of security (Figure 2).

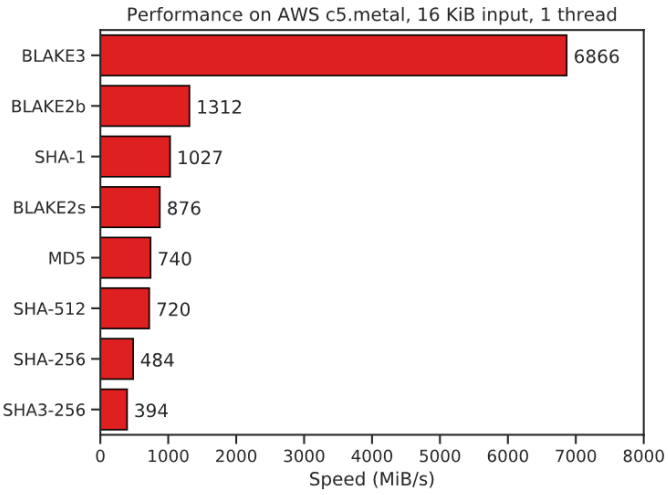


Fig. 2. Speed comparison of BLAKE3 to other popular hash functions [20].

In summary, the *SDEx* method will find application in almost every case where symmetric encryption is required. The version presented in this paper with two session keys using the BLAKE3 hash function will find the best application in the context of various types of network applications.

III. Research of the pseudo-randomness of the test files encrypted with the *SDEx* method

The methodology for testing the pseudo-randomness of test files encrypted using the *SDEx* method was the same as in the author's previous works [14], [15]. Five real files (not artificially generated) were selected for testing:

- ***TXT*** – text file with the letter "a" (10 000 000 bytes),
- ***TXT*** – text file containing the book's content (493 764 bytes),

- ***PDF*** – documentation of BLAKE3 (304 371 bytes),
- ***JPG*** – graphic file (4 226 902 bytes),
- ***MP3*** – music file (4 250 112 bytes).

The tests are performed using a program developed by NIST, the NIST Statistical Test Suite [21].Each file will be statistically analyzed for four tests, which were selected based on the publication [22]:

1. ***Frequency test*** - the purpose of this test is to determine whether the number of ones and zeros in a sequence is approximately the same as would be expected for a truly random sequence (minimum sequence size: 100 bits, acceptable p-value: ≥ 0.01).
2. ***Block Frequency test*** - the purpose of this test is to determine whether the frequency of ones in an M-bit block is approximately M/2, as would be expected under the assumption of randomness (minimum sequence size: 100 bits, acceptable p-value: ≥ 0.01).
3. ***Cumulative sum*** - the purpose of the test is to determine whether the cumulative sum of partial sequences occurring in the tested sequence is too large or too small in relation to the expected behavior of the cumulative sum for random sequences (minimum sequence size: 100 bits, acceptable p-value: > 0.01).
4. ***Runs test*** - the purpose of the Runs test is to determine whether the number of runs of ones and zeros of varying lengths is as expected for a random sequence. Specifically, this test determines whether the oscillation between such ones and zeros is too fast or too slow, which allows us to determine whether the sequence is random (minimum sequence size: 100 Bits, acceptable p-value: ≥ 0.01).

The number of sequences has been set to 1000 as a constant value. Due to varying file sizes, the number of bits in a sequence will need to be variable (see Table 1).

TABLE I
NUMBER OF BITS IN A SEQUENCE AND THEIR SIZES FOR INDIVIDUAL FILES

File type	File size (bytes)	Number of bits per sequence
TXT (a)	10 000 000	80 000
PDF	304 371	2 430
JPG	4 226 902	33 687
TXT	493 764	3 840
MP3	4 250 112	33 920

Additionally, each file will be subjected to a compression test using WinRAR to ZIP format in the "best" compression mode (see chapter 3).

Below there are the results of tests of individual files, which are presented in the form of:

- a histogram showing the frequency of the occurrence of P-values on 10 subintervals of interval [0,1]; all P-values from all tests were taken to the histogram (4 tests with 1000 sequences in each = 4000 P-values); the distribution should be as close as possible to the even distribution,
- a chart on which the individual numbers of statistical tests are on the X axis (order as in assumptions) and corresponding to them sequence proportions that passed the test for the selected significance level α = 0.01 on the Y axis; the ideal result is a situation in which 1 out of 100 sequences would be rejected or in other words 99% of the sequences would pass the test (on the graph it is a green line); the dashed gray line shows the minimum acceptable sequence ratio that should pass the selected statistical test - in presented research it is 98,06%.

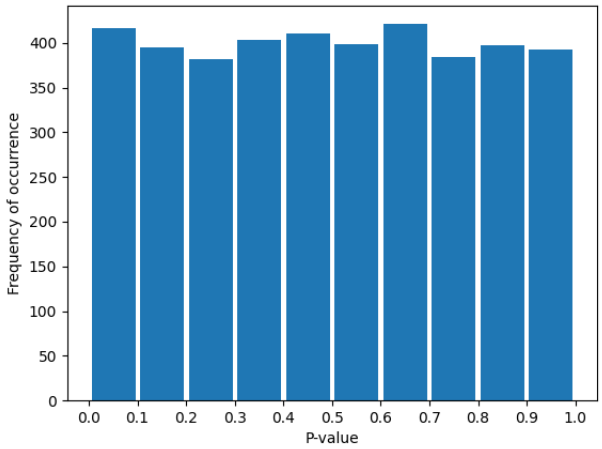
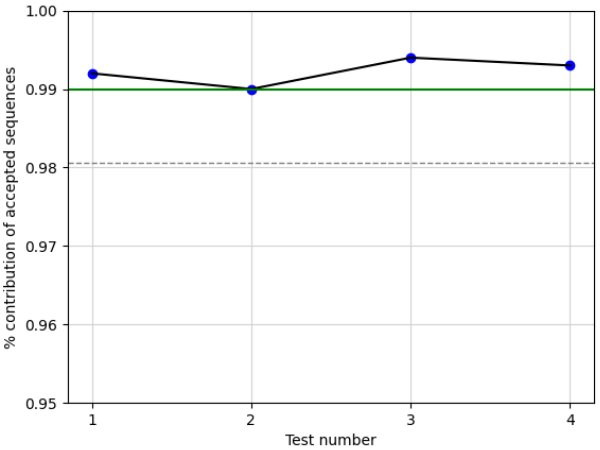


Fig. 3. Histogram for the cryptogram of the TXT file with the letter "a".



Fig, 4. Chart for the cryptogram of the TXT file with the letter "a".

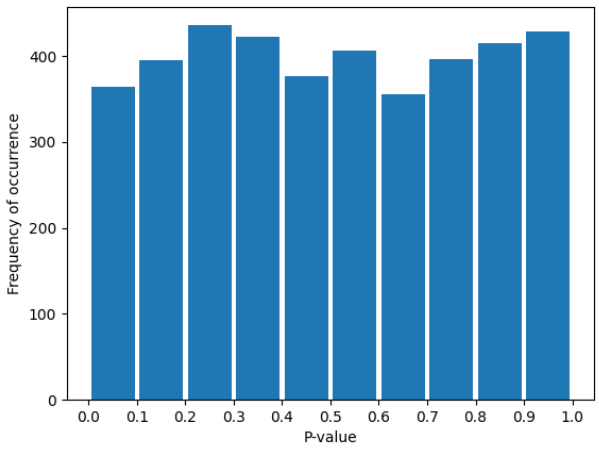


Fig. 5. Histogram for the cryptogram of the PDF file.

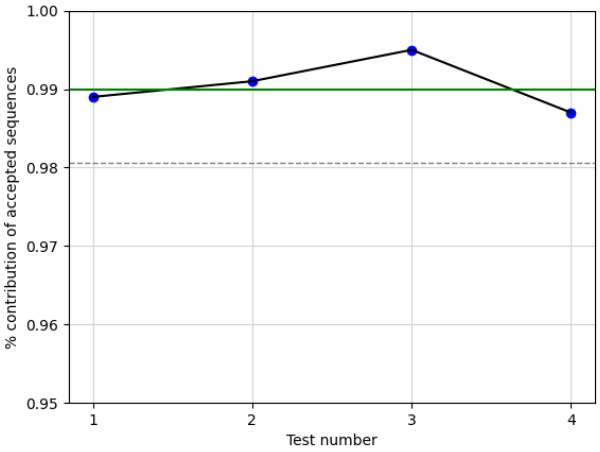


Fig. 6. Chart for the cryptogram of the PDF file.

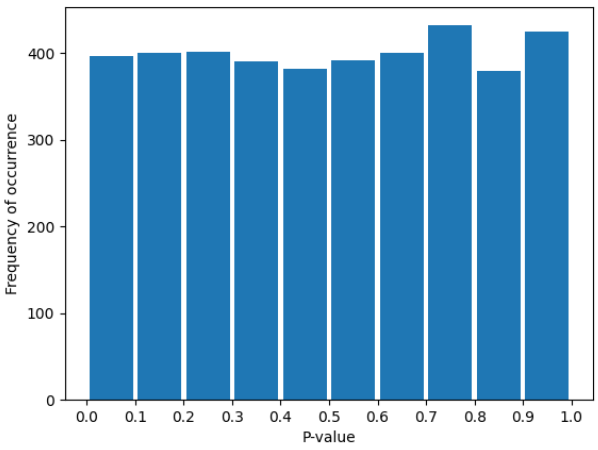


Fig. 7. Histogram for the cryptogram of the JPG file.

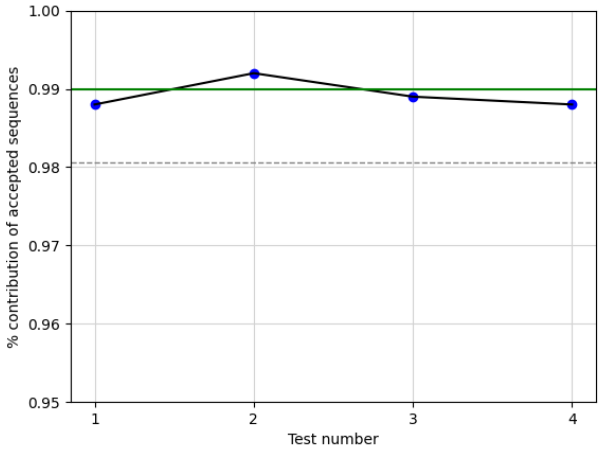


Fig. 8. Chart for the cryptogram of the JPG file.

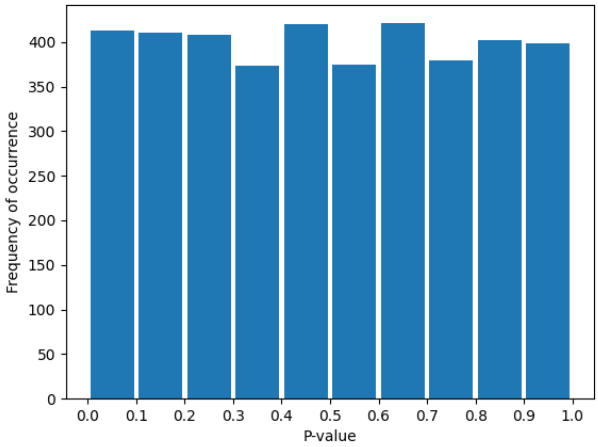


Fig. 9. Histogram for the cryptogram of TXT file of the book.

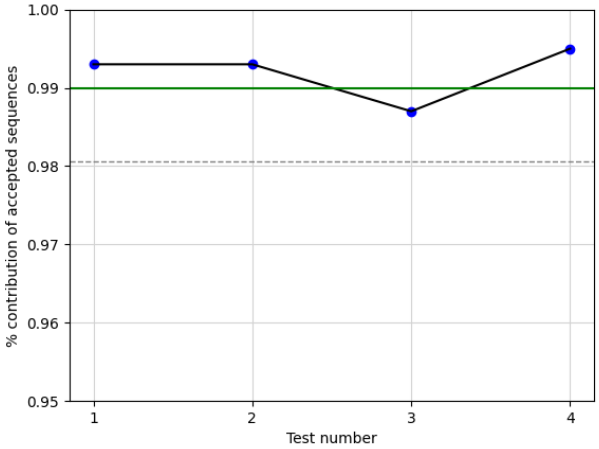


Fig. 10. Chart for the cryptogram of TXT file of the book.

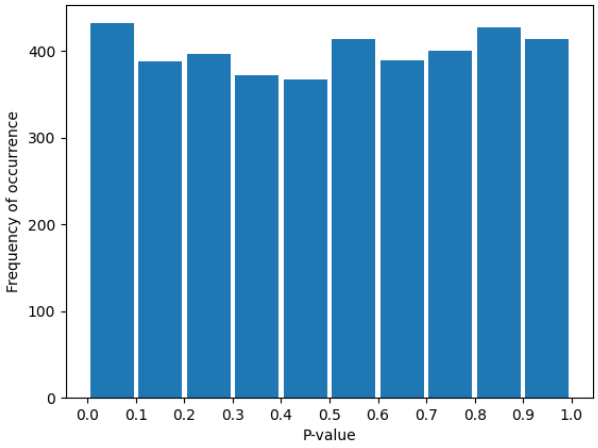


Fig. 11. Histogram for the cryptogram of the MP3 file.

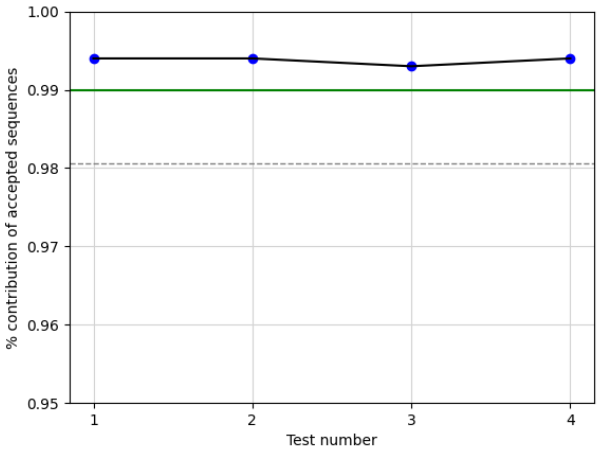


Fig. 12. Chart for the cryptogram of the MP3 file.

To summarize the results, each of the p-value distribution tests shows an even distribution among the given value ranges. Also, each test of accepted sequences for each file has a value close to the expected value of 0.99 and not less than 0.9805607 defined as the acceptable minimum.

IV. Compression Tests

Since the compression tests using the NIST package could not be performed due to the insufficient size of the test files, these tests were replaced by an alternative innovative approach presented in [23]. Based on [23], to determine whether a file is a random bit sequence, it should not be compressible. In other words, the size of the file after compression should be at least the same as its original size. The latest version of WINRAR 6.24 was used for the compression tests, and the files were compressed in the "BEST" mode. The following table (see Table II) shows the results of these tests.

As a result of compressing files encrypted with the *SDEx* method using the BLAKE3 hash function, it can be seen (see Table II) that the size of each file has increased minimally due to the additional information placed in their headers, such as the file name, size, etc. Their sizes are approximately 200-1700 bytes larger than the corresponding unencrypted files. To summarize, it can be stated that the *SDEx* method with BLAKE hash function applied successfully passed compression tests for selected test files.

TABLE II
RESULTS OF COMPRESSION TESTS

File type	File size (Bytes)	File size after compression (Bytes)		Level of compression (%)	
		unencrypt.	encrypted	unencrypt.	encrypted
TXT (a)	10 000 000	10 603	10 001 682	99,89	0,00
PDF	304 371	300 536	304 583	1,26	0,00
JPG	4 226 902	4 150 942	4 227 707	1,80	0,00
TXT	493 764	208 128	494 016	57,85	0,00
MP3	4 250 112	4 229 870	4 250 920	0,48	0,00

Conclusions

In the face of the dynamic development of digital infrastructure and the growing demand for secure and efficient data protection mechanisms, it is essential to design modern cryptographic solutions that combine strong attack resistance with low computational overhead. This article presented a statistical analysis of the Enhanced Secure Data Exchange (*SDEx*) encryption method, utilizing the BLAKE3 hash function—an algorithm distinguished by both a high level of security and excellent performance, particularly in real-time environments and distributed systems.

Previous research has shown that *SDEx*, in combination with various hash functions, meets rigorous cryptographic quality requirements, and its integration with BLAKE3 represents a significant step toward improving its efficiency. Thanks to its properties, BLAKE3 may serve as a robust alternative to traditional functions such as SHA-256 or SHA-512, while offering superior performance in the context of modern, scalable information systems.

These findings confirm the importance of further developing and implementing lightweight, optimized cryptographic algorithms in response to contemporary challenges related to data security and energy-efficient processing.

References

[1] Cisco „Cisco Annual Internet Report (2018–2023) White Paper", 2020

[2] M. Satyanarayanan, "The Emergence of Edge Computing" in Computer, vol. 50, no. 1, pp. 30-39, Jan. 2017. https://doi.org/10.1109/MC.2017.9

[3] H. HaddadPajouh, R. M. Parizi, "A survey on Internet of Things security: Requirements, challenges, and solutions", Internet of Things, vol. 14, pp. 100-129, 2019. https://doi.org/10.1016/j.iot.2019.100129

[4] W. Stallings, "Cryptography and Network Security: Principles and Practice", 8th ed., Pearson, 2023

[5] B. Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C", 20th Anniversary Edition, Wiley, 2015

[6] R. Roman, et al., "Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges", Future Generation Computer Systems, vol. 78, Part 2, pp. 680-698, 2018. https://doi.org/10.1016/j.future.2016.11.009

[7] S. Thakur, S. Banik and F. Regazzoni, "Energy Analysis of Cryptographic Algorithms in Server Environment", In Proceedings of the 2024 on Cloud Computing Security Workshop (CCSW '24), ACM, New York, pp. 3–14, 2024. https://doi.org/10.1145/3689938.3694775

[8] J. Soto-Cruz, et al., "A Survey of Efficient Lightweight Cryptography for Power-Constrained Microcontrollers" Technologies, *13*(1), 3, 2025. https://doi.org/10.3390/technologies13010003

[9] T. K. Goyal, et al., “Energy Efficient Lightweight Cryptography Algorithms for IoT Devices, IETE Journal of Research, 68(3), pp. 1722–1735, 2019. https://doi.org/10.1080/03772063.2019.1670103

[10] M. Minier, et al., “Energy-Efficient Cryptographic Engineering Paradigm”, In: Camenisch, J., Kesdogan, D. (eds) Open Problems in Network Security. iNetSec 2011. Lecture Notes in Computer Science, vol. 7039, pp. 78-80. Springer, Berlin, 2011. https://doi.org/10.1007/978-3-642-27585-2_7

[11] J.-P. Aumasson, et al., "BLAKE2: Simpler, Smaller, Fast as MD5”, In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (eds) Applied Cryptography and Network Security, ACNS 2013, Lecture Notes in Computer Science, vol 7954. Springer, Berlin, 2013. https://doi.org/10.1007/978-3-642-38980-1_8

[12] J.-P. Aumasson, "BLAKE3: one function, fast everywhere", 2020. https://github.com/BLAKE3-team/BLAKE3

[13] National Institute of Standards and Technology, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", FIPS PUB 202, 2015. https://doi.org/10.6028/NIST.FIPS.202

[14] A. Hłobaż, "Statistical Analysis of Enhanced SDEx Encryption Method Based on SHA-256 Hash Function" 2019 IEEE 44th Conference on Local Computer Networks (LCN), Osnabrueck, Germany, pp. 238-241 2019. https://doi.org/10.1109/LCN44214.2019.8990714

[15] A. Hłobaż, "Statistical Analysis of Enhanced SDEx Encryption Method Based on SHA-512 Hash Function" 2020 29th International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, 2020, pp. 1-6. https://doi.org/10.1109/ICCCN49398.2020.9209663

[16] A. Hłobaż, “Analysis of the Possibility of Using Selected Hash Functions Submitted for the SHA-3 Competition in the SDEx Encryption Method”, International Journal of Electronics and Telecommunications, 2022, pp. 57-62. https://doi.org/10.24425/ijet.2022.139848

[17] A. Hłobaż, K. Podlaski, P. Milczarski, “Enhancements of encryption method used in SDEx, Communications in Computer and Information Science”, vol. 718, pp. 134-143, Springer International Publishing, 2017. http://doi.org/10.1007/978-3-319-59767-6_11

[18] P. Milczarski, A. Hłobaż, K. Podlaski, “Analysis of enhanced SDEx method”, Proceedings of the 2017 IEEE 9th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS, 2017. https://doi.org/10.1109/IDAACS.2017.8095245

[19] Milczarski P., Podlaski K., Hłobaż A., Applications of Secure Data Exchange Method Using Social Media to Distribute Public Keys, Communications in Computer and Information Science Vol. 522, pp. 389-399, Springer International Publishing, 2015

[20] https://github.com/BLAKE3-team/BLAKE3/

[21] A. Rukhin, et al., “A statistical test suite for random and pseudorandom number generators for cryptographic applications”, NIST Special Publication 800–22, National Institute of Standards and Technology, 2010. https://doi.org/10.6028/NIST.SP.800-22r1a

[22] J. Soto, “Statistical Testing of Random Number Generators”, NIST, 1999. https://csrc.nist.rip/nissc/1999/proceeding/papers/p24.pdf

[23] G. Szewczyk, “The Dynamic Ciphers - New Concept Of Long-Term Content Protecting”, Annales Universitatis Apulensis Series Oeconomica, Faculty of Sciences, "1 Decembrie 1918" University, Alba Iulia, vol. 2(10), pp. 1-34, 2008.