Information technologies in economics and environmental sciences

Journal homepage: https://journals.nubip.edu.ua/index.php/Inf/
ISSN 3083-7502

COMPUTER SCIENCE ◈ КОМП'ЮТЕРНІ НАУКИ

Received 2026-03-19 *Accepted 2026-04-13* *Published 2026-04-22*

UDC 004.056.55:004.421.2 **DOI: 10.31548/itees.2026.01.045**

Sahun Andrii
PhD, Associated Professor of the Department of Computer Systems, Networks and Cybersecurity, National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0000-0002-5151-9203
E-mail: a.sagun@nubip.edu.ua

FROM MERKLE–DAMGÅRD TO SPONGE: ARCHITECTURAL IMPACT ON HASH FUNCTION SECURITY

Abstract. *The paper investigates the influence of cryptographic hash function architecture on their cryptographic strength. The main focus is on a comparative analysis of the classical Merkle–Damgård architecture used in the SHA-2 family and the Sponge architecture implemented in the SHA-3 standard. It is shown how the design features of the Sponge architecture, in particular the division of the internal state into speed (rate) and capacity parts, provide an increased margin of cryptographic strength and ensure low vulnerability to the inherent Merkle–Damgård constructions, including the message extension attack. The possibility of estimating the dispersion index for attributing a hash function to a cryptographic type has been confirmed. At the same time, the question remains about the unambiguity of the correspondence between theoretical statistical indicators of the quality of hash functions. The only known indicator of the quality of hash functions is based on the variance indicator and unambiguously shows only whether a particular hash function belongs to cryptographic or non-cryptographic. At the same time, it has been confirmed that the χ^2 test, as a "bias detector" can prove that the hash function is hack-resistant with high probability. But the question remains about the unambiguity of the correspondence between theoretical statistical indicators of the hash functions quality.*

Keywords: *Cryptographic Hash Functions; SHA-3; SHA-2; Merkle–Damgård Architecture; Sponge Architecture; Crypto Resistance; Safety Margin; Post-quantum Security.*

Introduction. Today cryptographic hash functions play a critical role in contemporary cybersecurity, enabling secure data storage, digital signatures, authentication subsystems, blockchain technologies and others applications. Modern secure hash function must satisfy principal properties such as preimage resistance, second preimage resistance, collision resistance.

For decades, the construction of Merkle–Damgård, using in SHA-1, SHA-2 hash-functions served as the de facto standard for building the most popular iterative hash functions. Despite its theoretical foundations, practical cryptanalysis has demonstrated that the MD paradigm introduces structural vulnerabilities that can be exploited independently of the underlying compression function [1, 2].

Otherwise, the Sponge construction, standardized through SHA-3, demonstrates a significant architectural departure.

The purpose of the research is to establish and demonstrate the relationship between the statistical parameters of hash functions based on the Merkle–Damgård and sponge architectures and cryptographic stability and the Security Margin parameter.

Literature Review. Today, there are three main approaches to constructing cryptographic hash functions:

- Merkle–Damgård.
- HAIFA;
- Sponge.

It is known from a number of sources that the sponge architecture of the SHA-3 hash function is significantly different from the architecture of the SHA-2 function based on Merkle–Damgård [3-4]. From practical application and results of cryptanalysis, it is known that the architecture determines the resistance to structural attacks, the possibility of expansion and the level of cryptographic reserve [5–7].

Modern cryptographic hash functions mainly consist of: 1) a block of internal permutation/compression function; 2) message processing mode block. At the same time, the sponge architecture has a unique resistance not only to the existing traditional methods of cryptanalysis, but also to quaternary methods of co-promotion of the function [4, 8, 9].

The sponge architecture of the SHA-3 function provides a disproportionately higher level of structural cryptoresistance compared to the classic Merkle–Damgård architecture used in the SHA-2 function. The division of the internal state into "rate" and "capacity" in SHA-3 allows this algorithm to formally control the margin of cryptographic strength. The same division eliminates the vulnerabilities of the SHA-3 hash function by the length-extension type, which makes it optimally suitable for modern information systems with increased security.

Comparing the theoretical cryptographic strength, one can see a strong difference between these three basic architectures (Table 1).

Table 1 – Comparison of theoretical cryptographic strength of basic hash function architectures

Indicators of cryptoresistance	**Architecture of hash functions**		
	Merkle–Damgård (MD5, SHA-1, SHA-2)	**HAIFA (BLAKE, SHAvite-3)**	**Sponge (SHA-3)**
Construction type	iterative	iterative (extended)	permutation
Inner state	n bits	n bits + salt + counter	$b = r + c$ bits
Function of compression	available	available (modificated)	disable
Padding	obligated	obligated	obligated
Length-extension attack	possible	removed partial	removed
Salt / randomization	disable	available	available (capacity)
Formal security evidence	limited	partial	strong
Output length flexibility	disable	disable	available (XOF, SHAKE)

Materials and Methods. Merkle–Damgård is a classic iterative scheme that involves initial addition of the input bit data, dividing it into blocks, followed by compression on each block. The scheme includes an initializing vector (IV) for 4 registers. At the end of the last round, we get the final hash value. This architecture is used in hash functions MD5, SHA-1, SHA-2.

In connection with the revealed presence of problems related to the cryptoresistance of hash functions based on the Merkle–Damgård architecture, described in [10-14], its modification - HAIFA architecture (Hash Iterative Framework Alternative) was proposed over time. This construction is an extension of Merkle–Damgård with some modifications. Namely: the cryptographic "salt" is a pseudo-random value for each call (eliminates reproducible hashes); block counter — takes into account the position of the block in the message; a modified compressor including an initialization vector (IV), a salt, and a counter.

All modifications of the basic Merkle–Damgård scheme available in the HAIFA architecture significantly improve its quality. This provides the following advantages to the HAIFA architecture: the Salt mechanism protects the architecture from attacks using "rainbow" tables by pre-calculating hash tables; the block counter counter neutralizes certain forms of attacks related to block positions.

Overall, such architectural improvements provide better security control compared to the classic Merkle–Damgård architecture. At the same time, there are certain limitations associated with the fact that the Merkle–Damgård architecture is based on an iterative approach. Therefore, not all the weaknesses of the well-studied Merkle–Damgård scheme are excluded (the "length-extension" type dependence is only partially reduced, but not completely excluded - unlike the sponge architecture). Therefore, the HAIFA architecture requires the use of additional mechanisms for increasing stability (salt, counter) — therefore, it has a more complex implementation.

Table 2 summarizes the comparative characteristics of resistance to popular vulnerabilities of all three considered hash function architectures.

Table 2 – Comparative characteristics of resistance to popular vulnerabilities of the Merkle–Damgård, HAIFA, Sponge architectures

Hash function architecture name	**Sensitivity to vulnerability type**			
	Length-extension	**Capacity-based**	**Flexibility**	**Theoretical safety evidence**
Merkle–Damgård	possible	disable	fixed	limited
HAIFA	partially reduced	partially available	improved	avarege
Sponge	disable	yes	high (XOF)	formal, strong

A theoretical comparison of the security levels of the classical (Merkle–Damgård) architecture of hash functions of the SHA type, its modernized version (HAIFA, and the Sponge architecture is particularly revealing. Table 3 shows the results of the collision resistance of classical and quantum evaluation.

Table 3 – Security level comparison under classical and quantum attack models

Construction	**Collision resistance (classical)**	**Prototypical resilience**	**Quantum assessment (Grover)**
Merkle–Damgård (SHA-256)	$\approx 2^{128}$	$\approx 2^{256}$	$\approx 2^{128}$
HAIFA (256 біт)	$\approx 2^{128}$	$\approx 2^{256}$	$\approx 2^{128}$
Sponge (SHA3-256)	$\approx 2^{128}$	$\approx 2^{256}$	$\approx 2^{128}$
Sponge (SHA3-512)	$\approx 2^{256}$	$\approx 2^{512}$	$\approx 2^{256}$

When interpreting the quantum security assessment for all hash functions listed in Table 3, we take into account that the Grover algorithm is optimal [15]. As can be seen from the data in Table 3, the overall estimates of collision occurrence in hash functions for classical brute-force and the quantum estimation algorithm (Grover) coincide, which is expected. The advantage of the Grover algorithm is that it speeds up traditional brute-force by allowing collision detection much faster than classical methods, while reducing the computational complexity to $\Theta\left(2^{n/2}\right)$.

Therefore, to counter quantum attacks, it is considered necessary to at least double the hash size (for example, from SHA-256 to SHA-512, etc.).

For a more objective comparison, we programmatically implement the MD5 hash function algorithms according to its official description given in the source [14], and the SHA-3 hash function algorithms according to the description given in [16].

Results and Discussion. We have the obtained results for the data of the input test examples, formed in the following indicators:

1) the average proportion of "1" bits;

2) bit dispersion values for hash functions;

3) indicator of the avalanche effect;
4) distribution of integer values of digests.

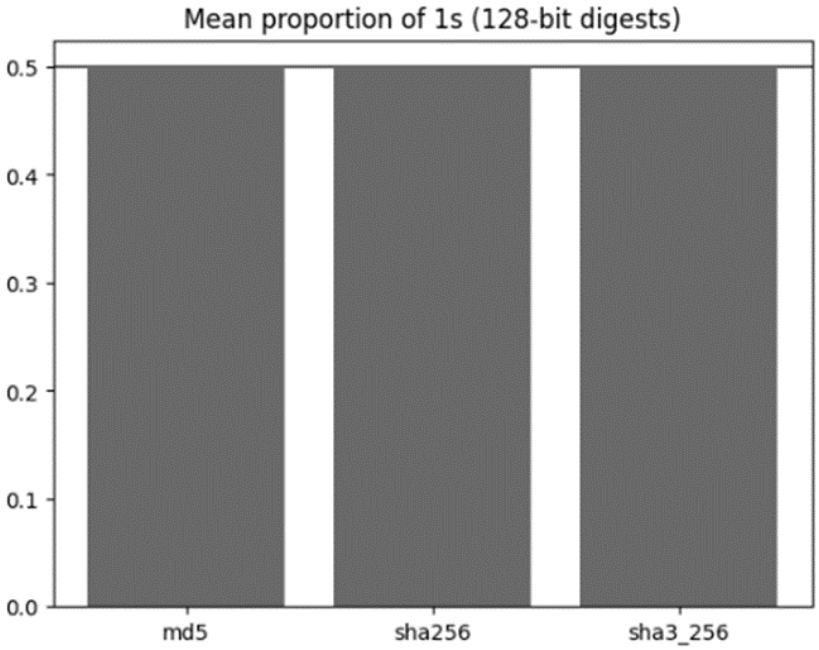

Figure 1 – The average proportion of "1" bits

The average proportion of "1" bits (Fig. 1) for MD5 is 0.499548 (0.0904%), for SHA3-128 this parameter = 0.5005 (0.1%), and for SHA-256 it is 0.499444 (0.112%). All indicators are very close to the theoretical 0.5.

To further investigate the quality of hash functions, we will use χ^2 tests. Such a test is an important statistical tool for assessing the quality of hash functions because for cryptographic applications, a hash function should have a uniform distribution of values (resulting in a reduced probability of collisions). This is the uniformity of the distribution (uniformity) [17].

If a hash function has such flaws, it will most likely create “hot spots” — areas where values fall more often than others.

The generalized analysis with the added χ^2-test of uniformity and comparison of MD5 against SHA-2 (SHA-256) and SHA-3 (SHA3-256) is shown in Table 4.

Table 4 – Bit uniformity values

Algorithm’s name	**Characteristics**			
	Avarage proportion of bis «1»	**Dispartion**	**χ^2 (bits)**	**p-value**
MD5	0.499548	0.250000	0.522	0.46999
SHA-256	0.499444	0.250000	0.792	0.37347
SHA3-256	0.498905	0.249999	3.071	0.07969

Analyzing the data presented in Table 3, it can be stated that all statistical indicators are very close to theoretical ones: the average proportion of bits "1" = 0.5, the dispersion value typical for cryptographic functions = 0.25 [7], and the p-value parameter > 0.05 in all cases (there is no reason to reject the existing hypothesis of uniformity). But the SHA-3 algorithm showed a slightly larger value of the χ^2 parameter, but this is a statistically insignificant number. Next, we obtain the χ^2-test for uniformity of bytes (0..255) – Table 5.

Table 5 – χ^2-byte uniformity test (0..255)

Algorithm's name	χ^2 (bytes)	p-value
MD5	282.94	0.11045
SHA-256	314.38	0.00660
SHA-3-256	251.90	0.54315

In the χ^2 test for byte uniformity, the SHA-3 algorithm showed the best uniformity of byte distribution, and SHA-256 has a uniformity index of $p < 0.01$, which is formally a statistical deviation from ideal uniformity (Figure 2). But such a deviation does not mean cryptographic weakness - under the conditions of using a large sample, even small fluctuations remain statistically significant. The MD5 algorithm in this test showed the accepted uniformity.

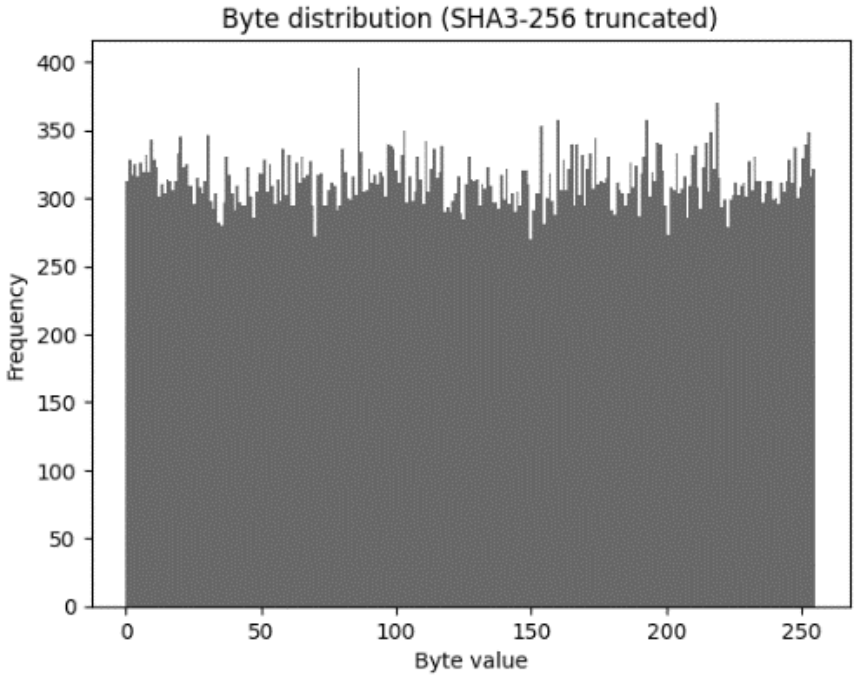

Figure 2 – Graphics of byte distribution for SHA3-256 truncted

The avalanche effect is of great importance as a characteristic of a hash function. The theoretical ideal for a 128-bit value is calculated as: 128/2 = 64, for a 256-bit value it is 128 bits.

Table 6 – Parameters of hash-function near-perfect avalanche effect

Algorithm's name	Average Hamming's distance	Std
MD5	63.953	5.695
SHA2-256	64.034	5.599
SHA-3-256	63.888	5.894

All algorithms are shown on Table 6 demonstrate a near-perfect avalanche effect, but SHA2-256 showed a result closest to the theoretical value of 64. Graphically, the distribution has a shape close to normal (corresponding to the binomial distribution B(128, 0.5)) and is shown in Figure 3.

The definition of the Security margin parameter in modern hash functions is particularly relevant. This is due to the fact that the Security margin is the difference between the declared theoretical stability of the algorithm and the computational complexity of the most effective known cryptoattack (expressed in the number of rounds or bits of endurance).

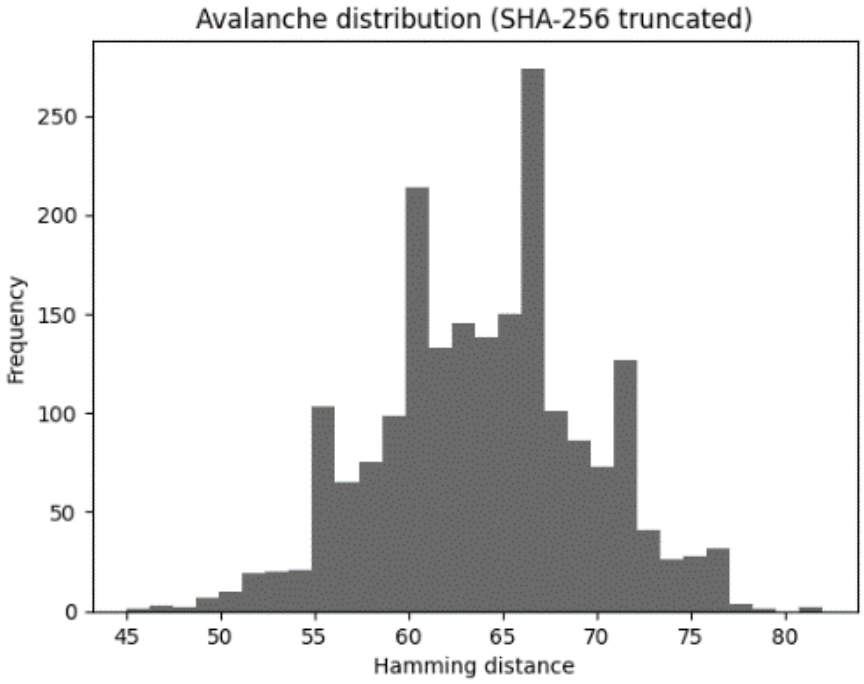

Figure 3 – Avalanche distribution for SHA-256 truncted

In fact, this parameter shows how reliable the function is if some of its rounds are compromised. Naturally, the increased reserve provides resistance to future cryptanalytic discoveries. Let's evaluate the Security margin parameter for two candidates (SHA2-256 and SHA3-256) as the most promising functions (Figure 4).

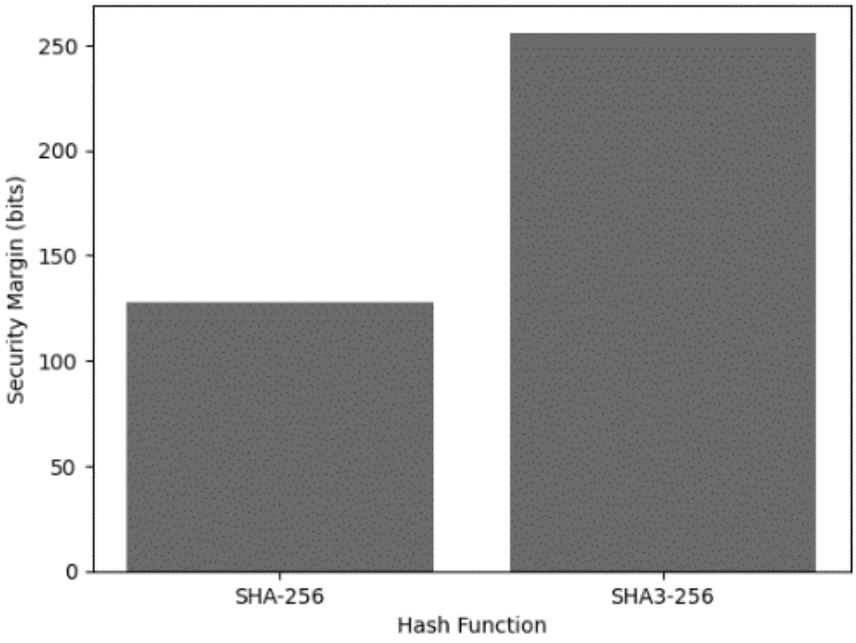

Figure 4 – Security Margin Comparison: SHA-2 vs SHA-3

We show in Figure 5 a graph with a comparison of the traditional and quantum estimates of the Security margin parameter, taking into account the results of the classical birthday paradox algorithm and the Grover algorithm for the SHA2-256 and SHA-3 hash functions. As you can see, due to its Sponge architecture, the Security Margin parameter is clearly almost an order of magnitude higher for the SHA3-256 algorithm. At the same time, all algorithms demonstrate: proximity to an

equiprobability distribution and a dispersion close to 0.25 (typical for all purely cryptographic functions) and a perfect avalanche effect. But, although MD5 algorithm statistically "looks good" by qualitative statistical parameters, it is not cryptographically secure. At the same time, SHA-2 and SHA-3 demonstrate equally good basic statistical properties. SHA-3 algorithm based on Sponge architecture demonstrates slightly better byte uniformity. Although SHA-256 and SHA3-256 provide the same nominal collision resistance of 2^{128} operations, SHA-3 offers a significantly larger security margin due to its sponge construction with a 512-bit capacity, resulting in an effective margin of 2^{256} operations.

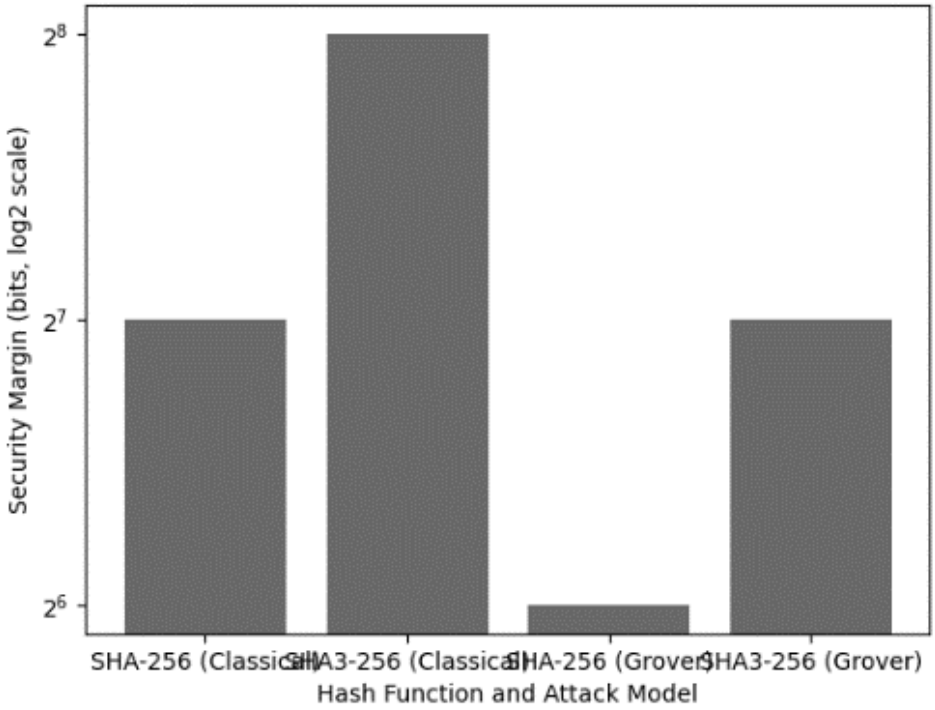

Figure 5 – Security Margin Comparison of SHA-2 and SHA-3 (Classical vs Quantum)

Conclusions. As can be seen from the research, the sponge architecture provides a safety margin of up to 2^{256} operations for collision attacks and completely eliminates the class of length-extension attacks for the Sponge type construction. It is clear that the internal state of the Sponge architecture, with a volume of 1600 bits, provides a theoretical increase in structural cryptographic strength of more than 2^{128} times compared to the minimum required level of security.

At the same time, it can be stated that the security of a hash function is determined not only by the cryptographic strength of its internal compression function and the length of the output value It can be possiblebe also by the architectural scheme of construction, although for a long time this paradigm of improving the security of hash functions prevailed in a number of scientific sources.

These findings, combined with advances in attack methodologies and increasing security demands, led to a paradigm shift toward more flexible hash constructions. At the same time, the question remains about the unambiguity of the correspondence between theoretical statistical indicators of the quality of hash functions. After all, in practical application, the χ^2 test, as a "bias detector", can only prove that the hash function is hack-resistant, but does not assess the quality of the functions.

The results of the evaluation of the Security Margin Comparison of SHA-2 and SHA-3 (Classical vs Quantum) parameter obtained in Figure 2 differ significantly, although they characterize the same indicator. This raises the question of which algorithm (classical brute force or Grover's optimal target algorithm) is more appropriate for estimating the security margin in hash functions. This issue requires further research.

References

1. Wang, X., & Yu, H. (2005). How to break MD5 and other hash functions. In R. Cramer (Ed.), Advances in cryptology – EUROCRYPT 2005 (Lecture Notes in Computer Science, Vol. 3494, pp. 19–35). Springer. https://doi.org/10.1007/11426639_2.
2. Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2015). Keccak. Cryptology ePrint Archive, Paper 2015/389. https://eprint.iacr.org/2015/389.
3. Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2007). Sponge functions [Public comment to NIST]. Ecrypt Hash Workshop. http://www.csrc.nist.gov/pki/HashWorkshop/PublicComments/2007 May.html.
4. Damgård, I. (1989). A design principle for hash functions. In G. Brassard (Ed.), Advances in cryptology – CRYPTO '89 (Lecture Notes in Computer Science, Vol. 435, pp. 416–427). Springer. https://doi.org/10.1007/0-387-34805-0_39.
5. Hamlin, B., & Song, F. (2019). Quantum security of hash functions and property-preservation of iterated hashing. In A. Boldyreva & D. Micciancio (Eds.), Advances in cryptology – CRYPTO 2019 (Lecture Notes in Computer Science, Vol. 11692, pp. 329–349). Springer. https://doi.org/10.1007/978-3-030-25510-7_18.
6. Sahun, A., Nikitenko, Y., Gikalo, P., Panasko, O., & Dudykevych, V. (2025). Method of quick hash functions quality determination. In I. Opirskyy et al. (Eds.), Proceedings of the Cyber Security and Data Protection (CSDP 2025) (CEUR Workshop Proceedings, Vol. 4042, pp. 291–299). CEUR-WS. https://ceur-ws.org/Vol-4042/short2.pdf.
7. Hoch, Jonathan J.; Shamir, Adi (2008). "On the Strength of the Concatenated Hash Combiner when All the Hash Functions Are Weak". Automata, Languages and Programming. Lecture Notes in Computer Science. Vol. 5126. pp. 616–630. doi:10.1007/978-3-540-70583-3_50.
8. Biham, Eli & Dunkelman, Orr. (2007). A framework for iterative hash functions-HAIFA. IACR Cryptology ePrint Archive. 2007. 278. https://eprint.iacr.org/2007/278.
9. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Hash functions and data integrity. In Handbook of applied cryptography (Chap. 9). CRC Press. https://cacr.uwaterloo.ca/hac/about/chap9.pdf.
10. Joux, A. (2004). Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions. In: Franklin, M. (eds) Advances in Cryptology – CRYPTO 2004. CRYPTO 2004. Lecture Notes in Computer Science, vol 3152. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-28628-8_19.
11. Kelsey, John & Kohno, Tadayoshi. (2005). Herding Hash Functions and the Nostradamus Attack.. IACR Cryptology ePrint Archive. 2005. 281.
12. Kelsey, J., & Schneier, B. (2004). Second preimages on n-bit hash functions for much less than 2^n work. Cryptology ePrint Archive, Paper 2004/304. https://eprint.iacr.org/2004/304.
13. Nielsen, M. A., & Chuang, I. L. (2010). Quantum computation and quantum information (10th anniversary ed.). Cambridge University Press. https://profmcruz.wordpress.com/wp-content/uploads/2017/08/quantum-computation-and-quantum-information-nielsen-chuang.pdf.
14. Rivest, R. (1992). The MD5 message-digest algorithm (RFC 1321). Internet Engineering Task Force (IETF). https://doi.org/10.17487/RFC1321
15. Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2011). The Keccak SHA-3 submission. Keccak Team. https://keccak.team/files/Keccak-submission-3.pdf.
16. Greenwood, C., & Nikulin, M. S. (1996). A guide to chi-squared testing. Wiley.

Сагун Андрій Вікторович
кандидат технічних наук, доцент, доцент кафедри комп'ютерних систем, мереж та кібербезпеки,
Національний університет біоресурсів і природокористування України
ORCID: https://orcid.org/0000-0002-5151-9203
E-mail: a.sagun@nubip.edu.ua

ВІД МЕРКЛЕ-ДАМГАРДА ДО SPONGE: ВПЛИВ АРХІТЕКТУРИ НА БЕЗПЕКУ ХЕШ-ФУНКЦІЙ

Анотація. *У статті досліджується вплив архітектури криптографічних хеш-функцій на їхню криптографічну стійкість. Основна увага приділяється порівняльному аналізу класичної архітектури Меркла-Дамгарда, що використовується в сімействі SHA-2, та архітектури Sponge, реалізованої в стандарті SHA-3. Показано, як конструктивні особливості архітектури Sponge, зокрема поділ внутрішнього стану на частини rate та capacity, забезпечують підвищений запас криптографічної стійкості та гарантують низьку вразливість до властивих конструкцій Меркла-Дамгарда, включаючи атаку розширення повідомлення. Підтверджено можливість оцінки індексу дисперсії для віднесення хеш-функції до криптографічного типу. Водночас залишається питання щодо однозначності відповідності між теоретичними статистичними показниками якості хеш-функцій. Єдиний відомий показник якості хеш-функцій базується на показнику дисперсії та однозначно показує лише те, чи належить певна хеш-функція до криптографічних чи некриптографічних. Водночас підтверджено, що χ^2-тест, як «детектор зміщення», може з високою ймовірністю довести, що хеш-функція є стійкою до злому. Але залишається питання щодо однозначності відповідності між теоретичними статистичними показниками якості хеш-функції.*

Ключові слова: *криптографічні хеш-функції, SHA-3, SHA-2, архітектура Меркла-Дамгарда, архітектура Sponge, криптостійкість, запас міцності, постквантова безпека.*