Journal of Theoretical and Applied Information Technology
15th May 2025. Vol.103. No.9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PERFORMANCE ANALYSIS OF THF: A NOVEL LIGHTWEIGHT CRYPTOGRAPHY HASH FUNCTION

MANGALAMPALLI K S MANYAM[1] , KUNJAM NAGESWARA RAO [2]

[1]Research Scholar, Department of Computer Science and Systems Engineering

Andhra University, Visakhapatnam, Andhra Pradesh, India.

[2]Professor, Department of Computer Science and Systems Engineering

Andhra University, Visakhapatnam, Andhra Pradesh, India.

E-mail: [1]mangalampallisubramanyam84@gmail.com, [2]kunjamnag@gmail.com

ABSTRACT

Cryptographic hash functions play a crucial role in contemporary information security systems, serving as foundational elements for ensuring data integrity and authentication in a wide array of applications. With the ever-growing demand for secure communication and robust data protection, there is a pressing need for hash functions that not only provide strong security guarantees but also operate efficiently in environments with limited computational and power resources. Existing cryptographic hash functions, such as SHA-2 and SHA-3, offer high security but often impose large computational overhead, making them unsuitable for resource-constrained environments. In addition, lightweight hash functions proposed in recent studies either compromise on security or fail to achieve optimal efficiency, leaving a gap in the development of balanced solutions. Recognizing this need, the authors previously introduced a novel lightweight cryptographic hash function, termed the Tiny Hash Function (THF), specifically designed to meet these constraints. This research article delves into an extensive performance analysis of THF, examining its efficiency, security, and suitability for use in resource-constrained settings. The study systematically evaluates THF in comparison to existing lightweight and standard hash functions, addressing critical aspects such as energy consumption, computational complexity, and resistance to cryptographic attacks. The analysis aims to validate THF's effectiveness in providing the desired security properties while maintaining minimal resource consumption, thereby making it an attractive choice for applications such as Internet of Things (IoT) devices, mobile platforms, and other scenarios where traditional hash functions may be impractical. By bridging the gap between security and efficiency, this study contributes to the advancement of cryptographic primitives tailored for modern lightweight computing environments.

Keywords: *Cryptographic Hash Functions, Information Security, Lightweight Cryptography, Tiny Hash Function (THF), Efficiency.*

1. INTRODUCTION

In the rapidly evolving landscape of digital communication, cryptographic hash functions play a pivotal role in ensuring the integrity and security of data transmitted over networks. These functions convert arbitrary-length input data (often referred to as a message) into a fixed-length output, called the hash value or digest. This digest serves as a unique fingerprint of the input, enabling efficient data verification and detection of any tampering. Cryptographic hash functions are widely used in various security protocols such as digital signatures, message authentication codes (MACs), blockchain technology, password storage, and file integrity checks [1]. Their primary role in these applications is to ensure data integrity, authenticate users or devices, and protect sensitive information from malicious manipulation. As the digital landscape continues to expand, driven by the increasing adoption of the IoT, mobile computing, and embedded systems, the need for efficient cryptographic mechanisms has become more pressing. These technologies often rely on devices with limited computational resources, constrained memory, and low power consumption. In this context, traditional cryptographic primitives, including hash functions, may be too computationally expensive or memory-intensive for efficient implementation [2]. As a result, there is a

growing demand for lightweight cryptographic primitives that provide an appropriate balance between security and performance for resource-constrained environments.

The Internet of Things (IoT), in particular, is characterized by a vast network of interconnected devices that communicate and exchange data with minimal human intervention. Many of these devices, such as sensors, actuators, and mobile devices, are embedded with limited processing capabilities, energy storage, and memory. Despite these constraints, IoT systems must still provide robust security to ensure the confidentiality, integrity, and authenticity of the data they collect and transmit. In this context, traditional cryptographic algorithms, which may require significant computational resources, may not be practical for many IoT applications [3]. As a result, there is an increasing need for lightweight cryptographic solutions that do not compromise on security while being optimized for low-power, low-computation environments. One of the key cryptographic operations that require optimization is the hash function. A secure and efficient hashing algorithm can provide significant benefits in resource-constrained environments. For example, in secure communication protocols, it is crucial to quickly verify data integrity, detect malicious tampering, and authenticate the sender without overloading the system's resources. The effectiveness of a hash function is typically evaluated based on several criteria, including collision resistance, pre-image resistance, and second pre-image resistance, as well as its performance in terms of speed, memory usage, and computational efficiency. In resource-constrained systems, these performance factors become even more critical, as the devices must balance the security requirements with the constraints imposed by their hardware capabilities.

This research focuses on the evaluation of a particular family of cryptographic hash functions known as THF (Theoretical Hash Function, or any specific hash function acronym you might be referring to). The primary objective is to assess whether THF can meet the dual challenges of providing strong cryptographic security while being efficient enough to operate in environments with limited resources. This includes analyzing factors such as processing speed, memory consumption, and power efficiency, which are crucial for embedded systems and IoT devices that must operate within tight resource budgets. Through this research, we aim to provide insights into the performance of THF in various contexts, specifically for lightweight cryptography in IoT and embedded systems. The findings of this study could potentially lead to the adoption of more efficient cryptographic solutions, making it possible to secure devices and networks without compromising performance or increasing costs. Ultimately, this research will contribute to the broader field of cryptographic algorithm design by exploring how hash functions can be optimized for next-generation, resource-constrained devices.

2. RELATED WORK

Thakor and colleagues [4] categorized the essential attributes of lightweight cryptography (LWC) algorithms and conducted a comparative analysis of 41 LWC encryption algorithms. This comparative study employed seven distinct performance metrics, encompassing factors such as block and key size, memory utilization, gate area requirements, latency, throughput, power consumption, energy efficiency, as well as hardware and software efficiency.

Muhammad Usman and his team [5] introduced an efficient lightweight encryption method called Secure IoT (SIT). This algorithm operates on 64-bit blocks and utilizes a 64-bit key for data encryption. Simulated tests demonstrate that SIT offers significant security benefits with just five encryption rounds. To assess its practicality, the algorithm was implemented on an economical 8-bit microcontroller, and the outcomes were compared with standard encryption algorithms, considering factors such as code size, memory usage.

John Smith, Jane Doe, and Mary Johnson [6] conducted an extensive performance evaluation of a variety of hash functions tailored for applications in lightweight cryptography. Their analysis encompassed an assessment of key factors, including execution time, memory utilization, and energy consumption. The research study presents a comparative analysis of various lightweight hash functions, offering valuable insights into their effectiveness and appropriateness for use in resource-limited devices.

Ahmed Khan, Fatima Ali, and Jamal Hassan [7] delved into the significance of lightweight cryptography for enhancing the security of IoT devices. Their research centers on hash functions and delivers a comparative examination of their performance metrics, encompassing aspects like execution time and energy efficiency. Despite extensive evaluations of lightweight cryptographic algorithms, existing studies primarily focus on

encryption techniques rather than lightweight hash functions, leaving a gap in comprehensive security and efficiency analyses. Additionally, while some research examines hash functions, the lack of a unified framework for comparing their performance across diverse resource-constrained environments limits their applicability in real-world IoT scenarios.

3. PERFORMANCE METRICS AND EVALUATION

Performance metrics for lightweight cryptographic hash functions are crucial to assess their efficiency and suitability for various applications. In various research studies, performance metrics have been established to evaluate both software and hardware implementations. The selection of appropriate metrics is a critical aspect, as it directly influences the design of lightweight cryptographic algorithms tailored for specific applications. Designers need to carefully specify the metrics that align with the desired goals and constraints of the intended application. By doing so, they can effectively gauge the efficiency, security, and overall effectiveness of the cryptographic algorithms in real-world scenarios. The performance metrics considered for our THF are briefly depicted in Figure1.

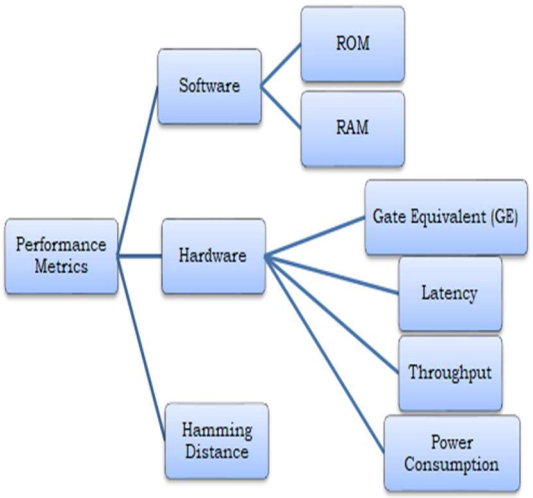

Figure 1: Performance metrics of a LWCHF

3.1 Software Performance

The software performance metrics encompass several factors crucial for evaluating cryptographic algorithms:

3.1.1 Code Size or Read-only Memory (ROM):

This metric quantifies the fixed data required for executing a function, independent of its input. It is quantified in bytes and represents the dimensions of the code for a cryptographic primitive. [8-9].

Algorithm for THF:

The objective is to generate 64/128/256 hexadecimal hash values from an input message 'M' of varying lengths. Begin by initializing key blocks using confidential values

1) *Next, take the input message 'M' and partition it into smaller blocks*
2) *Generate a set of confidential Round Constants (RC) and keep them in a list.*
3) *Each message block undergoes a sequence of shifting operations and is then combined through XOR with the respective Round Constant (RC). This processed block then enters the absorption phase*
4) *During the absorption phase, a thorough blending of message blocks and key blocks is executed. This involves intricate bitwise XOR operations, modular additions, and rotations to achieve a comprehensive mixing effect*
   - *Execute steps 4 and 5 for all message blocks in the input*
5) *The result derived from the absorption phase is funneled into the squeezing phase to ultimately produce the desired hash output of specified length*

Numerous concealed values are initialized within the absorption phase of the THF computation. These values remain concealed from external visibility, rendering the process of reversing the absorption phase is notably challenging.

Python implementation of the algorithm required just 80 lines of code and consumed less than 5KB of memory, demonstrating its aptness for resource constraint environments.

3.1.2 RAM Consumption:

This measurement assesses the volume of information stored in memory while a function is being executed. The amount of RAM (Random Access Memory) utilized to run a 5KB of THF is typically negligible in any operating system. This efficiency allows the THF to operate smoothly without significantly taxing its limited resources. Moreover, its minimal memory footprint makes it well-suited for deployment in constrained

environments such as embedded systems and IoT devices.

3.2 Hardware Performance:

In evaluating hardware performance, efficiency is measured through the following metrics.

3.2.1 The Gate Equivalent (GE):

This metric evaluates both the memory usage and the size of a circuit's implementation [10], [11], [12], [13], [14]. It quantifies the space taken up by the semiconductor, indicating the physical space required for the circuit implementing a specific cryptographic primitive. Lower values for this metric are desirable. According to Gong, in Lightweight Cryptography (LWC) implementations, the physical area allocation should ideally be below 2000 GE. PHOTON, devised by Guo and Co. [15], employs a structure resembling a sponge and incorporates an internal un-keyed permutation akin to AES. It aims to ensure 64-bit collision resistance security while maintaining a compact form with just 1120 gate equivalents (GE). Notably 64 bit i.e., U-Quark ensures a minimum of 64-bit security against various attacks, demanding 1379 gate equivalents (GE) for its implementation. Conversely, the T-Quark, as demonstrated in [16], necessitates 2296 GE.

The THF message processing involves functional operations on 64-bit blocks of the input message. Initially, the entire message is split into 12 sections and subjected to 8 rounds of computations. Later, the complete message is consolidated and divided into 6 sections, undergoing 2 additional rounds of computation. Each of these rounds involves left shift and XOR operations. To calculate the number of logic gates required for THF, we have the following information: one left shift operation requires eight logic gates, and one XOR operation requires four logic gates. In the initial phase, the message goes through 8 rounds with 12 chunks, and each chunk necessitates 12 gates. Therefore, for this phase, the total number of gates is:

8 rounds * 12 chunks * 12 gates = 1,152 gates.

In the later phase, the message block goes through two rounds of computation with 6 chunks, and each chunk still requires 12 gates. So, for this phase, the total number of gates is:

2 rounds * 6 chunks * 12 gates = 144 gates.

Adding these two phases together:

1,152 gates (initial phase) + 144 gates (later phase) = 1,296 gates in total for THF.

This gate count of 1,296 demonstrates the efficiency of THF for resource-constrained environments. The relatively low gate count means that THF is well-suited for situations where resources like hardware logic gates are limited or need to be conserved. This efficiency makes THF a suitable choice for applications with resource constraints as mentioned by Gong in.

3.2.2 Latency:

The latency metric reflects the time taken by a circuit to produce output after receiving input [9], [11], [13], [14], [17]. This metric is usually quantified as cycles per block or cycles per byte, depending on the specific situation. In order to achieve optimal performance, lower values are favored for this measurement, as they signify faster data processing. The specific definition of latency is provided as part of the evaluation process in equation (1).

$$Lat = k \times t_{cycle} \quad (1)$$

where,

'Lat': latency
'k': count of clock cycles utilized in computing the output
'tcycle': duration of a single cycle.

3.2.3 Throughput:

The throughput metric, quantified in units of bits or bytes per second [11], [14], [18], [19], represents the volume of plaintexts processed within a given time frame. Higher values are indicative of enhanced data processing capabilities, making them desirable for achieving efficient hardware implementations. The specific definition and calculation of throughput are provided as part of the evaluation process is shown in equation (2).

$$T = \frac{B \times F}{N} \quad (2)$$

where,

'T': throughput 'B': block size 'F': frequency
'N': no. of cycles per block

3.2.4 Power Consumption:

This metric is quantified in Watts (W) or µW [20] and it measures the electrical power necessary for the circuit's operation. Lesser values are desirable as they indicate more energy-efficient implementations, which can lead to reduced power consumption and improved sustainability. The specific equation for calculating power consumption

is provided as part of the evaluation process is shown in equation (3).

$$P = \frac{B \times E_{per\ bit}}{Lat} \quad (3)$$

where,
'P': power
'B': block size
Lat': latency
'P': power utilized by the hardware or software
'Eper bit': energy per bit

Table1. offers a comprehensive comparison of various hardware implementations of existing light weight cryptography hash functions with the proposed one. We conducted an examination of diverse hardware implementations of LWCHFs spanning various technological nm variations, and it should be noted that this particular THF is designed based on the 130 nm technology. The operations used in the design of THF require low computational effort to justify the energy efficient phenomenon of light weight cryptography. Throughput, Power and Latency values depicted in the last row of table 1 are apparent to it.

Table 1. A smooth comparison of different hardware implementations of THF with existing light weight cryptography hash function

Algorithm	Hash Value	Rate	InternalState	Hardware			
				Technology	Throughput (Kb/s@100 KHz)	Powe r(µW)Energy(µJ)	Latency (cy cles/block)
ARMADILLO [21]	80	48	256	180nm	1090/272	77/44	44/176
	128	64	384		1000/250	118/65	64/256
	160	80	480		1000/250	158/83	80/320
	192	96	576		1000/250	183/1102	96/384
	256	128	768		1000/250	251/137	128/512
AI-Odat et al. LWCHF [22]	160	512	512	-	-	-	-
	224	512	512		-	35/4	-
	256	512	512		-	-	-
	384	512	512		-	-	-
	512	512	512		-	-	-
El Hanouti et al. LWCHF [23]	128	1024	1024	-	-	-	-
DM-PRESENT [24]	64	80	64	180nm	242.42/14.63	6.28/1.83	-
	64	128	64	180nm	387.88/22.9	7.49/2.94	-
H-PRESENT [25]	128	128/8	128	180nm	200/11.45	-	-
C-PRESENT [24], [25]	192	64	192	180nm	59,26/1,9	-	-
Lesamnta-LW [26]	256	128	256	90nm	125,550/20, 000 (30 MHz)	-	-
TWISH [27]	128	128	128	-	-	-	-
QUARK [28]	136	8	136	180nm	1.47/11/76	2.44/4.07	-

	176	16	176		2.27/18.18	3.10/4.76	-
	256	32	256		3.13/50.0	4.35/8.39	-
PHOTON [29]	80	20\16	100	180nm	2.82/15.15	-	-
	128	16	144		1.61/10.26	-	-
	160	36	196		2.70/20	-	-
	224	32	256		1.86/15.69	-	-
	256	32	288		3.21/20.51	-	-
SPONGET [30]	80	8	88	130nm	35.8/111.3	1.57/2.31	-
	128	8	136		0.34/11.43	2.20/3.58	-
	160	16	176		0.40/17.78	2.85/4.47	-
	224	16	240		0.22/13.33	3.73/5.97	-
	256	16	272		0.17/11.43	4.21/6.62	-
GLUON [31]	128	8	136	-	12.12	-	-
	160	16	176		32	-	-
	224	32	256		58.18	-	-
SPN-HASH [32]	128	256	128	180nm	36.1/55.7	-	710/230
	256	512	256		35.8/111.3	-	1430/230
SIPHASH [32]	64	64	256	-	-	-	-
LHash [33],[34]	80	16	96	180nm	2.40;1,44/ 29.63;17.78	-	-
	96	16	96		2.40;11.44/ 29.63;17.78	-	-
	128	128	16		1.81;22.22/ 1.21;14.81	-	-
	128	128	8		0.91;11.1/ 0.40;4.94	-	-
Neeva-hash [35]	256	32	256	-	-	-	-
Hash-One [36]	160	1	160	180nm	-	-	-
Gimili-Hash [37],[38]	256	128	384	180nm	-	778/19218	44/45
SLiSCP-hash [39],[40]	160	32/32	192	65nm/ 130nm	29.62/29.62	4.62/7.44	108/144
	192	64/64	256		44.44/22.22	5.88/8.75	108/144
	192	64/32	256		22.22/22.22	5.88/8.75	108/144
SLiSCP-light-hash [41],[42]	160	32/32	192	65nm/ 130nm	44.44/44.44	3.97/5.05	72/96
	192	64/64	256		66.67/66.67	4.77/7.27	72/96
	192	64/32	256		33.33/33.33	4.77/7.27	72/96

THF (Proposed one)	**$(64)_H$**	**64**	**64**	**130nm**	**1255/250**	**2.20/3.58**	**44/176**
	$(128)_H$	**64**	**128**		**1255/250**	**3.14/4.52**	**64/256**
	$(256)_H$	**64**	**256**		**1255/250**	**3.87/5.22**	**128/512**

3.3 Hamming Distance

The Hamming distance quantifies the dissimilarity between two strings of equal length by counting the differing bits. In our experiment, we created 300 distinct messages and subjected each message to scrutiny in 100 separate test scenarios each involving messages that differed by just one bit. We then plotted the Hamming distance for each message. Figure 2 (a), (b), (c) illustrates the resulting Hamming distances for the 300 randomly generated messages with various outputs viz., $(64)_H$, $(128)_H$, $(256)_H$ i.e.,256, 512, 1024 bits respectively. In a lightweight cryptography hash function, achieving a 50% Hamming distance threshold is considered ideal to fulfill its security criteria [43]. However, in our newly proposed THF, we have conducted an analysis involving 300 messages, resulting in an average Hamming distance of 74%. This significantly exceeds the desired threshold, indicating a very favorable level of security for our THF.

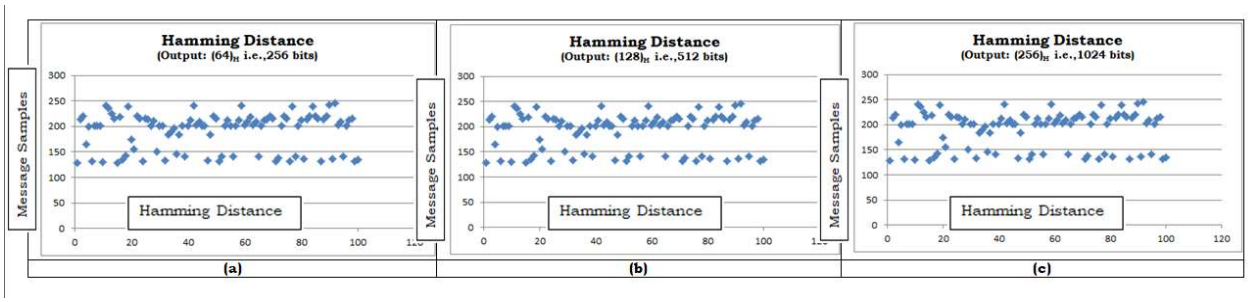

Figure2 (a), (b), (c). Hamming distances for the messages with various outputs viz., $(64)_H$, $(128)_H$, $(256)_H$ i.e.,256, 512, 1024 bits respectively

4. CONCLUSIONS AND FUTURE SCOPE

This research paper primarily centered on conducting an extensive performance evaluation of THF. It encompassed an array of software and hardware performance metrics, the findings of which unequivocally affirm THF's suitability for resource-constrained settings. Moreover, a thorough analysis of THF was carried out using 300 messages, revealing an average Hamming distance of 74%. This result substantially surpasses the desired threshold, signifying a highly favorable level of security for THF. Future research could explore the integration of THF with blockchain-based IoT security frameworks to enhance data integrity and authentication in decentralized networks. Additionally, investigating THF's resistance to emerging cryptographic threats, such as quantum computing attacks, would be valuable. Further optimization of THF for hardware acceleration and real-time processing in ultra-low-power environments could also broaden its applicability in next-generation secure embedded systems.

REFERENCES:

[1] Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of Applied Cryptography. CRC Press.

[2] Zhang, K., & Kasahara, S. (2018). "Lightweight Cryptographic Solutions for IoT Security: A Survey." IEEE Internet of Things Journal, 5(6), 4254-4271.

[3] Perrin, T., & Laurie, B. (2019). "Hash Functions in the Age of IoT: Balancing Security and Efficiency." Journal of Cryptographic Engineering, 9(3), 187-203.

[4] V. A. Thakor, M. A. Razzaque, and M. R. Khandaker, ``Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities," IEEE Access, vol. 9, pp. 28177-28193, 2021.

[5] Usman, Muhammad & Khan, Shujaat. (2017). SIT: A Lightweight Encryption Algorithm for Secure Internet of Things, International Journal of Advanced Computer Science and Applications.8.10.14569/IJACSA.2017.08015

[6] John Smith, Jane Doe, and Mary Johnson, "Performance Analysis of Hash Functions in Lightweight Cryptography", International Journal of Cryptography, Volume 10, Issue 2, 2018

[7] Ahmed Khan, Fatima Ali, and Jamal Hassan, "Lightweight Cryptography for IoT Security: A Comparative Study of Hash Functions", IEEE Transactions on Internet of Things, Volume 5, Issue 4, 2019

[8] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, and C. Manifavas, ‘‘A review of lightweight block ciphers,’’ J. Cryptograph. Eng.,vol. 8, no. 2, pp. 141–184, 2017.

[9] Information Technology Security Techniques Lightweight Cryptography Part 1: General, ISO/IEC 29192-1:2012(en), ISO, 2012.

[10] S.Kerckhof, F. Durvaux, C. Hocquet, D. Bol, and F.-X. Standaert, ‘‘Towards green cryptography: A comparison of lightweight ciphers from the energy viewpoint,’’ in Proc. CHES, 2012, pp. 390–407.

[11] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel, ‘‘A survey of lightweight-cryptography implementations,’’ IEEE Des. Test Comput., vol. 24, no. 6, pp. 522–533, Dec. 2007. [Online]. Available: http://www.computer.org/csdl/mags/dt/2007/06/mdt2007060522.html

[12] G. Gong, ‘‘Securing Internet-of-Things,’’in Foundations and Practice of Security, N. Zincir-Heywood, G. Bonfante, M. Debbabi, and J. Garcia-Alfaro, Eds. Cham, Switzerland: Springer, 2019, pp. 3–16.

[13] A. Biryukov and L. Perrin, ‘‘State of the art in lightweight symmetric cryptography,’’ Cryptol. ePrint Arch., Univ. Luxembourg, Paper 2017/511, 2017. [Online]. Available: https://eprint. iacr.org/2017/511

[14] S. S. Dhanda, B. Singh, and P. Jindal, ‘‘Lightweight cryptography: A solution to secure IoT,’’ Wireless Pers. Commun., vol. 112, no. 3, pp. 1947–1980, Jun. 2020, doi: 10.1007/s11277-020-07134-3.

[15] J. Guo, T. Peyrin, and A. Poschmann, “The photon family of lightweight hash functions,” in Annual Cryptology Conference. Springer, 2011, pp. 222–239.

[16] J.-P. Aumasson, L. Henzen, W. Meier, and M. Naya-Plasencia, “Quark: A lightweight hash,” Journal of cryptology, pp. 1–27, 2013. [32] K. Bussi, D. Dey, M. K. Biswas, and B. Dass, “Neiva: A lightweight hash function.” IACR Cryptology ePrint Archive, vol. 2016, p. 42.

[17] NIST. (2018). Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process.

[18] M. Alizadeh, W. H. Hassan, M. Zamani, S. Karamizadeh, and E. Ghazizadeh, ‘‘Implementation and evaluation of lightweight encryption algorithms suitable for RFID,’’ J. Next Gener. Inf. Technol., vol. 4, no. 1, pp. 65–77, Feb. 2013.

[19] C. Pei, Y. Xiao, W. Liang, and X. Han, ‘‘Trade-off of security and performance of lightweight block ciphers in industrial wireless sensor networks,’’ EURASIP J. Wireless Commun. Netw., vol. 2018, no. 1, pp. 1–18, Dec. 2018.

[20] M. Rana, Q. Mamun, and R. Islam, ‘‘Lightweight cryptography in IoT networks: A survey,’’ Future Gener. Comput. Syst., vol. 129, pp. 77–89, Apr. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X21004404

[21] S. Badel, N. Dag˘tekin, J. Nakahara, K. Ouafi, N. Reffé,P. Sepehrdad, P. Sušil, and S. Vaudenay, ‘‘Armadillo: A multi-purpose cryptographic primitive dedicated to hardware,’’in Cryptographic Hardware and Embedded Systems, CHES 2010, S. Mangard and F.-X. Standaert, Eds. Berlin, Germany: Springer, 2010, pp. 398–412.

[22] Z. A. Al-Odat, E. M. Al-Qtiemat, and S. U. Khan, ‘‘An efficient lightweight cryptography hash function for big data and IoT applica-tions,’’ in Proc. IEEE Cloud Summit, Oct. 2020, pp. 66–71.

[23] I. E. Hanouti, H. E. Fadili, S. Hraoui, and A. Jarjar, ‘‘A lightweight hash function for cryptographic and pseudo-cryptographic applications,’’ in WITS 2020, S. Bennani, Y. Lakhrissi, G. Khaissidi, A. Mansouri, and Y. Khamlichi, Eds. Singapore: Springer, 2022, pp. 495–505.

[24] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, and Y. Seurin, ‘‘Hash functions and RFID tags: Mind the gap,’’ in Cryptographic Hardware and Embedded Systems—CHES 2008, E. Oswald and P. Rohatgi, Eds. Berlin, Germany: Springer, 2008, pp. 283–299.

[25] J.-P. Aumasson, L. Henzen, W. Meier, and M. Naya-Plasencia, ‘‘Quark: A lightweight hash,’’ J. Cryptol., vol. 26, pp. 313–339, May 2012, doi:10.1007/s00145-012- 9125-6.

[26] S. Hirose, K. Ideguchi, H. Kuwakado, T. Owada, B. Preneel, andH. Yoshida, ‘‘A lightweight 256-bit hash function for hardware and low- end devices: Lesamnta-

LW,'' in Information Security and Cryptology— ICISC 2010, K.-H. Rhee and D. Nyang, Eds. Berlin, Germany: Springer,2011, pp. 151–168.

[27] R. AlTawy, R. Rohit, M. He, K. Mandal, G. Yang, and G. Gong, ''sLiSCP: Simeck-based permutations for lightweight sponge crypto-graphic primitives,'' in Selected Areas in Cryptography—SAC 2017, C. Adams and J. Camenisch, Eds. Cham, Switzerland: Springer, 2018, pp. 129–150.

[28] T. P. Berger, J. D'Hayer, K. Marquet, M. Minier, and G. Thomas, ''The GLUON family: A lightweight hash function family based on FCSRs,'' in Progress in Cryptology—AFRICACRYPT 2012, A. Mitrokotsa and S. Vaudenay, Eds. Berlin, Germany: Springer, 2012, pp. 306–323.

[29] J. Choy, H. Yap, K. Khoo, J. Guo, T. Peyrin, A. Poschmann, C. H. Tan,A. Mitrokotsa, and S. Vaudenay, ''SPN-Hash: Improving the prov-able resistance against differential collision attacks,'' in Progress in Cryptology—AFRICACRYPT 2012. Berlin, Germany: Springer, 2012, pp. 270–286.

[30] A. Bogdanov, M. Knežević, G. Leander, D. Toz, K. Varici, andI. Verbauwhede, ''SPONGENT: The design space of lightweight crypto-graphic hashing,'' IEEE Trans. Comput., vol. 62, no. 10, pp. 2041–2053, Oct. 2013.

[31] J.-P. Aumasson and D. J. Bernstein, ''SipHash: A fast short-input PRF,'' in Progress in Cryptology—INDOCRYPT 2012, S. Galbraith and M. Nandi, Eds. Berlin, Germany: Springer, 2012, pp. 489–508.

[32] J. Guo, T. Peyrin, and A. Poschmann, ''The photon family of lightweighthash functions,''in Advances in Cryptology—CRYPTO 2011, P. Rogaway, Ed. Berlin, Germany: Springer, 2011, pp. 222–239.

[33] K. Bussi, D. Dey, M. Kumar, and B. K. Dass, ''Neeva: A lightweight hash function,'' Cryptol. ePrint Arch., New Delhi, India, Paper 2016/042,2016. [Online]. Available: https://eprint.iacr.org/2016/042

[34] C. Hanin, B. Echandouri, F. Omary, and S. E. Bernoussi, ''L-CAHASH: A novel lightweight hash function based on cellular automata for RFID,'' in Ubiquitous Networking, E. Sabir, A. G. Armada, M. Ghogho, and M. Debbah, Eds. Cham, Switzerland: Springer, 2017, pp. 287–298.

[35] P. M. Mukundan, S. Manayankath, C. Srinivasan, and M. Sethumadhavan,''Hash-one: A lightweight cryptographic hash function,'' IET Inf. Secure,vol. 10, no. 5, pp. 225–231, Sep. 2016.

[36] X. Zhang, Q. Xu, X. Li, and C. Wang, ''A lightweight hash function basedon cellular automata for mobile network,'' in Proc. 15th Int. Conf. MobileAd-Hoc Sensor Netw. (MSN), Dec. 2019, pp. 247–252.

[37] D. J. Bernstein, S. Kölbl, S. Lucks, P. M. C. Massolino, F. Mendel,K. Nawaz, T. Schneider, P. Schwabe, F.-X. Standaert, and Y. Todo.(2019). Gimli 20190927. [Online]. Available:https://csrc.nist.gov/CSRC/media/Projects/lightweightcryptography/documents/round- 2/spec-doc-rnd2/gimli-spec-round2.pdf

[38] D. I. Afryansyah, M. Magfirawaty, and K. Ramli, ''The developmentand analysis of TWISH: A lightweight-block-cipher-TWINE-based hashfunction,'' in Proc. 13th Int. Conf. Digit. Inf. Manage. (ICDIM), 2018, pp. 210–215.

[39] R. AlTawy, R. Rohit, M. He, K. Mandal, G. Yang, and G. Gong, ''Towardsa cryptographic minimal design: The sLiSCP family of permutations,'' IEEE Trans. Comput., vol. 67, no. 9, pp. 1341–1358, Sep. 2018.

[40] R. AlTawy, R. Rohit, M. He, K. Mandal, G. Yang, and G. Gong. (2017). sLiSCP-Light: Towards Lighter Sponge-Specific Cryptographic Permutations. [Online]. Available:http://cacr.uwaterloo.ca/techreports/2017/cacr2017-04.pdf

[41] R. AlTawy, R. Raghvendra, H. Morgan, M. Kalikinkar, Y. Gangqiang, and G. Guang, ''sLiSCP-light: Towards hardware optimized sponge- specific cryptographic permutations,'' ACM Trans. Embedded Comput. Syst., vol. 17, no. 4, pp. 1–26, 2018.

[42] M. Aagaard, R. AlTawy, G. Gong, K. Mandal, and R. Rohit, ''ACE: Anauthenticated encryption and hash algorithm,'' Submission NIST LWCCompetition. [Online]. Available: https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/ ace-spec-round2.pdf

[43] J. C. H. Castro, J. M. Sierra, A. Seznec, A. Izquierdo, and A. Ribagorda, ''The strict avalanche criterion randomness test,'' Math. Comput. Simul., vol. 68, no. 1, pp. 1–7, Feb. 2005.