Design and Implementation of a Lightweight PHOTON Hash Function on FPGA platform

Raghavendra A[1], Sai Kiran Oruganti[2]

[1] Postdoctoral Researcher, Lincoln University College, Malaysia;

[2] Professor, Lincoln University College, Malaysia;

Email ID: pdf.drRaghavendra@lincoln.edu.my, saisharma@lincoln.edu.my.

Abstract: Devices with limited resources, such RFID systems, embedded platforms, and Internet of Things nodes, must be secured using lightweight cryptographic primitives. For low-area FPGA deployments, traditional hash functions and integrated encryption-signature techniques are inappropriate due to their substantial computational and hardware overhead. In this study, we design and implement the PHOTON-80/20/16 lightweight hash function using Verilog-HDL on an Artix-7 FPGA. The architecture operates on a 5x5 state matrix over GF (2^4) and is based on the sponge construction with an AES-like substitution-permutation network. In order to effectively balance throughput and hardware utilization, a round-based iterative architecture with a 100-bit datapath is used. For the SubCells and MixColumns transformations, LUT-based optimization is used to minimize computing complexity. At a maximum frequency of 347.403 MHz, the suggested implementation uses 125 slices and achieves a throughput of 2316.02 Mbps with a latency of 15 clock cycles. A comparison with current PHOTON-80/20/16 architectures shows better throughput-to-area efficiency. The outcomes validate that the suggested design offers a dependable, low-power hashing solution appropriate for embedded security and secure authentication based on FPGA.

Keywords: IoT; PHOTON hash; Lightweight Cryptography; Hardware platform;

Introduction

Confidentiality and authentication are essential for reliable interaction. These needs are typically met by combining digital signatures and encryption using "encrypt-then-sign" or "sign-then-encrypt" paradigms. Nevertheless, these sequential methods come with a significant communication and processing expense. Zheng [1] developed the idea of signcryption in 1997 as a solution to this inefficiency, allowing concurrent encryption and authentication inside a single logical action. Compared to conventional cryptographic designs, signcryption drastically lowers processing complexity and transmit overhead, making it a practical alternative for protected data transfer and device authentication.

Applications like key management, e-administration, and online shopping frequently use authentication techniques. Authentication may be based on biometric characteristics (like fingerprints), knowledge-based data (like passwords), or user-owned privileges (like smart cards), regardless of the verification model. These systems fall into one of three categories: multifactor, two-factor, or one-factor authentication. Secure identification and encrypted transmission are becoming essential due to the

quick spread of lightweight devices like RFID tags and Internet of Things (IoT) nodes. However, because of their low processing power, these restricted platforms are extremely vulnerable to information tampering and leaking. Conventional cryptographic hash functions' high space and power needs make them frequently inappropriate for these kinds of situations. This has led to the development of a number of lightweight hashing and authentication methods [2–3], their implementation in hardware and software architectures [4–6], and their cryptanalysis research [7–8].

The basic goal of ultralight hash function construction is to maximize area-performance trade-offs while preserving reasonable security guarantees. While some methods focus on software efficiency, others are hardware-oriented, and some attain balanced achievement in both areas. Despite having restricted logic and memory capacity, limited devices nonetheless require real-time processing, which calls for properly optimized architectures. Guo et al. [9] presented a thin hash function PHOTON in this context, which combines software performance with hardware simplicity. Although its datapath width and state parameters are different, its internal permutation is identical to the LED block cipher [10]. PHOTON's AES-like permutation structure, which was inherited from LED, allows for efficient and small hardware realization [11], which makes it a good option for ultralight safety features based on FPGA.

Problem Statement: The need for reliable and efficient authentication methods has increased due to the quick spread of IoT gadgets, RFID technologies, and embedded solutions. Traditional hash functions and hybrid encryption-signature methods are examples of standard cryptographic primitives that offer robust security guarantees but have substantial computational, space, and power expenses that make them inappropriate for hardware with limited resources. Real-time data processing, secure connectivity, and robust authentication must all be supported by lightweight devices that adhere to stringent logic, memory, and energy usage guidelines. Even with the introduction of lightweight hash functions like PHOTON, it is still very difficult to balance area, throughput, latency, and power on FPGA devices. For realistic IoT deployments, a lightweight hash function that lowers resource consumption while retaining good performance and security strength requires an optimized hardware implementation.

Contribution: The PHOTON-80/20/16 lightweight hash function is designed and implemented on FPGA utilizing a round-based iterative architecture in this research. The suggested solution achieves hardware compactness while preserving strong diffusion and nonlinearity properties by utilizing the sponge architecture and AES-like permutation structure. Throughput and space utilization are balanced using a 100-bit datapath architecture, and SubCells and MixColumns activities across GF(2^4) are realized efficiently using LUT-based optimization approaches. A thorough performance evaluation is conducted in terms of slice utilization, LUT count, operating frequency, latency, throughput, power usage, and efficiency after the implementation is synthesized on an Artix-7 FPGA using Verilog-HDL. A comparison with current PHOTON-80/20/16 implementations shows that throughput-to-area efficiency has increased while resource use has remained reasonable. The suggested architecture is appropriate for secure embedded applications and lightweight authentication since it strikes a good balance between hardware footprint and efficiency.

Related Works

Significant advancements have been made in lightweight cryptographic fundamentals designed for IoT

contexts with limited resources, as evidenced by recent literature. In comparison to standalone deployments, Al-Shatari et al. [12] introduced a lightweight authenticated encryption architecture that integrates the PHOTON hash function with the LED block cipher, resulting in a 14% reduction in logic area and a 46.04% reduction in power usage. In comparison to traditional DRBG methods, Ngo et al.'s TRNG design [13] lowered power usage by 65% and area utilization by five times by utilizing ring oscillator jitter in conjunction with PHOTON-based post-processing. Similar to this, HashLEA was found to be a safe and effective option with strong avalanche properties and a even distribution by Sevin and Mohammed [14], who assessed lightweight hash functions for blockchain-powered IoT.

Sophisticated cryptographic frameworks have been investigated in parallel for safe authentication and transmission. In order to guarantee privacy, exclusivity, and defense against impersonating and eavesdropping assaults, Prajapat et al. [15] suggested a quantum e-voting technique that makes use of no-cloning and quantum key sharing features. A hyperelliptic curve-based signcryption system was created by Khan et al. [16], which was appropriate for devices with limited resources because it decreased computational expenses by 38.16% and communication expenses by 62.5%. Furthermore, a novel lightweight hash architecture was presented by Sevin and Çavuşoğlu [17] with the goal of reducing computational difficulty, processing time, and energy usage for IoT devices.

Parallelism and newly developed safe interaction mechanisms are also highlighted in recent publications. In order to improve server-side throughput, Choi et al. [18] used GPU and AVX-512 parallel frameworks to optimize the Korean Compact Secure Hash (LSH). A tree-based compact hash design that satisfies security criteria and improves efficiency was presented by Sevin et al. [19] for data-parallel applications. Additionally, Zhang et al. [20] created stable device IDs in unpredictable wireless contexts by combining locally sensitive hashing with channel status data. In contrast to traditional QKD methods, Solaiman et al. [21] introduced a combination quantum key distribution architecture that combines atomic and photonic technologies to achieve enhanced security against eavesdropping.

Method

A lightweight cryptographic primitive, the PHOTON hash function was created especially to accomplish low computational complexity and low power usage while preserving sufficient security strength for devices with limited resources. As seen in Figure 1, its architecture adheres to the sponge construction approach developed by Bertoni et al. The two different functions of the sponge structure are squeezing and absorbing. A predetermined permutation function f is applied after the input message m is divided into blocks of size r and XORed into the internal state during the absorbing phase. The squeezing phase starts once the full message has been absorbed. In this phase, output blocks of size r' are taken out of the state, and the permutation is applied again between extractions until the necessary hash length is produced. Five variations of the PHOTON family are represented by the notation PHOTON- $n/r/r'$, where n is the hash output size, r and r' is the input and output rate, respectively.

Table 1 summarizes the trade-offs between security level, performance, and logic utilization. The PHOTON-80/20/16 version is taken into consideration in this work. This design uses an 80-bit hash digest, a 100-bit internal state, a 20-bit input rate, and a 16-bit output rate, as shown in Table 1. The 80-bit capacity guarantees an 80-bit level of security against common assaults. Each cell in the state is 4

bits in size ($s = 4$), forming a 5×5 matrix (dimension $d = 5$). Thus, the irreducible polynomial $x^4 + x + 1$ defines the finite field $GF(2^4)$ over which the permutation actions. There are twelve rounds in the permutation ($N_r = 12$).

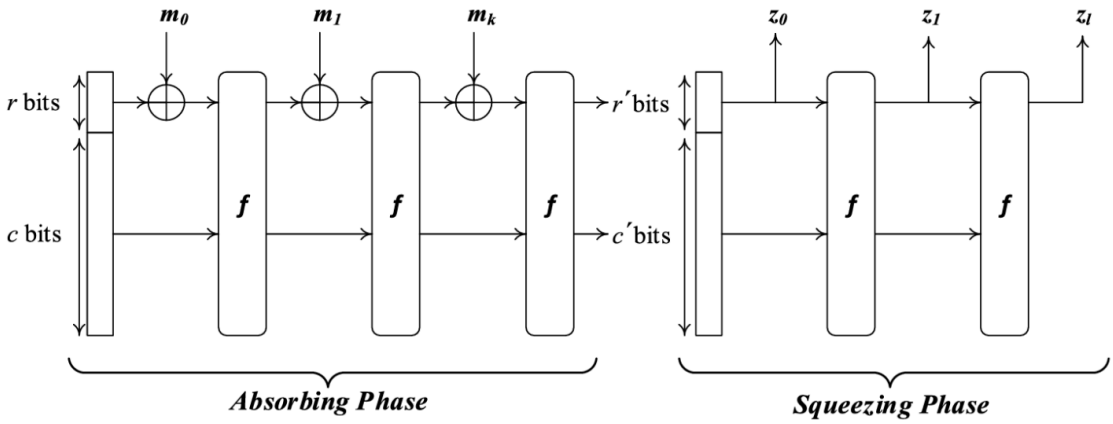

Figure 1. PHOTON sponge construction

Table 1. Specifications of PHOTON -80/20/16

Parameters	**Photon-80/20/16**
State Size t (bit)	100
Hash Digest n (bit)	80
Input Rate r (bit)	20
Output Rate r´ (bit)	16
Capacity c (bit)	80
Cell Size s (bit)	4
Matrix Size d (cell)	5
Rounds N_r	12

Figure 2 shows the internal permutation function f, which uses a Substitution–Permutation (SP) network topology similar to AES but modified for lightweight implementation. Four transformations are performed in each round: AddConstants (AC), SubCells (SC), ShiftRows (SR), and MixColumns (MC). Only the first column of the state matrix is altered by the AddConstants functioning, which XORs two constants: an internal constant (IC_d) and a round constant (RC). Table 2 shows that the round constant is a 4-bit number that changes with each round and is produced using an LFSR sequence.

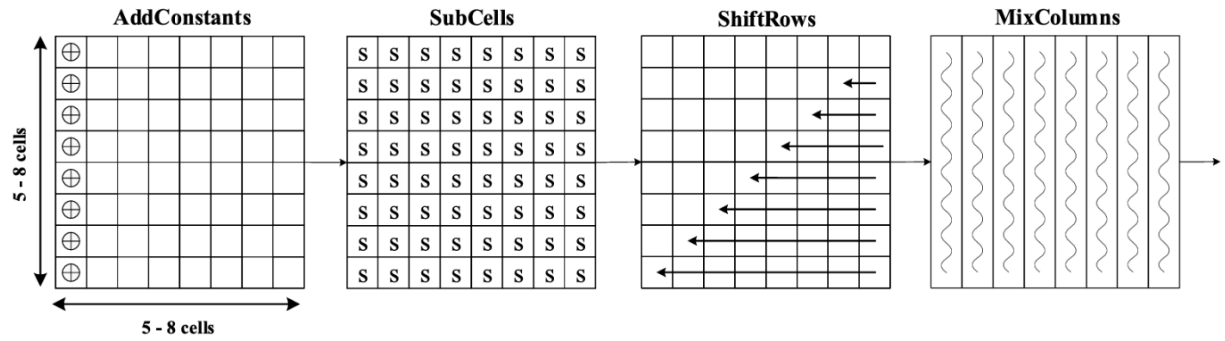

Figure 2. PHOTON permutation

The PHOTON-80/20/16 hardware design is shown in Figure 3. The input message M is XORed with the state initialized by a predetermined initialization vector (IV), and the resultant state is processed through the round transformation blocks. In this approach, padding is not used and only one 20-bit input block is taken into consideration for simplicity. To produce the 80-bit hash output h, one absorbing process is followed by squeezing processes; each 16-bit output necessitates a further 12-round permutation.

$$IV_{100} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 4\ 1\ 4\ 1\ 0\ ] \quad (1)$$

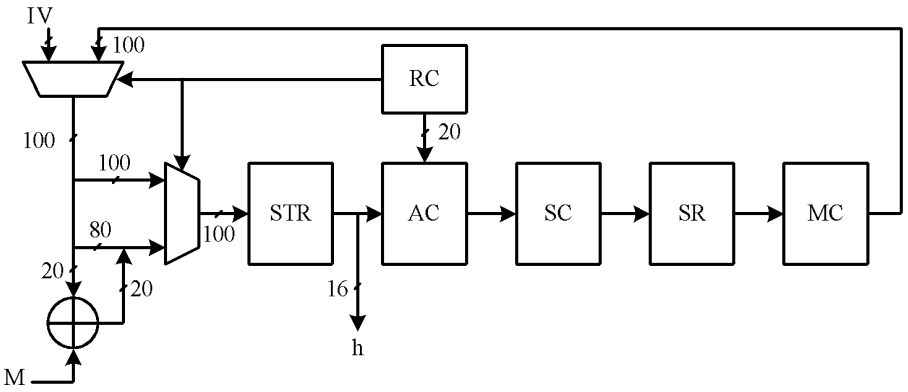

Figure 3. Hardware Architecture of PHOTON-80/20/16

The definition of the internal constant for PHOTON-80/20/16 is $IC_d = [0, 1, 3, 6, 4]$, which is dependent on the matrix dimension. These constants stop structural assaults and create asymmetry (Table 2). The PRESENT cipher S-box is used in the SubCells transformation, which independently performs a nonlinear substitution to each 4-bit cell of the state matrix. By mapping each input bit $x \in \{0, \ldots, F\}$ to its matching $S[x]$, the substitution numbers are presented in Table 3, which increases nonlinearity and confusion.

Table 2. Round dependents for PHOTON-80/20/16

Row \ NR	1	2	3	4	5	6	7	8	9	10	11	12
1	1	3	7	E	D	B	6	C	9	2	5	A
2	0	2	6	F	C	A	7	D	8	3	4	B

3	2	0	4	D	E	8	5	F	A	1	6	9
4	7	5	1	8	B	D	0	A	F	4	3	C
5	5	7	3	A	9	F	2	8	D	6	1	E

Table 3. Substitution Box of Present Cipher

x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
S[x]	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

The ShiftRows procedure cyclically moves each state matrix row to the left by an offset proportional to its row index after replacement. In particular, the second, third, fourth, and fifth rows are moved by one, two, three, and four spots, respectively, while the first row stays the same. Through wire permutations, this procedure redistributes the replacement values between columns, aiding in inter-column diffusion without requiring a large amount of hardware overhead. MixColumns, the last transformation in each cycle, performs matrix multiplication of each column vector with a predetermined 5×5 matrix A over $GF(2^4)$.

The linear transformation performed to each column is defined by the matrix A, which is provided in Equation (2). By ensuring that every output cell is a linear combination of every cell in the associated column, this step greatly increases resistance to both linear and differential cryptanalysis. MixColumns can be implemented effectively utilizing LUT-based approaches to reduce hardware difficulty, despite the fact that it is computationally demanding due to finite field multiplications.

$$A_{100} = [0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,1\,2\,9\,9\,2]^5 = [1\,2\,9\,9\,2\,2\,5\,3\,8\,D\,D\,B\,A\,C\,1\,1\,F\,2\,3\,E\,E\,E\,8\,5\,C] \quad (2)$$

Results and Discussion

The suggested PHOTON-80/20/16 hash construction was created in Verilog-HDL and implemented on an Artix-7 FPGA device employing the Xilinx ISE 14.7 framework. Throughput measurements, power usage, latency, timing functionality, and hardware resource usage are the main areas of research. A 100-bit datapath, round-based iterative architecture, and LUT-based realization of nonlinear and diffusion aspects are all used in the implementation. Table 4 summarizes the synthesis and post-implementation outcomes, while Table 5 presents a comparison benchmarking against current PHOTON-80/20/16 versions on the same FPGA platform.

The performance parameters and resource utilization of the suggested design are given in Table 4. The implementation uses 109 LUT-based flip-flops, 420 LUTs, and 125 slices. This suggests a small footprint appropriate for integrating lightweight cryptography in FPGA systems with limited resources. Despite including finite-field MixColumns processes, the design reaches a maximum working frequency of 347.403 MHz, exhibiting high timing closure. With dynamic power limited to 20 mW and estimated static power usage of 82 mW, the total power usage comes to 102 mW. The design has a latency of 15 clock cycles per hash procedure, which is equivalent to one cycle of controlled squeezing and absorbing

permutation. A throughput of 2316.02 Mbps is attained by the architecture because of the high clock speed and low latency. Additionally, each slice's throughput-to-area efficiency reaches 18.53 Mbps, demonstrating an ideal trade-off between silicon use and performance.

The suggested implementation is contrasted with earlier PHOTON-80/20/16 architectures, such as round-based, serialized, SRL16-based, and DLP-PHOTON variations, in Table 5. The suggested architecture uses fewer slices while achieving a far higher throughput than the round-based approach in Ref [4], which uses 142 slices and reaches 387.75 Mbps. Despite achieving minimal space, the serialized and 4-bit datapath layouts in Ref [4] have very high latency (648 and 360 clock cycles, respectively), which results in very low throughput numbers (10.17 Mbps and 18.33 Mbps). The DLP-PHOTON system in Ref [22] reaches extremely high frequency (900 MHz) and throughput (3010 Mbps), but at the cost of significant slice consumption (402 slices), which lowers its area efficiency. The suggested work exhibits significantly better throughput and better efficiency (18.53 Mbps/slice versus 4.33 Mbps/slice) in comparison to Ref [12], which uses 145 slices and produces 627.38 Mbps throughput. Overall, the suggested architecture maintains a moderate space consumption while achieving the maximum efficiency of the examined designs

Table 4. Resource Utilization of PHOTON-80/20/16 hash function on Artix-7 FPGA

Resources	**Usage**
Slices	125
LUTs	420
LUT-FFs	109
Max. Frequency (MHz)	347.403
Static Power (mW)	82
Dynamic Power (mW)	20
Total Power (mW)	102
Latency (CC)	15
Throughput (Mbps)	2,316.02
Efficiency (Mbps/Slice)	18.53

Table 5. Performance comparison of proposed PHOTON-80/20/16 hash with existing PHOTON-80/20/16 hash on the same Artix-7 FPGA

Parameters	**Ref [4]**	**Ref [4]**	**Ref [4]**	**Ref [22]**	**Ref [12]**	**This work**
Architecture Type	Round	Serialized	SRL16	DLP-PHOTON	Round	Round
Datapath Bits	100	4	20	100	100	100
Slices	142	67	58	402	145	125
LUTs	336	167	144	NA	363	420
LUT-FFs	111	134	89	NA	188	109
Max. Frequency (MHz)	232.65	329.51	329.95	900	376.43	347.403
Latency (CC)	12	648	360	30	60	15
Throughput (Mbps)	387.75	10.17	18.33	3010	627.38	2316.02

Efficiency (Mbps/Slice)	2.73	0.15	0.32	7.48	4.33	18.53

Discussion: The results show that the suggested PHOTON-80/20/16 architecture successfully achieves a balance between computational performance and hardware efficiency. The current round-based architecture uses a 100-bit datapath, which allows for significant performance enhancement without corresponding increases in logic resources, in contrast to highly serialized versions that decrease area at the expense of excessive delay. The MixColumns module's LUT-based optimization, which avoids complicated finite-field multipliers and improves timing efficiency, is largely responsible for the LUT count (420), which is marginally larger than some previous round-based implementations. The suggested design's architectural benefit is evident from the efficiency statistic (Mbps per slice). With a speed of 18.53 Mbps/slice, the hardware resources are used more efficiently than in previous implementations. Although DLP-PHOTON offers an elevated actual throughput, its much bigger surface makes it less appropriate for embedded or limited IoT applications. On the other hand, the suggested approach is more suitable for lightweight hashing and authentication units in FPGA-based IoT nodes since it strikes a workable balance between speed and silicon footprint.

The lightweight design goal is further validated by the power usage findings. Even at high clock frequencies, the framework does not experience excessive switching activity, as confirmed by the low dynamic power of 20 mW. This feature is especially crucial for edge devices that run on batteries or have energy constraints. Additionally, efficiency in real-time cryptographic processes is guaranteed by the reasonable latency of 15 clock cycles. The suggested PHOTON-80/20/16 implementation offers a favorable trade-off among area, speed, power, and efficiency, according to the synthesis findings and comparison analysis. Because of this, the design is ideal for secure FPGA-based embedded systems that need lightweight, high-performance hash capability.

Conclusion

The PHOTON-80/20/16 lightweight hash function was optimized for FPGA in this work, aimed at resource-constrained situations. Substantial cryptographic diffusion and nonlinearity are achieved while preserving hardware compactness in the suggested design by utilizing the sponge structure and an AES-like permutation structure. To reduce area overhead and enhance timing performance, the design uses a round-based iterative method with a 100-bit datapath and LUT-based implementation of nonlinear and finite-field operations. The architecture comprises 125 slices and reaches a maximum operating frequency of 347.403 MHz, leading to in a throughput of 2316.02 Mbps with minimal power usage, according to implementation findings on an Artix-7 FPGA. Compared to a number of current PHOTON-80/20/16 implementations, comparative study demonstrates that the suggested architecture provides better throughput-to-area efficiency. All things considered, the implementation that is being provided offers a workable compromise between hardware consumption, performance, and security. It is ideal for Internet of Things nodes, embedded authentication systems, and lightweight secure communication modules that need FPGA-based cryptographic acceleration because to its small size and great efficiency.

References

1. Y. Zheng, “Digital signcryption or how to achieve cost (signature & encryption) << cost(signature) + cost(encryption),” in Advances in Cryptology– CRYPTO ‘97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings, pp. 165–179, 1997.
2. A. Bogdanov, M. Kneevi, G. Leander, D. Toz, K. Varc, and I. Verbauwhede, SPONGENT: A lightweight hash function, in Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst. Berlin, Germany: Springer-Verlag, 2011, pp. 312325.
3. J.-P. Aumasson, L. Henzen, W. Meier, and M. Naya-Plasencia, Quark: A lightweight hash, in Proc. Int. Workshop Cryptograph. Hardw. Embed ded Syst., 2010, pp. 115.
4. N. N. Anandakumar, T. Peyrin, and A. Poschmann, A very compact FPGA implementation of LED and PHOTON, in Proc. Int. Conf. Cryp tol., 2014, pp. 304321.
5. A. Alzahrani and F. Gebali, Multi-core data flow design and implementation of secure hash Algorithm-3, IEEE Access, vol. 6, pp. 60926102, 2018.
6. R. Martino and A. Cilardo, SHA-2 acceleration meeting the needs of emerging applications: A comparative survey, IEEE Access, vol. 8, pp. 2841528436, 2020.
7. C.-Y. Lu, Y.-W. Lin, S.-M. Jen, and J.-F. Yang, Cryptanalysis on PHOTON hash function using cube attack, in Proc. Int. Conf. Inf. Secur. Intell. Control, Aug. 2012, pp. 278281.
8. C. Dobraunig and B. Mennink, Key Recovery Attack on PHOTON Beetle, NIST, Gaithersburg, MD, USA, Tech. Rep.
9. J.Guo, T. Peyrin, and A. Poschmann, The PHOTON family of lightweight hash functions, in Proc. Annu. Cryptol. Conf., 2011, pp. 222239.
10. J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, The LED block cipher, in Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst., 2011, pp. 326341.
11. M. Al-Shatari, F.A.Hussin, A.A.Aziz, G. Witjaksono, M.S. Rohmad, and X. T. Tran, An efficient implementation of LED block cipher on FPGA, Proc. 1st Int. Conf. Intell. Comput. Eng. (ICOICE), 2019, pp. 15.
12. M. Al-Shatari, F. A. Hussin, A. A. Aziz, T. A. E. Eisa, X.-T. Tran, and M. E. E. Dalam, “IoT edge device security: An efficient lightweight authenticated encryption scheme based on LED and PHOTON,” *Appl. Sci. (Basel)*, vol. 13, no. 18, p. 10345, 2023, doi: 10.3390/app131810345.
13. C. T. Ngo, H. W. Ko, J. W. Choi, J.-W. Nam, and J.-P. Hong, “A lightweight and high yield complementary metal-oxide semiconductor true random number generator with lightweight Photon post-processing,” *Sensors (Basel)*, vol. 24, no. 23, p. 7502, 2024, doi: 10.3390/s24237502.
14. A. Sevin and A. A. Osman Mohammed, “Comparative study of blockchain hashing algorithms with a proposal for HashLEA,” *Appl. Sci. (Basel)*, vol. 14, no. 24, p. 11967, 2024, doi: 10.3390/app142411967.
15. S. Prajapat, U. Gautam, D. Gautam, P. Kumar, and A. V. Vasilakos, “Designing a robust quantum signature protocol based on quantum key distribution for E-voting applications,” *Mathematics*, vol. 12, no. 16, p. 2558, 2024, doi: 10.3390/math12162558.
16. J. Khan, C. Zhu, W. Ali, M. Asim, and S. Ahmad, “Cost-effective signcryption for securing IoT: A novel signcryption algorithm based on hyperelliptic curves,” *Information (Basel)*, vol. 15, no. 5, p. 282, 2024, doi: 10.3390/info15050282.

17. A. Sevin and Ü. Çavuşoğlu, “Design and performance analysis of a SPECK-based lightweight hash function,” *Electronics (Basel)*, vol. 13, no. 23, p. 4767, 2024, doi: 10.3390/electronics13234767.
18. H. Choi, S. Choi, and S. Seo, “Parallel implementation of Lightweight Secure Hash algorithm on CPU and GPU environments,” *Electronics (Basel)*, vol. 13, no. 5, p. 896, 2024, doi: 10.3390/electronics13050896.
19. A. Sevin, “Implementation of a data-parallel approach on a lightweight hash function for IoT devices,” *Mathematics*, vol. 13, no. 5, p. 734, 2025, doi: 10.3390/math13050734.
20. A. Zhang, T. Zhang, Z. Xi, P. Chen, J. Wei, and Y. Liu, “Secure device-to-device communication in IoT: Fuzzy identity from wireless channel state information for identity-based encryption,” *Electronics (Basel)*, vol. 13, no. 5, p. 984, 2024, doi: 10.3390/electronics13050984.
21. S. Solaiman, “Enhancing quantum key distribution security through hybrid protocol integration,” *Symmetry (Basel)*, vol. 17, no. 3, p. 458, 2025, doi: 10.3390/sym17030458.
22. B. T. Hammad, Y. A. Abbas, N. Jamil, M. E. Rusli, and M. R. Z`aba, “FPGA implementation of DLP-PHOTON hash function,” *Int. J. Future Gener. Commun. Netw.*, vol. 10, no. 12, pp. 71–78, 2017, doi: 10.14257/ijfgcn.2017.10.12.07.