International Journal of Safety and Security Engineering
Vol. 16, No. 2, February, 2026, pp. 335-341
Journal homepage: http://iieta.org/journals/ijsse

FPGA Architectures for ANU-PHOTON-Based Authenticated Encryption: A Comparative Evaluation of Encrypt-then-MAC and MAC-then-Encrypt Designs

Miaad Husam Mahdi*, Shaima Miteb Sadoon, Omar Hatem Zaidan, Riyadh Salam Mohammed

University of Diyala, Diyala 32001, Iraq

Corresponding Author Email: miaad.hm@uodiyala.edu.iq

https://doi.org/10.18280/ijsse.160208

Received: 13 December 2025
Revised: 31 January 2026
Accepted: 20 February 2026
Available online: 28 February 2026

Keywords:
authenticated encryption, lightweight cryptography, field-programmable gate array, ANU block cipher, PHOTON hash function, Encrypt-then-MAC, MAC-then-Encrypt

ABSTRACT

An authenticated encryption (AE) primitive achieves both confidentiality and authenticity simultaneously. In this paper, two field-programmable gate array (FPGA) datapath architectures based on the ANU block cipher and the PHOTON hash function are proposed to support AE employing the encryption-then-message authentication code (EtM) and message authentication code-then-encryption (MtE) constructions, denoted as ANU-PH I and ANU-PH II. To the best of our knowledge, this is one of the first works that presents FPGA-based comparative analysis of EtM and MtE constructions using ANU-PHOTON lightweight algorithms. An iterative looping architecture for both algorithms is designed and implemented on an FPGA using Xilinx ISE 14.7. On the Virtex-5 platform, the ANU-PH I architecture yielded a maximum frequency of 495 MHz and a throughput of 646.53 Mbps for the EtM method, while ANU-PH II reached a throughput of 644.63 Mbps. For the EtM method, ANU-PH II demonstrated higher throughput due to reduced latency. On the Spartan-3 platform, the proposed architectures achieved throughputs of 344.29 Mbps and 353.53 Mbps for ANU-PH I and ANU-PH II, respectively. A controller is designed to control this AE scheme that supports EtM or MtE approaches.

1. INTRODUCTION

Many devices in IoT applications have limited resources, which include computing power, storage, area, and power consumption. These constraints constitute the main challenges in securing it. Consequently, it is not feasible to apply conventional encryption algorithms. Hence, the requirement is to evolve cipher algorithms for supplying both confidentiality and integrity, and satisfying the limited resources of these devices. A security algorithm supplying confidentiality and integrity is referred to as authenticated encryption (AE). Lightweight authenticated encryption (LAE) is an AE for limited-resource devices. The encryption process outputs the ciphertext and tag pair. When verification passes, the decryption process provides the plaintext; otherwise, it provides an error. Communication protocols that need AE include Secure Socket Layer (SSL)/Transport Layer Security (TLS) [1]. There are three methods to implement the AE, which include:

Encrypt-and-MAC (E&M): the plaintext processed by the encryption algorithm and Authentication at the same time.

Encrypt-then-MAC (EtM): this approach provides the ciphertext and then computes the MAC over it.

MAC-then-encrypt (MtE): this method computes the MAC for the plaintext, then the result is enciphered together with the original plaintext [1].

Achieving parallelism with E&M is possible. Therefore, it provides the most efficient results. However, E&M does not destroy the statistical dependencies of the original data and MAC; therefore, it is the least secure. The sequential execution of the EtM and MtE approaches slowed them. But both approaches conceal the statistical correlations of the plaintext. EtM provides the most secure results [1, 2].

To provide confidentiality for limited-resource devices, lightweight block ciphers are the most suitable option. Hash functions create a message authentication code (MAC) scheme. It offers data integrity and authenticity. AE integrates block cipher algorithms and hash functions to ensure confidentiality and integrity, with authentication at the same time [3]. However, existing approaches often face limitations in scalability, resource overhead, and resilience against advanced attacks in hybrid wireless networks.

Many algorithms have been proposed for AE, including Beetle [4], Encrypt-then-Authenticate Transform (ESTATE) [5], Minalpher [6] and Abbas et al. [7]. Moreover, several AE modes such as Galois/Counter Mode (GCM), Counter with CBC-MAC (CCM), Encrypt-then-Authenticate-then-Translate (EAX mode) EAX, and Offset Codebook Mode (OCB) have been proposed. Recent studies have further explored LAE schemes for resource-constrained environments [8-10]. Among these, AES-GCM is a commonly used algorithm. The parallelism is an inherent feature of the AES-GCM algorithm; hence, low-cost and low-latency implementations can be designed with high speed [11, 12]. Although of the growing interest in LAE for resource-constrained devices, existing studies mainly emphasize either

software solutions or single-architecture field-programmable gate array (FPGA) implementations, providing little comprehensive comparison among different authentication encryption constructions. Specifically, limited attention has been given to the hardware-level comparison of EtM and MtE methods when designed using lightweight cryptographic techniques.

Moreover, although lightweight block ciphers and hash functions, for instance, ANU and PHOTON, have been individually explored, their integration into a combined FPGA-based AE hardware architecture is still insufficiently investigated. Most prior work also provides limited analysis of the trade-offs between area, throughput, and latency when multiple architectural implementations are studied.

To tackle these limitations, this study provides two FPGA-based AE datapath architectures that leverage ANU [13] and PHOTON [14], and the resulting design is implemented on a FPGA using Xilinx 14.7. Furthermore, this work presents a comparative analysis of EtM and MtE methods. The work focuses on investigating performance trade-offs in terms of hardware resource utilization, throughput, latency, and efficiency via different FPGA platforms.

2. LITERATURE REVIEW

Recent studies have highlighted the importance of lightweight cryptography for securing resource-constrained embedded and IoT devices while reducing power, memory, and hardware overhead. Among existing lightweight block ciphers, ANU has been reported to achieve a good trade-off between low area consumption and high throughput, making it suitable for FPGA implementations. Previous research has demonstrated efficient hardware datapath designs for ANU on the FPGA platform, focusing on optimizing gate count, latency, and energy consumption. Comparative evaluations show that ANU outperforms several contemporary lightweight ciphers in terms of throughput and efficiency metrics [2].

Since its standardization, the Advanced Encryption Standard (AES) and its authentication mode, AES-GCM, have been widely adopted in security and power-constrained applications, spurring significant research on efficient hardware implementations. Previous work has evaluated and optimized ASIC architectures for AES-GCM components, focusing specifically on low-complexity and low-power S-box designs, as well as high-speed Galois field arithmetic in 65-nm CMOS technology. These studies demonstrate that optimized AES-GCM architectures can achieve significantly higher throughput and lower latency than previous implementations, while maintaining low hardware complexity [15]. Lightweight block ciphers have been widely proposed to address the security challenges of resource-constrained IoT devices, where traditional cryptographic algorithms are inefficient due to power and processing limitations. Recent FPGA-based implementations of hybrid substitution-permutation and Feistel network (SFN) ciphers demonstrate lower hardware utilization and power consumption while maintaining high performance compared to similar lightweight architectures [16, 17]. Hardware replay protection relies on a fixed-size Bloom filter for nonce tracking, which can saturate as the number of unique nonces increases, leading to increased false-positive rates in long-term or high-throughput operations and possibly requiring periodic resets, which could lead to unintentional replay attacks after the entry is cleared [18].

Despite prior work that has dealt with LAE and FPGA-based designs, most previous studies explore individual architectures or software-based implementations and provide restricted analysis of the trade-offs between different authentication approaches. Particularly, a detailed FPGA-based comparison of EtM and MtE constructions employing ANU-PHOTON has not been sufficiently investigated.

To address these limitations, this work provides two FPGA-based AE architectures and presents a comparative evaluation of their performance in terms of area, throughput, latency, and efficiency.

3. AUTHENTICATED ENCRYPTION SCHEME

The ANU block cipher uses one of two keys with lengths of 128 or 80 bits. This work uses l28 bit key scheduling to provide more security. An iterated looping architecture has been designed and implemented on various FPGA devices. The ANU block cipher architectures of the encryption, decryption, and key scheduling are presented in Figures 1(a) and (b). The photon hash function has five different flavours. Each flavour has a different internal state size. This work used a permutation of size 100 bits.

To implement the MtE and EtM approaches, the two architectures are merged. Two designs for ANU and PHOTON AE are proposed. The first design is called (ANU-PH I), and the second design is called (ANU-PH II). The key difference between the two designs lies in their control mechanism and the number of clock cycles involved, particularly for the MtE construction. ANU-PH I follows a sequential control strategy, resulting in a higher number of clock cycles, while ANU-PH II incorporates an optimized control approach that minimizes latency and maximizes throughput.

3.1 ANU-PH I

For the first design (ANU-PH I), the top-level block of the AE scheme, composed of ANU block cipher and PHOTON hash function, is presented in Figure 2. The input to the hardware architecture consists of 128 bits for the encryption key and a 64-bit message block. Also, the signals ctr, clk, rst, rst1, rst2, sel0, sel1, and EtM are inputs to the circuit, each with one bit only. The signal ctr operates as an enable to the logic circuit. Clk is the clock signal. rst, rst1, and rst2 represent the reset signals where the rst signal resets the whole circuit, while rst1 and rst2 signals reset the PHOTON logic elements and ANU logic elements, respectively. The control signals EtM, sel0, and sel1 are needed to implement AE approaches. Table 1 explains the operation of the AE concerning the EtM and MtE approaches.

The input to our AE scheme is only 64 bits. Therefore, 84 bits with zero value are generated and added to this input as the most significant bits since the input of the PHOTON hash function is 148 bits. The output of the AE scheme is 164 bits in two ports. A one-hundred-bit port is used for the hash value. A 64-bit port is used when the ciphertext is produced.

3.2 ANU-PH II

The second design (ANU-PH II) of AE is the same as the first design, with some differences. The top-level interface of the second design is shown in Figure 3. Six input ports, which include ShKEY, shmessage, EtM, rst, rst1, and clk. The

operation of these six input ports is the same as for the ANU-PH I design. Four output ports cipher_text, MACcipher_text, anu_ready and phot_ready. The operation of the ANU-PH II design is simple. When the EtM signal is equal to one, the ANU-PH II would work as an encryption, then a message authentication code. In contrast to the ANU-PH I design, the ANU-PH II design works as a message authentication code, then encryption, when the EtM signal is equal to zero.

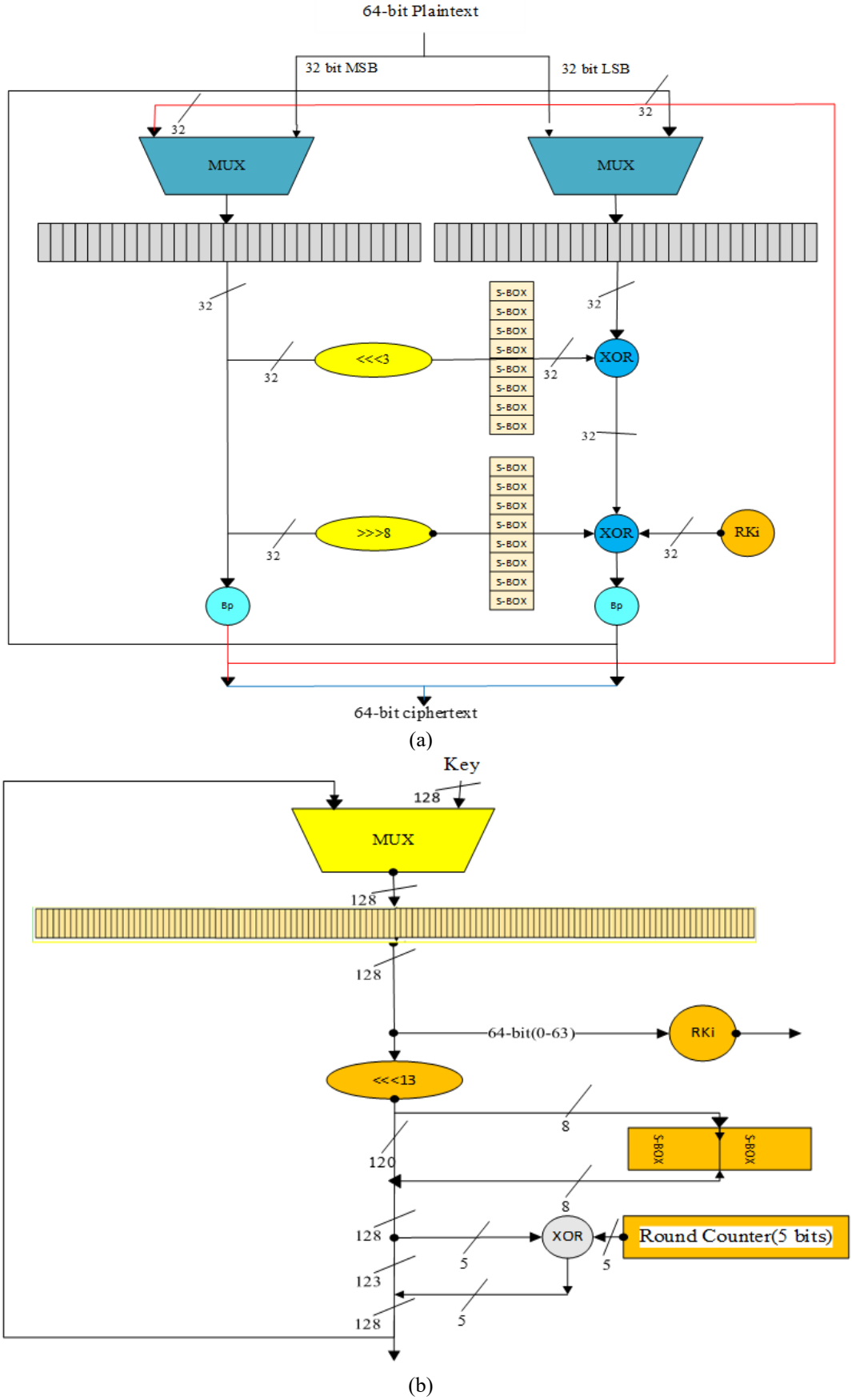

Figure 1. ANU algorithm: (a) ANU encryption architecture and (b) Key scheduling

Table 1. The operation of the authenticated encryption (AE) scheme to implement EtM and MtE methods using ANU and PHOTON

S.	EtM	Sel0	Sel1	Output	Operation	Explanation
1	1	1	x	Ciphertext (0 to 63) pin	EtM	The first step of the EtM method is to encrypt the plaintext and produce the ciphertext on the cipher_text port.
2	1	0	x	Hash value (0 to 99) pins	EtM	The second step of the EtM method is to process the ciphertext and produce the hash value on the MACcipher_text port.
3	0	1	1	Ciphertext (0 to 63) pin	MtE	The first step of the MtE method is to encrypt the plaintext and produce the ciphertext on the cipher_text port.
4	0	1	0	-	MtE	The second step of the MtE method is to process the plaintext and produce the hash value.
5	0	0	1	Ciphertext (0 to 63) pin	MtE	The output hash value in the previous step, from (0 to 63 bits), is further encrypted and produced on the cipher_text port.
6	0	0	0	Ciphertext (0 to 63) pin	MtE	The remaining output hash value from (64 to 99 bits) is further encrypted and produced on the cipher_text port.

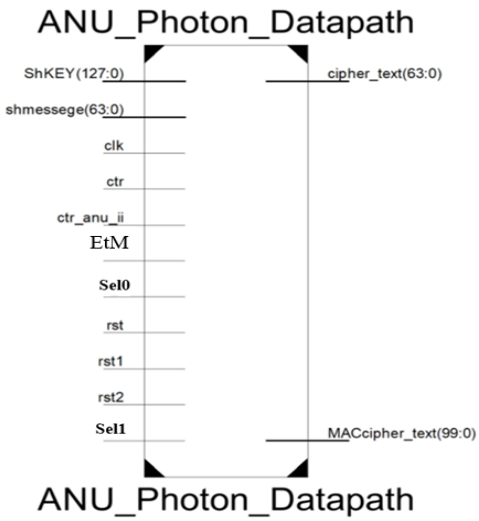

Figure 2. The top-level block of the authenticated encryption (AE) scheme (ANU-PH I)

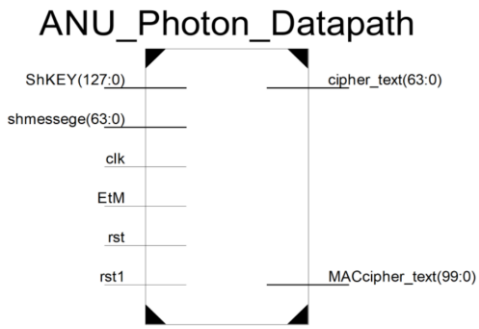

Figure 3. The top-level block of the authenticated encryption (AE) scheme (ANU-PH II)

The two signals anu_ready and phot_ready in Figure 4 are included in the datapath to implement the ANU-PH II design. The operation of EtM and MtE in ANU-PH II design is controlled externally by the EtM signal only. The signals ctr, ctr_anu, and rst2 are generated internally to apply the two approaches of AE ANU-PH II.

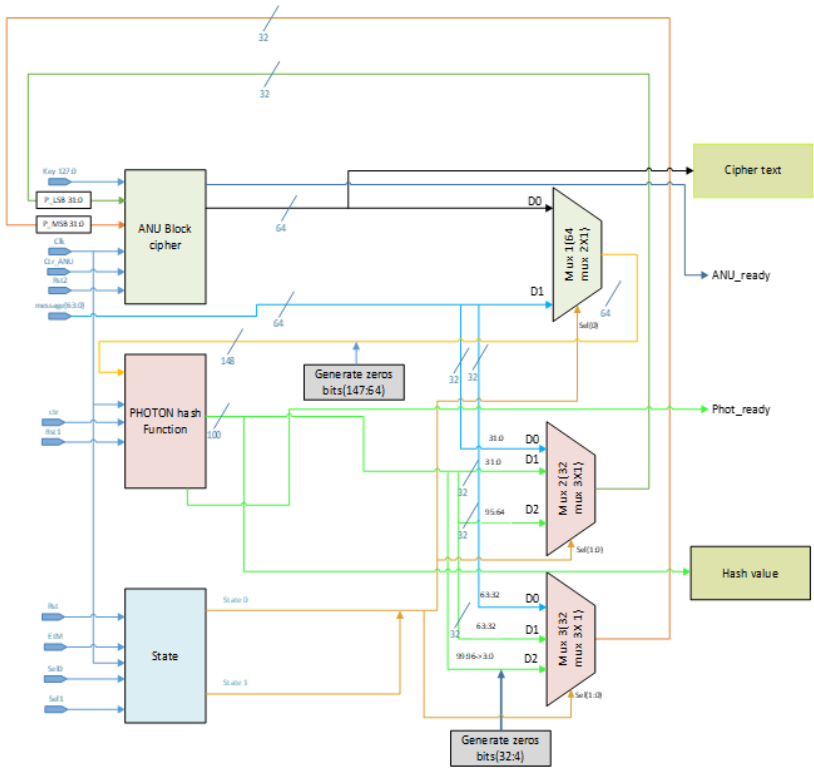

Figure 4. Datapath of authenticated encryption (AE) approaches: (a) MtE. (b) EtM for both designs (ANU-PH I, ANU-PH II)

4. ANU ALGORITHM AND PHOTON DATAPATH

The encryption process is performed by the ANU block cipher, and the hash value is produced by the PHOTON hash function as presented in Figure 4. The Multiplexer (Mux-1) consists of 64 (2 × 1) multiplexers. The remaining two multiplexers (Mux-2 and Mux-3) each consist of 32 multiplexers with 2 inputs and one output. These multiplexers are necessary to implement the encryption then message authentication code (EtM) or message authentication code then encryption (MtE). In EtM, firstly, the message input goes through a multiplexer (Mux-2) and a multiplexer (Mux-3) through (D0) inputs to produce the plaintext for the ANU block cipher. The D0 input of Mux-2 produces the least significant bits (P_LSB 31:0) input for the ANU block cipher, and the D0 input of Mux-3 produces the most significant bits (P_MSB 31:0) input for the ANU block cipher. Then, the plaintext is encrypted, and the ciphertext goes through Mux-1 to produce the input for the PHOTON hash function. Since the output of the encryption process is only 64 bits and the input of the PHOTON hash function is 148 bits, it is necessary to generate zero bits. The generated zero bits (147:64) block is used to fill bits from (147) to (64) with zeros. After that, the ciphertext is appended with zero bits, and then it is entered into the PHOTON hash function to produce the hash value.

In MtE, the message input goes through a multiplexer (Mux-1). The output of this multiplexer is appended with zeros to produce a 148-bit input to the PHOTON hash function. The output of the PHOTON hash function is a 100-bit value that constitutes the hash value. The first 64 bits of the PHOTON output or hash value are connected to D1 inputs of Mux-2 and Mux-3. The output of Mux-2 consists of bits from 31 to 0, which are connected to the P_LSB input of the ANU block. Similarly, the output of Mux-3 consists of bits from 63 to 32, which are connected to the P_MSB input of the ANU block. The remaining 36 bits are connected to the D2 inputs of the two multiplexers. The process of encryption for the hash value is performed in two stages. In the first stage, the least significant 64 bits are encrypted. In the second stage, the remaining 36 bits are first appended with zeros to constitute a 64-bit input for the ANU block cipher.

In the ANU-PH I design, the state block provides the necessary logic to perform the process of AE. The control signals connected to the logic of the state block include a clock signal, EtM, Sel0, and Sel1. The clock signal is connected to the three blocks, which are the ANU block cipher, the PHOTON hash function, and the state logic block. The EtM signal is used to perform either the EtM approach or the MtE approach of AE. When the EtM signal is equal to 1, the encryption and message authentication code (EtM) is performed. Otherwise, when the EtM signal is equal to 0, the message authentication code then encryption (MtE) is performed. The Sel0 and Sel1 signals are necessary for the MtE approach. When both signals Sel0 and Sel1 are equal to 1, the plaintext is encrypted. When the signals are equal to (1, 0), the hash value is computed over the plaintext. When the signals are equal to (0, 1), the least significant 64 bits of the hash value, produced in the previous step, are encrypted. And, when both signals are equal to (0, 0), the remaining 36 bits are appended with zeros and encrypted. Three reset signals are used. These reset signals include (Rst, Rst1, and Rst2). These reset signals are connected to the state block, PHOTON hash function, and ANU block cipher, respectively. Two enable signals are used, which are the ctr_ANU, which is connected to the ANU block, and the ctr, which is connected to the PHOTON hash function.

Although the two signals anu_ready and phot_ready are included in the datapath in Figure 4, they are used only for ANU-PH II design. However, the sel0 and sel1 signals are utilized for the ANU-PH I design only. The operation of EtM and MtE in ANU-PH II design is controlled externally by the EtM signal only. The signals ctr, ctr_anu, and rst2 are generated internally to apply the two approaches of AE.

5. FUNCTIONAL RESULTS OF THE AUTHENTICATED ENCRYPTION SCHEME

To verify the functional consistency of the proposed architectures, a set of test vectors was applied to the ANU-PH I and ANU-PH II structures. The validation procedure involved validating the generated ciphertext and authentication tag with the predicted outputs for both EtM and MtE constructions. The simulations were carried out using Xilinx ISE 14.7, and the input data involved different plaintext blocks, key values, and hash inputs. The results confirmed that the output values were consistent with the expected behavior of the ANU block cipher [13] and PHOTON hash function [14].

In MtE, the plaintext is first encrypted and placed at the output at clock cycle 13. The plaintext is again processed by the hash function that produces the hash value, which is also encrypted. Thirty-six clock cycles are needed for the hash value to be available. Since the hash value is 100 bits, it will be encrypted as two blocks.

A 0'z will be added to the second block to be 64 bits. Seventy-five clock cycles are needed in total to process one 64-bit data block and produce the encrypted data blocks.

In EtM, the plaintext is first encrypted and placed at the output at clock cycle 13. Then the encrypted plaintext is processed by the hash function (PHOTON) that produces the hash value. Forty-nine clock cycles are needed in total to process one 64-bit data block.

The difference between ANU-PH I and ANU-PH II designs is that only sixty-two cycles are needed for MtE by the ANU-PH II design to produce the tag and ciphertext on the output. Also, the throughput of the same design is higher than the ANU-PH I's throughput. However, the cost of the second design is higher than that of the first design.

6. PERFORMANCE RESULTS AND COMPARISON WITH OTHER CIPHERS

The performance metrics include throughput, maximum frequency, area, latency, and efficiency. In ANU-PH I design, the throughput of the proposed work is 646.530612 Mbps and 422.4 Mbps for Virtex-5 and Spartan-3, respectively. The maximum frequency is obtained from the synthesis report of Xilinx ISE (14.7). The maximum frequencies of the two platforms, Virtex-5 and Spartan-3, are 495 and 270.672, respectively. The area includes the number of occupied slices, the number of look-up tables (LUT), and the number of flip flops. The area parameters are obtained from the device utilization summary of the ISE 14.7. The latency of the EtM AE is a 49-clock cycle. The ANU cipher consumes 13 clock cycles to produce the ciphertext. Then, the PHOTON hash function consumes 36 clock cycles to produce the hash value

for the ciphertext produced previously. However, the latency of the MtE in ANU-PH I AE is 75 clock cycles. These 75 clock cycles are distributed as follows: the PHOTON algorithm consumes 36 clock cycles to produce the 100-hash value. The 100-hash value is encrypted by the ANU cipher as two blocks. The first block is the least significant 64-bit. The second block consists of the remaining bits, which are padded with zeros to form a 64-bit block. Consequently, encrypting the two blocks requires 26 clock cycles using the ANU cipher. The original plaintext is encrypted within 13 clock cycles.

As shown in Table 2, the proposed ANU-PH I architecture implemented on the Virtex-5 platform achieves a maximum frequency of 495 MHz and a throughput of 646.53 Mbps for the EtM construction, indicating strong performance in terms of speed and efficiency. When compared with PRESENT and SPONGENT [3], the proposed architecture operates at a significantly higher frequency and achieves higher throughput. However, it should be noted that these implementations differ in platform specifications, parameter configurations, and reporting units, and therefore, the comparison is indicative rather than strictly uniform.

In comparison with LED and PHOTON implementations [9, 19], the proposed design shows lower hardware area in terms of slice utilization on similar FPGA platforms. Moreover, the key size used in this work (128 bits) is larger than that of LED and PHOTON designs, which typically use 64-bit keys. This demonstrates a higher security level in this work; however, direct comparison should take into consideration differences in design objectives and parameter settings.

The number of slices for the ANU-PH II design is higher than the number of slices for the ANU-PH I design for the two devices. The latency for the MtE approach of the second design is less than the latency of the first design. So, the throughput of the MtE method is increased to be higher than the throughput of the first design for the two platforms.

In addition to the numerical results, it is essential to evaluate the trade-offs between the two proposed architectures. ANU-PH I achieves higher area efficiency, entailing fewer hardware resources in terms of slices and LUTs. Nevertheless, this comes at the cost of higher latency because of the larger number of clock cycles necessary for processing, particularly in the MtE design.

In contrast, ANU-PH II decreases the number of clock cycles through a refined control mechanism, producing lower latency and enhanced throughput. This makes ANU-PH II more appropriate for applications where high performance and high-speed processing are required, while ANU-PH I is more suitable for resource-constrained environments where hardware area is a major factor.

Table 2. Results of the authenticated encryption (AE) scheme and comparison with other schemes

Block Cipher	Target Device	Method	# Of Slices	# Of LUT	# Of Flip Flop	Max. Freq. (MHz)	Latency	Block Size	Key	Hash Value	Throughput (Mbps)	Efficiency (Mbps/Slice)
Proposed work (ANU-PH I)	Virtex- 5 xc5vlx330t-2ff1738	EtM	390	940	750	495	49	64	128	100	646.530612	1.6577708
		MtE					75				422.4	1.08307692
Proposed work (ANU-PH II)	Virtext 5 xc5vlx330t-2ff1738	EtM	415	970	715	493.55	49	64	128	100	644.636735	1.55334153
		MtE					62				509.470968	1.22764089
PRESENT & SPONGENT [19]	Virtex-5	EtM	174	-	149	100khz	121	64	80	88	82.64kbps	0.47
		MtE	174	-	149	100khz	152	64	80	88	65.78kbps	0.37
LED & PHOTON [20]	Virtex-5	EtM/MtE	415	-	-	587.9	44	100	64	100	1336	3.2
	Spartan-3		825	-	-	332.7	44				756.1	0.92
LED & PHTON [7]	Spartan-3	EtM/MtE	863	-	-	287.5	44	64	64	100	653.41	0.76
Proposed work (ANU-PH I)	Spartan 3 xc3s5000-5fg900	EtM	740	1250	750	263.602	49	64	128	100	344.29649	0.465265527
		MtE					75				224.940373	0.303973477
Proposed work (ANU-PH II)	Spartan 3 xc3s5000-5fg900	EtM	842	1340	715	270.672	49	64	128	100	353.530776	0.41987028
		MtE					62				279.403355	0.331832963

7. CONCLUSION

An AE datapath composed of two algorithms is implemented on FPGA. The algorithms used are the ultra-lightweight ANU block cipher and the PHOTON hash function. An iterative looping architecture for both algorithms is designed and implemented using Xilinx ISE 14.7. Two AE methods, namely EtM and MtE, are investigated. Two architectures, ANU-PH I and ANU-PH II, have been developed. The main difference between the two architectures lies in the number of clock cycles required to produce the output, particularly for the MtE method. Several performance metrics are used to evaluate and compare the proposed architectures with other AE constructions. Other architecture types can be designed for the two algorithms to further increase throughput or reduce hardware resource utilization.

REFERENCES

[1] Agrawal, M., Zhou, J., Chang, D. (2019). A survey on lightweight authenticated encryption and challenges for securing industrial IoT. Advanced Sciences and Technologies for Security Applications, 2019: 71-94. https://doi.org/10.1007/978-3-030-12330-7_4

[2] Yousif, N.H., Abbas, Y.A., Ali, M.H. (2022). Lightweight ANU-II block cipher on field programmable gate array. International Journal of Electrical and Computer Engineering, 12(3): 2194. https://doi.org/10.11591/ijece.v12i3.pp2194-2205

[3] Suryateja, P.S., Rao, K.V. (2024). A survey on lightweight cryptographic algorithms in IoT. Cybernetics and Information Technologies, 24(1): 21-34. https://doi.org/10.2478/cait-2024-0002

[4] Chakraborti, A., Datta, N., Nandi, M., Yasuda, K. (2018). Beetle family of lightweight and secure authenticated encryption ciphers. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2018(2): 218-241. https://doi.org/10.46586/tches.v2018.i2.218-241

[5] Chakraborti, A., Datta, N., Jha, A., Mancillas-López, C., Nandi, M., Sasaki, Y. (2020). ESTATE: A lightweight and low energy authenticated encryption mode. IACR Transactions on Symmetric Cryptology, 350-389. https://doi.org/10.13154/tosc.v2020.iS1.350-389

[6] Kosug, M., Yasuda, M., Satoh, A. (2015). FPGA implementation of authenticated encryption algorithm Minalpher. In 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, pp. 572-576. https://doi.org/10.1109/GCCE.2015.7398679

[7] Abbas, Y.A., Jidin, R., Jamil, N., Z'aba, M.R. (2018). Reusable data-path architectures for EtM and MtE on FPGA. Advanced Science Letters, 24(1): 757-761. https://doi.org/10.1166/asl.2018.11809

[8] Soto-Cruz, J., Ruiz-Ibarra, E., Vázquez-Castillo, J., Espinoza-Ruiz, A., Castillo-Atoche, A., Mass-Sanchez, J. (2024). A survey of efficient lightweight cryptography for power-constrained microcontrollers. Technologies, 13(1): 3. https://doi.org/10.3390/technologies13010003

[9] Kumar, S., Kumar, D., Dangi, R., Choudhary, G., Dragoni, N., You, I. (2024). A review of lightweight security and privacy for resource-constrained IoT devices. Computers, Materials and Continua, 78(1): 31-63. https://doi.org/10.32604/cmc.2023.047084

[10] Alaba, F.A., Othman, M., Hashem, I.A.T., Alotaibi, F. (2017). Internet of Things security: A survey. Journal of Network and Computer Applications, 88: 10-28. https://doi.org/10.1016/j.jnca.2017.04.002

[11] Ahmed, S., Ahmad, N., Shah, N.A., Abro, G.E.M., Wijayanto, A., Hirsi, A., Altaf, A.R. (2025). Lightweight AES design for IoT applications: Optimizations in FPGA and ASIC with DFA countermeasure strategies. IEEE Access, 13: 22489-22509. https://doi.org/10.1109/ACCESS.2025.3533611

[12] Kaur, J., Cintas Canto, A., Mozaffari Kermani, M., Azarderakhsh, R. (2025). A survey on the implementations, attacks, and countermeasures of the NIST lightweight cryptography standard: Ascon. ACM Computing Surveys, 58(1): 1-16. https://doi.org/10.1145/3744640

[13] Bansod, G., Patil, A., Sutar, S., Pisharoty, N. (2016). ANU: An ultra lightweight cipher design for security in IoT. Security and Communication Networks, 9(18): 5238-5251. https://doi.org/10.1002/sec.1692

[14] Guo, J., Peyrin, T., Poschmann, A. (2011). The PHOTON family of lightweight hash functions. In Annual Cryptology Conference, pp. 222-239. https://doi.org/10.1007/978-3-642-22792-9_13

[15] Sung, B.Y., Kim, K.B., Shin, K.W. (2018). An AES-GCM authenticated encryption crypto-core for IoT security. In 2018 International Conference on Electronics, Information, and Communication (ICEIC), Honolulu, HI, USA, pp. 1-3. https://doi.org/10.23919/ELINFOCOM.2018.8330586

[16] Abbas, Y.A., Mahdi, M.H., Al-Azawi, S. (2025). FPGA Implementation of SFN lightweight encryption algorithm. International Journal of Safety & Security Engineering, 15(9): 1819-1825. https://doi.org/10.18280/ijsse.150906

[17] Li, L., Liu, B., Zhou, Y., Zou, Y. (2018). SFN: A new lightweight block cipher. Microprocessors and Microsystems, 60: 138-150. https://doi.org/10.1016/j.micpro.2018.04.009

[18] Gladis Kurian, M., Chen, Y. (2025). Ascon on FPGA: Post-quantum safe authenticated encryption with replay protection for IoT. Electronics, 14(13): 2668. https://doi.org/10.3390/electronics14132668

[19] Hatzivasilis, G., Floros, G., Papaefstathiou, I., Manifavas, C. (2016). Lightweight authenticated encryption for embedded on-chip systems. Information Security Journal: A Global Perspective, 25(4-6): 151-161. https://doi.org/10.1080/19393555.2016.1209259

[20] Abbas, Y.A., Jidin, R., Jamil, N., Zaba, M.R. (2016). Reusable data-path architecture for encryption-then-authentication on FPGA. International Review on Computers and Software, 11(1): 56-63. https://doi.org/10.15866/irecos.v11i1.8367