Novel High Throughput-to-Area Efficiency and Strong-Resilience Datapath of AES for Lightweight Implementation in IoT Devices

Pao-Ying Cheng, Ying-Cheng Su, and Paul C.-P. Chao, *Fellow, IEEE*

***Abstract*—A new datapath for the advanced encryption standard (AES) is proposed in this work, which is successfully optimized with a high efficiency of throughout-to-area for lightweight applications in Internet of Things (IoT) devices. The proposed AES architecture enables parallel encryption of 32-bit blocks for efficient processing of 128-bit data while minimizing hardware area. Optimization is achieved by utilizing shift registers instead of conventional registers in the ShiftRows, MixColumns, and key expansion stages of the 32-bit AES operation. Our implementation, based on the TSMC 40-nm process, achieves a throughput of 692.65 Mb/s, with a gate count of 5.65K and a figure of merit (FOM) of 122.59-Mb/s/k-gate, better than all the previous works in terms of efficiency. Furthermore, our proposed 32-bit datapath ensures security against correlation power analysis attacks owing to designed simultaneously active encryption and decryption, as the 32-bit key out of 128 bits remains unrevealed even with 100 000 traces for attack.**

***Index Terms*—32-bit datapath, advanced encryption standard (AES), differential power analysis (DPA), high efficiency, lightweight, low area.**

I. Introduction

CRYPTOGRAPHIC algorithms are designed and applied for the encryption and decryption of data communicated between devices and systems, which is particularly crucial for ensure the security of data transmission between Internet of Things (IoT) devices and the cloud. Well-designed cryptographic algorithms are expected to provide authentication, confidentiality, and data integrity for communications initiated by IoT devices. The performance of such crypto-algorithms is often evaluated by security strength and the execution speed when implemented in various hardware forms. There are a few crypto-algorithms being used now for data communication of IoT devices. The most popular one is the advanced encryption standard (AES, which was published in 2001 by the national institute of standards and technology (NIST) [1]. The Rijndael algorithm was selected as the standard for AES. AES in fact replaced the older and weaker data encryption standard (DES) and has become the most widely used cryptographic algorithms used for data confidentiality during communication. To date, numerous circuit designs for implementing AES computations in hardware have been reported, all of which can be categorized into two groups for different ends, high throughput, or low cost by small chip area [2], [3].

One of the challenges in achieving the required security of AES is the limited hardware resources available in IoT devices for conducting required computations of encryption and decryption. Therefore, the hardware implementation of AES is expected to be more resource efficient, also known as "lightweight," to be accommodated in the IoT edge device. On the other hand, there is a growing demand for AES encryption and decryption devices to achieve higher processing speeds to meet various application requirements. High-speed applications, such as network routers, require gigabit-per-second processing capabilities [3], while other devices, such as handheld or home devices, may only need to process data at the megabit-per-second level. However, due to the substantial computational complexity of AES, simultaneously reducing the operation time and circuit area of the designed AES is challenging. It is not an easy task to balance the throughput and hardware resource consumption while maintaining a positive user experience. Nonetheless, researchers continue to explore the possibility of elevating the throughput-to-area efficiency of AES. Even though the algorithm has been around for more than 20 years since 2001, ongoing efforts are valuable to enhance its performance, especially for IoT devoices.

In recent years, efforts are dedicated by researchers to develop implementations of AES with a favorable balance between throughput and chip area. Satoh et al. [4] are the first to describe the compact high-speed hardware architecture and logic optimization method for the AES algorithm. This work combined the encryption and decryption datapaths and optimized the S-Box structure by introducing a new compound domain. Pramstaller et al. [5] proposed a compact AES co-processor that covers all key lengths (128, 192, and 256 bit) for encryption and decryption and supports cipher block chaining mode (CBC). Mathew et al. [6] achieved the process of performing encryption and decryption with minimal energy

Manuscript received 29 September 2023; revised 31 December 2023; accepted 22 January 2024. Date of publication 29 January 2024; date of current version 9 May 2024. This work was supported in part by the PUFsecurity Corporation; in part by the National Science and Technology Council under Grant 111-2221-E-A49-159-MY3 and Grant 112-2223-E-A49-006; in part by the Higher Education Sprout Project of the National Yang Ming Chiao Tung University and Ministry of Education (MOE), Taiwan; and in part by the Hsinchu and Southern Taiwan Science Park Bureaus, Ministry of Science and Technology, Taiwan, under Contract 110CE-2-02 and Contract 112AO28B. *(Corresponding author: Paul C.-P. Chao.)*

The authors are with the Department of Electronics and Electrical Engineering, National Yang Ming Chiao Tung University, Hsinchu 300093, Taiwan (e-mail: ballin747.ee10@nycu.edu.tw; buck800817@gmail.com; pchao@mail.nctu.edu.tw).

Digital Object Identifier 10.1109/JIOT.2024.3359714

usage by optimizing the GF$(2^4)^2$ circuit to reduce the delay of the S-Box unit and using the folded data path design to reduce the wiring complexity of the ShiftRow arrangement. In subsequent years, several researchers [7], [8], [9], [10], [11] focus on minimizing the area to below 10k gate count. Wang and Ha [12] dedicated to achieving encryption computations within a single cycle to significantly boost throughput. Kouser et al. [13] and other works [14], [15], [16], [17], [18], [19], [20] were committed to enhancing throughput, achieving a maximum of 2457 Mb/s. Some research works, including Bui et al. [21] and others [22], [23], [24] managed to keep their power consumption below 0.8 pJ/bit by minimizing the active power in S-box, clock gating strategies, and data paths. Chong et al. [25] achieved the lowest power consumption of 0.23 pJ/bit. More recently, both Kundi et al. [26] and Ng et al. [27] have been seeking a balance between area and throughput. Notably, Kundi et al. [26] achieved the best experimental result with a figure of merit (FOM) of 106.18 Mb/s/k-gate, which is actually the ratio of the throughput divided by gate count.

The focus of this study is to achieve high throughput-to-area efficiency for varied lightweight implementations of IoT devices, while maintaining strong resilience to attacks. Toward this end, the *composite field algorithm* is employed herein to compute the S-box, while *a matrix multiplication method* for Mix/InvMix-Columns and *novel ShiftRow/Inverse-ShiftRow operation* are adopted, successfully minimizing the area to 5.65K gates, and increasing the throughput to 692.65 Mb/s. As a result, the FOM delivered by the proposed AES achieves as high as 122.59 Mb/s/k-gate, *the highest to date* compared to all the reported works. On top of the favorable efficiency achieved is the security evaluation on the proposed datapath conducted by this study. It is shown by experiments that thanks to the designed simultaneously active operations in forward and inverse datapaths during either encryption or decryption by the proposed novel AES architecture, the power attacks are successfully thwarted. This countermeasure design against power attacks employs a hiding strategy through asynchronous operation [28], [29]. This involves simultaneously executing encryption and decryption operations within the same cycle to hide the sensitive power leakage from the S-box. Experiments show the 32-bit key out of 128 bits remains secure against correlation power analysis attacks, which actually requires more than 100 000 traces for potential leakage, showing strong resilience offered by the designed AES.

The remainder of this article is organized as follows. Section II describes the proposed datapath architecture for AES, highlighting the optimizations employed. Section III presents the experimental results, including throughput, gate count, and the FOM, and discusses the security evaluation of our design. Finally, Section IV concludes this article and suggests directions for future research.

II. Proposed Hardware Architecture on AES

A new hardware architecture of AES is proposed herein to minimize hardware area based on shared hardware resources among different computation sub-blocks, including SubBytes, ShiftRows, MixColumns, AddRoundKey, and KeyExpansion for encryption, while Inverse SubBytes, Inverse ShiftRows, Inverse MixColumns, AddRoundKey, and Inverse KeyExpansion for decryption. As illustrated in Fig. 1, the designed AES scheme proposed herein conducts forward and inverse operations simultaneously, as shown in black blocks, for encryption in blues and decryption in greens, respectively, and determining which operation result to use based on whether encryption or decryption is being performed. This approach helps reduce sensitivity between power and computation of AES, thereby enhancing defense ability of AES against power attacks. Furthermore, toward the minimized chip area, all the multiplications involved in the afore-mentioned sub-blocks of SubBytes and Inverse SubBytes are carried out in a composite field instead of storing the S-box in memory for the computation. On the other hand, the MixColumns and Inverse MixColumns are conducted by smaller matrix multipliers. In addition, the proposed AES adopts a dynamic key scheduling architecture for KeyExpansion and Inverse KeyExpansion to reduce hardware resources, eliminating the need to generate key storage in memory for each round operation. Note that

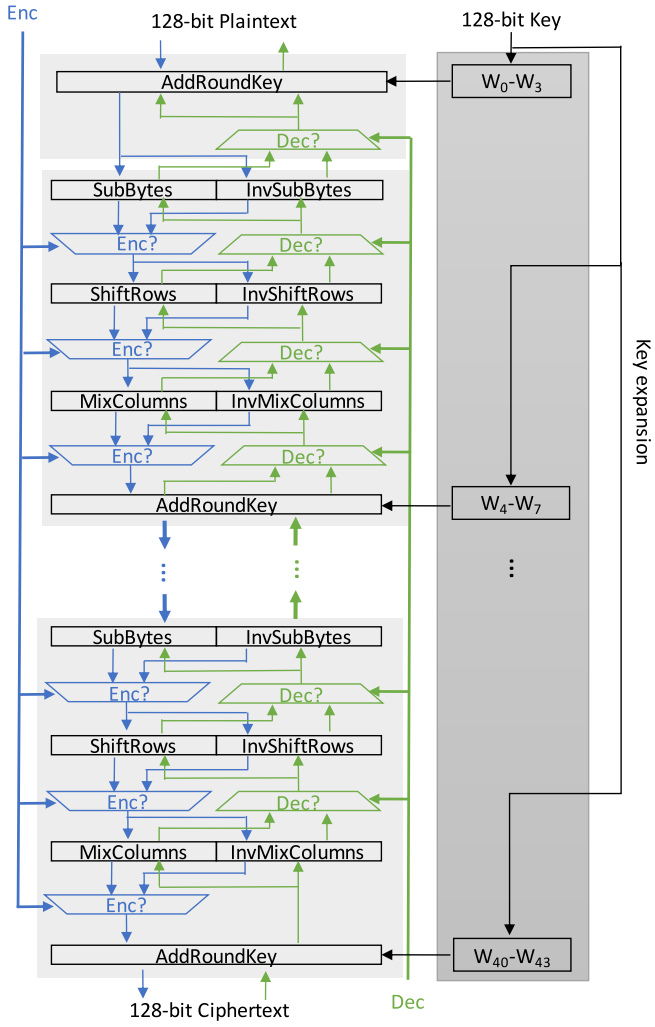

Fig. 1. Computation flow of AES in blocks.

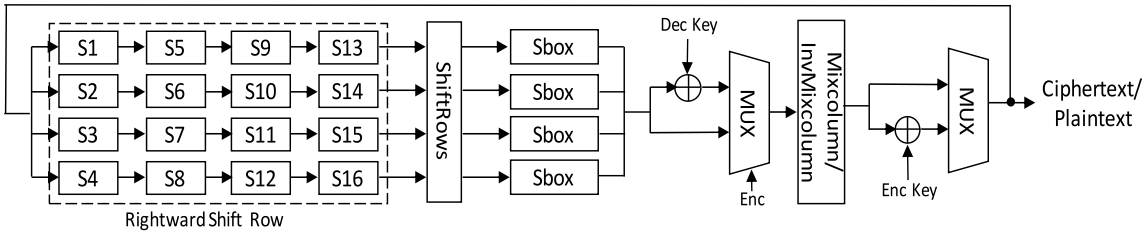

Fig. 2. Architecture of the proposed 32-bit datapath for AES.

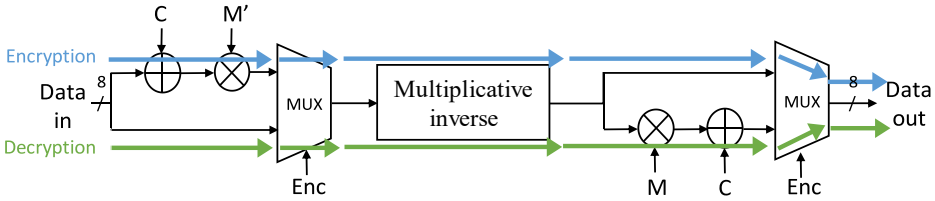

Fig. 3. Computation of SubBytes and Inverse SubBytes in circuit blocks.

in this way of design, no multiplexer needed for selection between modes of encryption and decryption, and being able to provide efficient hardware resources for each mode, finally resulting in reduced overall hardware area. To achieve high throughput, the computation for 128-bit AES is subchanneled into four pipelines, with each pipeline handling computation for one row data of 32 bits. Each pipeline conducts the computation for each round by right-shifting the values in each row, while sequentially does SubBytes, MixColumns, and AddRoundKey operations, as depicted in Fig. 2. The entire encryption or decryption process requires a total of 44 clock cycles to complete.

A. SubBytes and Inverse SubBytes

The composite field arithmetic [2] is employed herein to carry out the computation of Sbox, which is more efficient in area than the traditional design as presented in [28] where a look-up table (LUT) is used to carry out SubBytes and InvSubBytes. The composite field arithmetic adopts

$$y = M * X^{-1} \oplus C \tag{1}$$

for the SubBytes module, while

$$x = \left(M' * (Y \oplus C)\right)^{-1} \tag{2}$$

for the InvSubBytes module. Both modules of SubBytes and InvSubBytes share the same designed datapath for GF(2^8) multiplicative inverse. In the proposed design, for encryption, both multiplexers select the calculation results from the lower path as seen in Fig. 3, which are the sequence of performing the inverse affine transform, followed by the multiplicative inverse. Conversely, during decryption, both multiplexers select the calculation results from the upper path in Fig. 3, corresponding to the sequence of first performing the multiplicative inverse and then applying the affine transform for SubBytes. Also noted in Fig. 3 are the muxes accepting "1" or "0" for switching between the datapaths for encryption or decryption, respectively, as seen in blues or greens in Fig. 1.

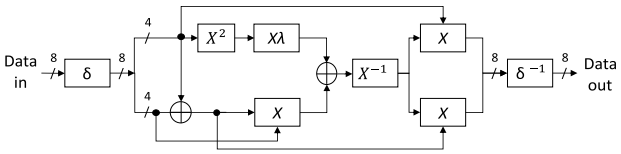

Fig. 4. Computation of multiplicative inverse in blocks.

The muxes designed herein facilitate the always-on operations of encryption and decryption in the designed AES.

The designed AES mathematical operation for multiplicative inverse is mainly on bytes, and uses the calculation method of finite field GF(2^8), where GF represents as Galois Field. Both addition and multiplication are conducted in GF(2^8). This work implements the GF(2^8) multiplicative inverse method where GF(2^8) is converted to GF(2^4)^2 for reducing the complexity of the hardware, and the irreducible polynomials [29] to decompose GF(2^8) to GF(2^4)^2, i.e.,

$$GF(2)^4 : P(X) = X^4 + X + 1 \tag{3}$$

and

$$GF\left(\left(2^4\right)^2\right) : Q(X) = X^2 + X + \lambda \tag{4}$$

where λ is $\{1001\}_2$. The structure of multiplicative inverse is shown in Fig. 4, where δ means isomorphic to GF(2^4), X does the multiplication operation in GF(2^8), and X*λ is the multiplication with constant λ in GF(2^4). The operations of each block in the multiplicative inverse can be presented in terms of bits. The isomorphic mapping is employed to convert the GF(2^8) to GF((2^4)^2), while the inverse isomorphic mapping does GF((2^4)^2) back to GF(2^8). A new operation for multiplicative inversion in GF(2^4) is in fact proposed herein by this study, which uses the Karnaugh map to reduce logic gates to AND, OR, and inverting gates, instead of using XOR only. In this way, GF(2^8) can be converted into GF((2^4)^2). This optimization results in a reduction in gate count from 392 to 352, leading to a more efficient design in terms of area. The

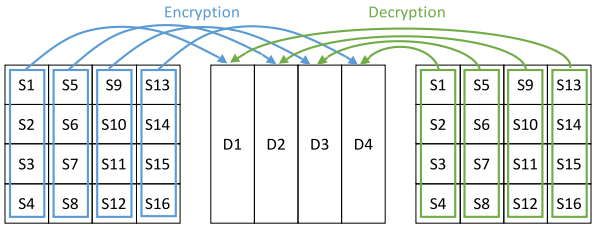

Fig. 5. Proposed method for Shiftrows and Inverse Shiftrows.

computation via GF$(2^4)^2$ is synthesized toward a good balance between chip area and latency.

For the Inverse SubByte computation, the approach of the Multiplicative Inverse with logic reduction is engineered, which allows to share a common circuit unit for multiple inverse multiplications to achieve minimal area. As a result, the area of the S-box is reduced approximately by 10% as compared to the Multiplicative Inverse without the logic reduction method applied. Furthermore, a 32-bit datapath operating simultaneously by 4 S-boxes is designed to carry out both 8-bit conversions of SubBytes and Inverse SubBytes. This design is aimed for enhancing the resistance against attacks that relies on exploiting the correlation between data and power traces. Thus, it is difficult now for attacks to separate measured power to those due to SubBytes and Inverse SubBytes, enhancing security greatly.

B. Shiftrows and Inverse Shiftrows

A novel data storage approach for implementing ShiftRows and inverse ShiftRows is engineered by this work. Fig. 5, illustrates the architecture of this new data storage approach. The 128-bit data is divided into four 32-bit blocks to store, where the encrypted data is stored in blocks D1–D4, while the decrypted data is stored in reverse order, from D4 to D1. Notably, there are MUXes to control how the row data is stored, as seen in blues and greens in Fig. 1 for encryption and decryption, respectively. By adopting this approach, the same results can be obtained using only the ShiftRows operation, regardless of whether the data is encrypted or decrypted. This work effectively reduces the area by 54.5% by sharing hardware between Shiftrows and inverse Shiftrows, compared to designing them separately.

C. Mix Columns and Inverse MixColumns

The Inverse MixColumns operation is decomposed into a matrix multiplication between the MixColumns matrix and a decomposition matrix, which is

$$\begin{bmatrix} 0xe & 0xb & 0xd & 0x9 \\ 0x9 & 0xe & 0xb & 0xd \\ 0xd & 0x9 & 0xe & 0xb \\ 0xb & 0xd & 0x9 & 0xe \end{bmatrix} = \begin{bmatrix} 0x2 & 0x3 & 0x1 & 0x1 \\ 0x1 & 0x2 & 0x3 & 0x1 \\ 0x1 & 0x1 & 0x2 & 0x3 \\ 0x3 & 0x2 & 0x1 & 0x1 \end{bmatrix} \times \begin{bmatrix} 0x5 & 0x4 & 0x0 & 0x0 \\ 0x0 & 0x5 & 0x4 & 0x0 \\ 0x0 & 0x0 & 0x5 & 0x4 \\ 0x4 & 0x0 & 0x0 & 0x5 \end{bmatrix}. \quad (5)$$

As illustrated in Fig. 6, each column's four elements are inputs into the matrix multiplication using the variables a0 to a1 shown in the figure. The operation is conducted for either encryption or decryption according to the selection by a multiplexer. For encryption operation, only the matrix multiplication of the MixColumn matrix following the multiplexer is conducted. As for decryptions, the matrix multiplication with the decomposition matrix is finished first prior to the matrix multiplication of the MixColumn matrix. This approach achieves a gate-level area of 352 gate count and a gate-level delay of 1.31 ns, with the operations of encryption and decryption sharing the same circuit for matrix multiplication of the MixColumn matrix. In comparison, the alternative method, which uses a separate MixColumns matrix and two different decomposition matrices for matrix addition, results in a 10.4% larger area and a 9.1% longer delay.

D. Key Expansion

The key expansion process in the AES algorithm is responsible for deriving the round keys from the initial key. In the 32-bit AES architecture, the first step, SubWord, involves parallel input of four S-boxes, with each S-box being applied for four times. The second step, RotWord, performs byte displacement using hardware connections. The third step, Rcon, utilizes an on-the-fly key architecture that directly interacts with AddRoundKey, eliminating the need for additional memory to store the keys generated in each round for Rcon operations. This work introduces a new design to implement Rcon without the need for additional triggers, as illustrated in Fig. 7(a). This method employs logical functions to calculate the new value of Rcon based on the current Rcon, while multiplexers are utilized to select the Rcon value for either encryption or decryption, resulting in a gate-level area of only 46 Gate count. This approach reduces the area by 50.5% compared to using Xtime logical functions and by 61.7% compared to the traditional LUT-based method. Toward high throughput, the proposed 32-bit AES key expansion architecture operates on 32 bits with four S-boxes within a single clock cycle, as illustrated in Fig. 7(b). Note that this design incorporates right-shifting registers, as shown in Fig. 2, to facilitate AES pipelining, then allowing for the computation of round key in each cycle to avoid the immediate complete expansion of the key, finally easing the requirement of additional storage capacity.

E. Countermeasure Effect on Power Attack

It is pertinent to note herein that both forward and inverse operations for encryption and decryption, respectively, are executed simultaneously for all the tasks, even for each time of operation, it could be for most of times only encryption to decryption being requested. This proposed both-on computation of encryption and decryption pose extreme difficulty for attackers to identify points of interests, let alone to extract keys based on power analysis. In each computational stage, the needed subcircuits are selected from the results of both operations, to conduct both encryption and decryption. This designed architecture of AES with both encryption and

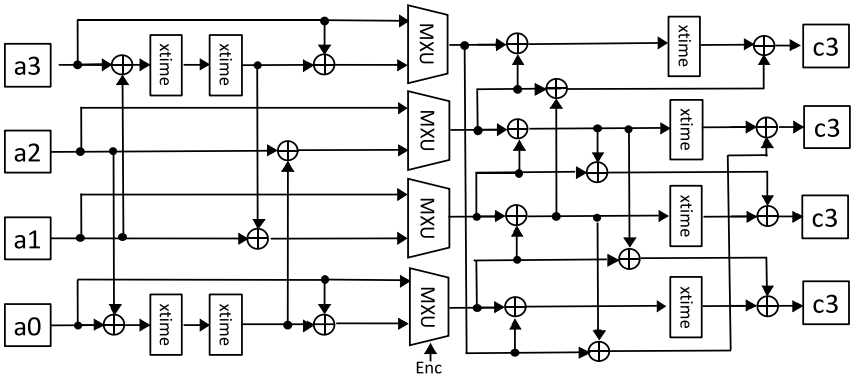

Fig. 6. 32-bit matrix multiplication for MixColumns and Inverse MixColumn.

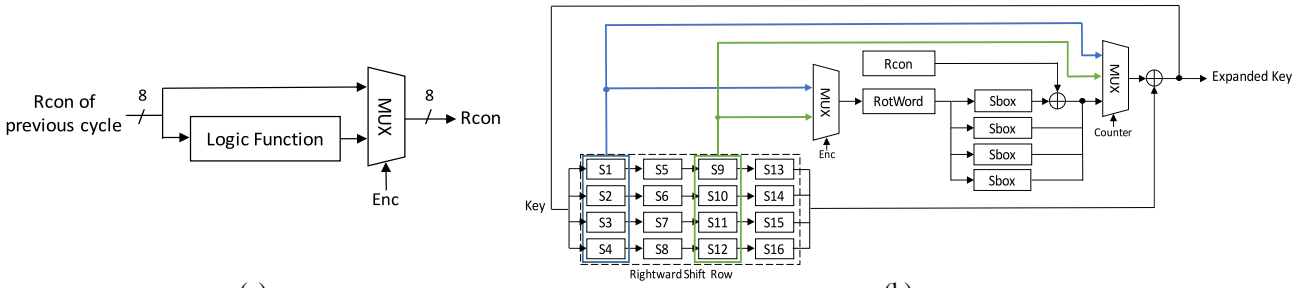

Fig. 7. Computation of (a) Rcon with logic function and (b) proposed key expansion for AES in circuit blocks.

decryption on surely increases power consumption during computation, but it does not significantly raise power consumption due to designed lightweight features of composite field algorithm matrix multiplication for Mix/InvMix-Columns and novel ShiftRow/Inverse-ShiftRow operations. Actually, our design is still quite competitive to the leading group of reported works in power consumption in terms of energy/bit, which will be discussed and evidenced in Section III-D. Furthermore, this computational approach makes it very difficult for attackers to identify points of interests through power analysis, let alone triggering significant circuit branches by determining whether encryption or decryption is being performed, thereby increasing significantly the security level of the designed AES chip.

III. Experimental Results

The afore-proposed AES architecture designed with high throughput-to-area efficiency and strong resilience to attacks is implemented by this study for performance validation.

A. Implementing AES in an FPGA Board

An experimental hardware system as shown in Fig. 8 was built to validate the performance of the proposed AES architecture. The hardware comprises a field programmable gate array (FPGA) board for implementing AES and a mode control unit based on electronic codebook (ECB) [30] to manage input/output data in real time. The plaintext to be encrypted is segmented in blocks via ECB for encryption [31]. To the end of implementing the afore-proposed AES, the designed 32-bit datapath of AES, including SubBytes, ShiftRows,

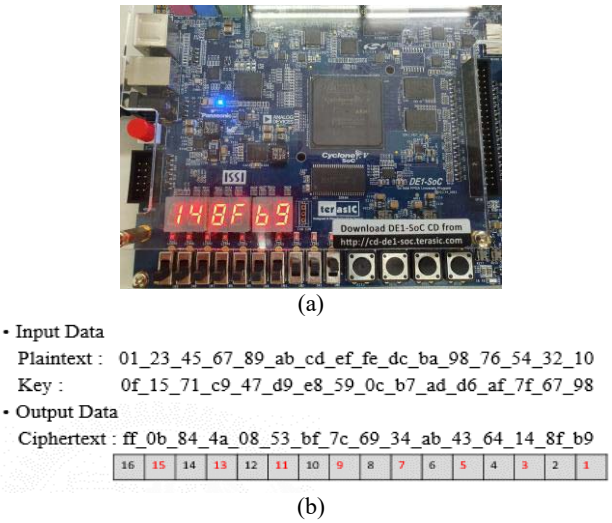

Fig. 8. (a) Displayed output data in 16-bit values on the FPGA board. (b) Example of plaintext and correct cyphertext.

MixColumns, AddRoundKey and KeyExpansion, and their inverse counterparts are synthesized via Verilog and further burnt in to an FPGA board for experiment. Two sets of 128-bit shift registers are synthesized to, respectively, store input plaintexts and other intermediate computed data, as well as keys.

To conduct physical verification of the designed AES circuit using FPGA, this work used first memory to store test signals and data in the test bench and then compare the post-simulation results of the FPGA with the register-transfer level

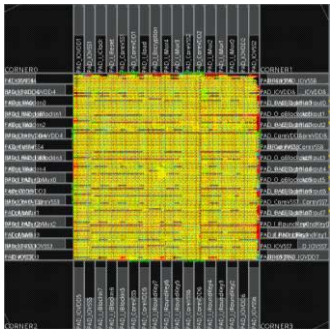

Fig. 9. Chip layout of the designed 32-bit AES datapath.

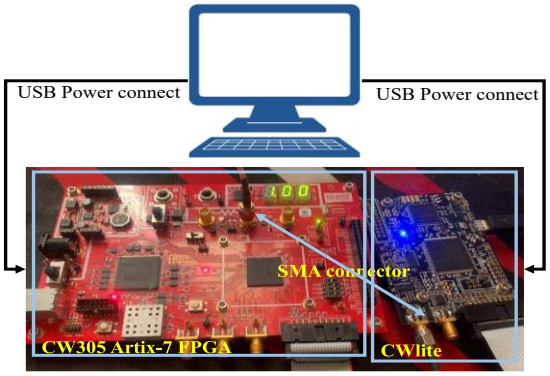

Fig. 10. Experimental setup.

(RTL) simulation results. The initial value of the test data in memory is the input data of the RTL simulation. When the reset signal is activated, data is input to the AES, while the output is displayed on the seven-segment display. The RTL code for the proposed AES is implemented on a Quartus FPGA board. There are six switch buttons to control the output of the FPGA seven-segment display as shown in Fig. 8(a), with the operation results in Fig. 8(b). It has been verified that the ciphertext given by the FPGA is the same with the output result of ModelSim, indicating that the proposed AES algorithm also works correctly for both encryption and decryption in the FPGA board.

B. Chip Implementation and Performance Achieved

The AES chip of encryption and decryption is designed for processing 32-bit data with minimal gate count. The user input interface includes control pins such as encryption/decryption selection, data input and data output selection. The overall architecture consists of the round unit, the Control Unit, the Key Expansion, and the Data Register Unit. The inputs interface includes control pins such as the 1-bit encryption/decryption mode selection, 1-bit data input, 1-bit key input, 1-bit initial vector for reset, 1-bit data output, and the 1-bit output ready signal. All inputs and outputs of the AES designed are integrated into a chip. For synthetization, the Genus synthesis software by Cadence were utilized to implement the designed AES into an FPGA board and further to form a chip. The designed AES architecture was simulated at RTL level using ModelSim to verify the performance of the designed datapath. The RTLs are converted into a gate-level circuit using the TSMC 40-nm Cell-Library to generate timing-delay and netlist files. The simulation results in ModelSim indicate that the proposed AES core can operate at a clock frequency of up to 238.10 MHz without any timing violation. The proposed AES architecture have a gate count of 5.65k and a throughput of 692.65 million bits per second, with a chip area of 13 340 μm^2.

Fig. 9 shows the designed layout of the AES encryption and decryption chip. The core operation part in the layout in Fig. 10 contains the round unit and key expansion. There is a control unit that regulates the core's internal processes in response to external control signals. The datapaths of the round units designed are in 32 or 8 bit, while the input and output pins of the chip are in 8 bits. Note that the serial data were converted in parallel to the round unit for computation, while the data is output serially again once the computations are completed. This work used Synopsys Innovus software for automatic places and routes. While automatic places and routes are finished, the DRC and LVS were conducted for verification. Fig. 10 shows the 32-bit chip layout with a core area of 13 340 um^2. The chip has 68 pins. Among them, eight groups of internal power supplies are used for the chip, while other seven groups of external power supplies are used for I/O pads.

C. Performance Against Attacks

The hardware platform implementing the differential power analysis (DPA) for attacks is shown in Fig. 10. Seen in the photograph in this figure are a red FPGA board CW305 at left with AES implemented inside and under attacks, while another black board CWlite at right as the target board to generate power glitches on CW305. Power traces collected during the process of simultaneous encryption and decryption are captured and analyzed on computer for key guessing. In fact, in order to validate the security of the proposed AES, another unprotected 128-bit AES datapath was also synthesized and implemented in the CW305 board along with the proposed AES in the board for performance comparison in terms of resilience to correlation power attack.

Partial guessing entropy (PGE) is employed as a metric for assessing algorithm security, actually quantifying the number of missing bits before key guessing succeeds. The results are shown in Fig. 11. Evidenced from these figures, the PGE values of the unprotected AES do converge close to zero while the proposed AES does not, as seen clearly from Fig. 11(a) versus (d). Also seen from Fig. 11(b) and (e) are the correlations converged to nonzeros between measured power traces and guessed keys in Fig. 11(b) by the unprotected AES, while to zeros in Fig. 11(e) by the proposed AES datapath. It indicates that the unprotected AES was attacked successfully, while the proposed AES was not, even with 10 000 traces collected. Finally, seen in Fig. 11(c) are significantly high correlations between sample nos. 44 and 47 in traces, indicating a found point of interests enabling effective attacks, while a point of interests cannot be found at all with the proposed AES as seen clearly in Fig. 11(f) where there is no peak standing

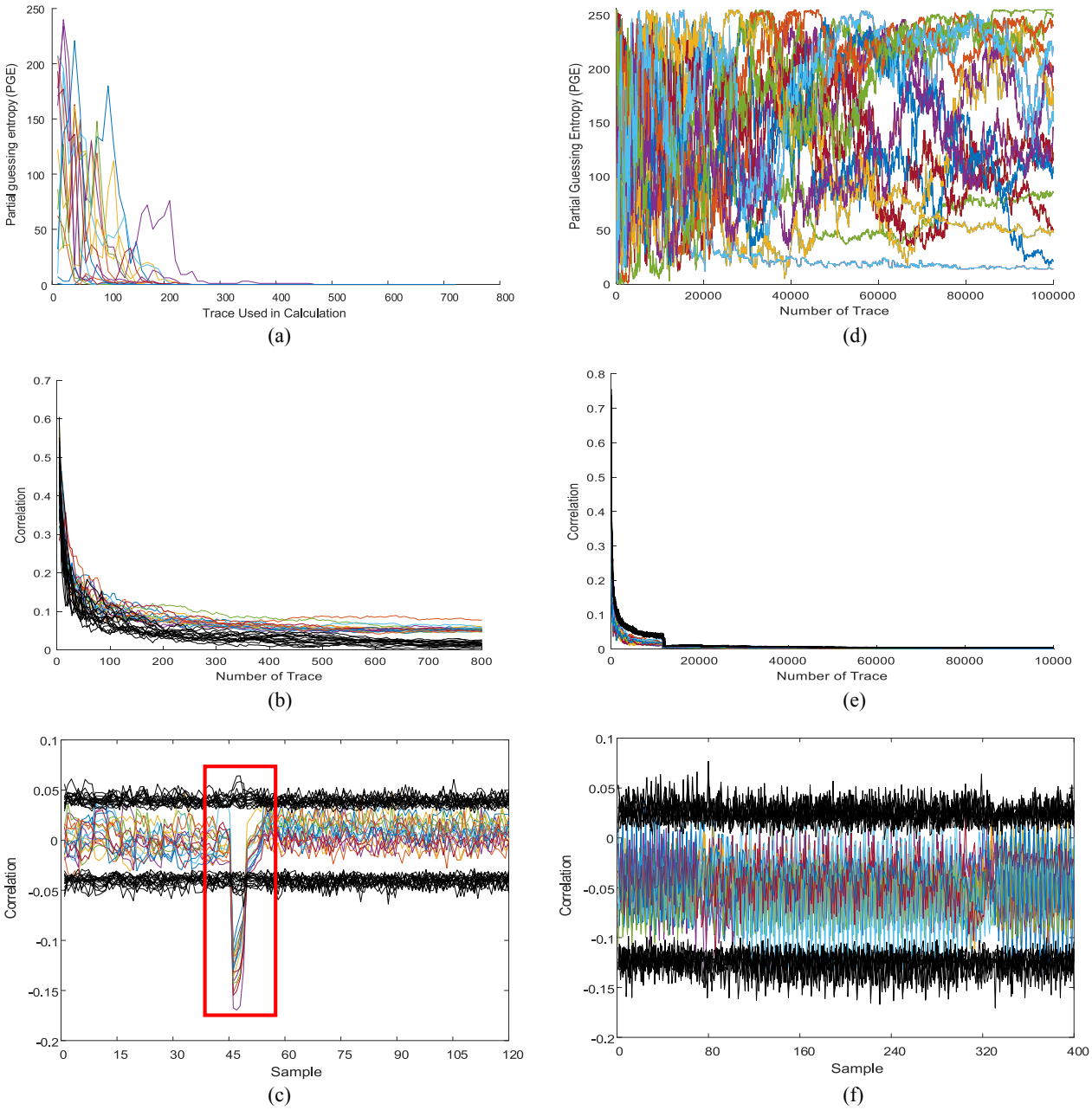

Fig. 11. (a) PGE versus No. of Traces, (b) Correlation versus No. of Traces and (c) Correlation versus Samples by a typical, unprotected AES; (d) PGE versus No. of Traces, (e) Correlation versus No. of Traces, and (f) Correlation versus Samples for the proposed, protected AES.

out of the other correlations, in other words, no chance for successful power attacks. The resilience to attacks shown herein attributes to mixed. parallel computation of encryption and decryption in the AES proposed by this work. In fact, the unprotected AES did surrender to key guessing within 800 power traces with the PGE trend in Fig. 11(a) showing a declining pattern, while the proposed AES showed strong resilience against power attacks even with detecting traces exceeding 100 000, where as seen in Fig. 11(e) the correlation remains zero, still not attacked successfully. The success herein is owing to the extreme difficulty for attacks to separate measured sum power to those due to SubBytes and Inverse SubBytes, enhancing security greatly; in other words, the correlation between power and keys is averaged out from those by SubBytes and Inverse SubBytes before being viciously attacked.

D. Performance Comparison With Others

Performance comparison between the proposed lightweight AES and other prior arts are presented herein via Table I and Fig. 12. Table I provides a comprehensive comparison of performance by the proposed AES and all the related prior works in terms of area, power consumption, throughput, and an FOM defined as throughput divided by area in (Mb/s/k-gate), while Fig. 12 presents the performances from the perspective of FOM versus "energy per bit." Note that in Table I the present work offers two different versions of performance,

TABLE I
PERFORMANCE COMPARISON AMONG THE PROPOSED AND OTHER REPORTED 128-BIT AES DATAPATHS

	Datapath	Board / Tech (nm)	Energy (pJ/bit)	Frequency (MHz)	Clock Cycle	Throughput (Mbps)	Gate count[1] (k-gates)	FOM (Mbps/k-gate)
Satoh *et al.* in 2001 [4]	32	110	-	137	44	398.55	6.3	63.26
Pramstaller *et al* in 2004 [5]	32	60	-	50	92	69.57	8.54	8.15
Mathew *et al.* in 2011 [6]	2-stage pipeline	45	0.511	255	20	1632	100	16.32
Pham *et al.* in 2012 [7]	128	Cyclone II	5.84	78.39	160	28.5	1.108	25.72
Wang *et al.* in 2013 [12]	128	Virtex-6	-	14.69	1	1880	182.60	10.30
Banik *et al.* in 2015 [8]	32	90	6.2	10	46	27.83	5.5	5.06
Mathew *et al.* in 2015 [9]	8	22	3.9	1113	336	432	4.04	106.93
Kouser *et al.* in 2016 [13]	128	Spartan-6	-	239.96	41	749.14	68.33	10.96
Bui *et al.* in 2016 [10]	64	65	3.54	10	37	16.5	2.11	7.82
Bui *et al.* in 2017 [21]	32	28	0.65-0.8	10	44	29.09	8.6	3.38
Kim *et al.* in 2019 [22]	32	65	0.78	10	186	6.88	5.4	1.27
Jain *et al.* in 2019 [14]	128	Artix-7	-	273.29	44	795.03	16.12	49.32
Dong *et al.* in 2019 [24]	128	45	0.51	870	11	10123	164.5	61.54
Ruby *et al.* in 2020 [11]	128	Virtex-7	-	0.258	129	33.07	6.50	5.09
Pandey *et al.* in 2020 [15]	128	180	4.42	200	73	289.34	53	5.46
Lata *et al.* in 2020 [16]	8	Spartan-6	-	272.33	-	3485	59.20	58.87
Kundi *et al.* in 2020 [26]	32	Virtex-6	4.32	332.98	20	1856	37.80	49.10
Chong *et al.* in 2021 [25]	128	Kintex-7	**0.23**	12	10	153.6	17	9.04
Scripcariu *et al.* in 2021 [17]	128	Virtex-6	-	60.02	10	700	25.468	9.04
Gunasekaran *et al.* in 2021 [18]	32	Virtex-7	4.3	161	74	278.49	13.70	20.33
Davis *et al.* in 2022 [23]	32	90	0.66	20	61	41.97	**0.83**	27.49
Lee *et al.* in 2022 [19]	32	Kintex-7	-	333	-	805	5.18	50.87
Lee *et al.* in 2022 [20]	32	16	-	526	94	716.26	8.55	83.77
Ng *et al.* in 2022 [27]	128	Kintex-7	7.10	22	11	256	2.57[2]	99.61
The proposed chip	32	Kintex-7	2.97	166.49	44	493.06	5.34	92.33
	32	40	1.16	238.10	44	**692.65**	**5.65**	**122.59**

[1] The gate counts extracted from each of the prior works may vary, depending on the respective baseline designs.
[2] Converted from the number of slices without LUTs and FFs.

by the Kintex-7 FPGA board and the TSMC 40-nm process. Note also that the column regarding energy consumptions in pJ/bit by the present work and all the prior arts are estimated in adequate precision by the design and verification tool of the FPGA board and that of the adopted process, to arrive at fair performance evaluation and comparison. A quick analysis on Table I reveals that Davis and John [23] presented the smallest AES in 2022, utilizing a mere 0.83K gate count. Chong et al. [25] achieved the lowest energy consumption with their AES, demonstrating an energy per bit of only 0.23 pJ/bit. Different from these studies, this work not only emphasizes area and power consumption but also aiming for a meaningful balance among area, power consumption, and throughput for IoT devices, as aimed also by Kundi et al. [26]. The degree of this balance can be quantified by an FOM as (Mbps/k-gates), i.e., the throughput over area. It evidenced from Table I and Fig. 12 that the proposed AES synthesized by the TSMC 40-nm process stands out of all the others synthesized by

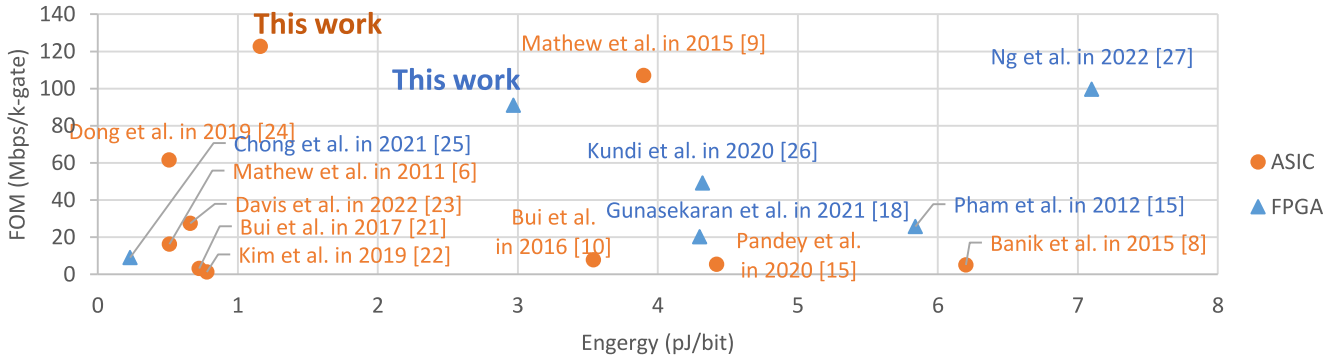

Fig. 12. Comparison of FOM versus energy per bit among AESes.

either some FPGA boards or processes, as the best with the highest FOM as 122.59 Mb/s/k-gate, and very small area of 5.65 k-gates while comparatively lower power in energy per bit, underscoring its exceptional performance. Focusing on the performances only by FPGA boards, this work implemented in Kintex-7 shows also excellent performance, second only to that by Ng et al. [27]. However, the AES datapath by Ng et al. [27] was successfully attacked within 5000 power traces, as opposed to more than 100 000 required for the AES proposed by this study, thanks to the unique design of mixed, parallel computation of encryption and decryption in this proposed AES, as detailed in Section III-C. In addition, the proposed AES achieves 193% more in throughput and a 42% further reduction in energy per bit as compared to [27].

IV. Conclusion

A novel low-cost lightweight AES is proposed by this effort, optimized successfully with high efficiency for lightweight applications in IoT devices. The proposed AES architecture is designed for 32-bit encryption in parallel for the 128-bit AES to reduce hardware area. The optimization is achieved via using shift registers instead of plain registers in the operation stage of ShiftRows to optimize the 32-bit datapath AES, ShiftRows, MixColumns, and key expansion. This work was implemented on an Intel Altera FPGA board and by the TSMC 40-nm Cell-Library, achieving a throughput of 692.65 Mb/s, a gate count of 5.65K, and a FOM defined as the ratio of throughput to area of 122.59 Mb/s/k-gate, with a chip area of 13340 μm^2, which is the best over all the reported works. As for security evaluation, the proposed AES with both computations of encryption and decryption on safeguards successfully the correlation power analysis attacks even with 100 000 traces collected.

References

[1] W. Saunders, "KDOQI clinical practice guidelines and clinical practice recommendations for diabetes and chronic kidney disease," *Am. J. Kidney Dis.*, vol. 49, no. 2, pp. 12–154, 2007.

[2] X. Zhang and K. K. Parhi, "On the optimum constructions of composite field for the AES algorithm," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 10, pp. 1153–1157, Oct. 2006.

[3] S. M. Farhan, S. A. Khan, and H. Jamal, "Mapping of high-bit algorithm to low-bit for optimized hardware implementation," in *Proc. 16th Int. Conf. Microelectron.*, 2004, pp. 148–151.

[4] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-box optimization," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2001, pp. 239–254.

[5] N. Pramstaller, S. Mangard, S. Dominikus, and J. Wolkerstorfer, "Efficient AES implementations on ASICs and FPGAs," in *Proc. Int. Conf. Adv. Encrypt. Stand.*, 2004, pp. 98–112.

[6] S. K. Mathew et al., "53 Gbps native GF$(2^4)^2$ composite-field AES-encrypt/decrypt accelerator for content-protection in 45 nm high-performance microprocessors," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, pp. 767–776, Apr. 2011.

[7] T. A. Pham, M. S. Hasan, and H. Yu, "Area and power optimisation for AES encryption module implementation on FPGA," in *Proc. 18th Int. Conf. Autom. Comput. (ICAC)*, 2012, pp. 1–6.

[8] S. Banik, A. Bogdanov, and F. Regazzoni, "Exploring energy efficiency of lightweight block ciphers," in *Proc. Int. Conf. Sel. Areas Cryptogr.*, 2015, pp. 178–194.

[9] S. Mathew et al., "340 mV–1.1 V, 289 Gbps/W, 2090-gate NanoAES hardware accelerator with area-optimized encrypt/decrypt GF (2 4) 2 polynomials in 22 nm tri-gate CMOS," *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 1048–1058, Apr. 2015.

[10] D.-H. Bui, D. Puschini, S. Bacles-Min, E. Beigné, and X.-T. Tran, "Ultra low-power and low-energy 32-bit datapath AES architecture for IoT applications," in *Proc. Int. Conf. IC Design Technol. (ICICDT)*, 2016, pp. 1–4.

[11] A. M. Ruby, S. M. Soliman, and H. Mostafa, "Dynamically reconfigurable resource efficient AES implementation for IoT applications," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2020, pp. 1–5.

[12] Y. Wang and Y. Ha, "FPGA-based 40.9-Gbits/s masked AES with area optimization for storage area network," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 1, pp. 36–40, Jan. 2013.

[13] Z. Kouser, M. Singhal, and A. M. Joshi, "FPGA implementation of advanced encryption standard algorithm," in *Proc. Int. Conf. Recent Adv. Innov. Eng. (ICRAIE)*, 2016, pp. 1–5.

[14] N. Jain, D. S. Ajnar, and P. K. Jain, "Optimization of advanced encryption standard algorithm (AES) on field programmable gate array (FPGA)," in *Proc. Int. Conf. Commun. Electron. Syst. (ICCES)*, 2019, pp. 1086–1090.

[15] J. G. Pandey, S. Gupta, and A. Karmakar, "A unified architecture for AES/PRESENT ciphers and its usage in an SoC environment," in *Proc. IEEE 11th Latin Amer. Symp. Circuits Syst. (LASCAS)*, 2020, pp. 1–4.

[16] K. Lata and S. Saini, "Hardware software co-simulation of an AES-128 based data encryption in image processing systems for the Internet of Things environment," in *Proc. IEEE Int. Symp. Smart Electron. Syst. (iSES) (Formerly iNiS)*, 2020, pp. 260–264.

[17] L. Scripcariu, D. Burdia, and F. Diaconu, "FPGA synthesis of an AES encoder circuit for vehicular communication networks," in *Proc. Int. Symp. Signals, Circuits Syst. (ISSCS)*, 2021, pp. 1–4.

[18] M. Gunasekaran, K. Rahul, and S. Yachareni, "Virtex 7 FPGA implementation of 256 bit key AES algorithm with key schedule and sub bytes block optimization," in *Proc. IEEE Int. IOT, Electron. Mechatron. Conf. (IEMTRONICS)*, 2021, pp. 1–6.

[19] U. Lee, H. K. Kim, Y. J. Lim, and M. H. Sunwoo, "Resource-efficient FPGA implementation of advanced encryption standard," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2022, pp. 1165–1169.

[20] J. S. Lee, P. Choi, and D. K. Kim, "Lightweight and low-latency AES accelerator using shared SRAM," *IEEE Access*, vol. 10, pp. 30457–30464, 2022.

[21] D.-H. Bui, D. Puschini, S. Bacles-Min, E. Beigné, and X.-T. Tran, "AES datapath optimization strategies for low-power low-energy multisecurity-level Internet-of-Things applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 12, pp. 3281–3290, Dec. 2017.
[22] H. K. Kim and M. H. Sunwoo, "Low power AES using 8-bit and 32-bit datapath optimization for small Internet-of-Things (IoT)," *J. Signal Process. Syst.*, vol. 91, nos. 11–12, pp. 1283–1289, 2019.
[23] C. Davis and E. John, "Shared round core architecture: A novel AES implementation for implantable cardiac devices," in *Proc. IEEE 65th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, 2022, pp. 1–4.
[24] P.-K. Dong, H. K. Nguyen, and X.-T. Tran, "A 45nm high-throughput and low latency AES encryption for real-time applications," in *Proc. 19th Int. Symp. Commun. Inf. Technol. (ISCIT)*, 2019, pp. 196–200.
[25] K.-S. Chong et al., "Dual-hiding side-channel-attack resistant FPGA-based asynchronous-logic AES: Design, countermeasures and evaluation," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 11, no. 2, pp. 343–356, Jun. 2021.
[26] D.-E.-S. Kundi, A. Khalid, A. Aziz, C. Wang, M. O'Neill, and W. Liu, "Resource-shared crypto-coprocessor of AES Enc/Dec with SHA-3," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 4869–4882, Dec. 2020.
[27] J.-S. Ng et al., "An asynchronous-logic masked advanced encryption standard (AES) accelerator and its side-channel attack evaluations," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2022, pp. 2256–2260.
[28] S. Heron, "Advanced encryption standard (AES)," *Netw. Secur.*, vol. 2009, no. 12, pp. 8–12, 2009.
[29] J. Wolkerstorfer, E. Oswald, and M. Lamberger, "An ASIC implementation of the AES SBoxes," in *Proc. Cryptogr. Track RSA Conf.*, 2002, pp. 67–78.
[30] M. J. Dworkin, "Recommendation for block cipher modes of operation: Methods and techniques," U.S. Dept. Comm., Nat. Instit. Stand. Technol., Rep. TR-800-38A, 2001.
[31] V.-L. Dao, V.-P. Hoang, A.-T. Nguyen, and Q.-M. Le, "A compact, low power AES core on 180nm CMOS process," in *Proc. Int. Conf. IC Design Technol. (ICICDT)*, 2016, pp. 1–5.

Pao-Ying Cheng received the bachelor's degree from the Department of Electronics Engineering, National Yang-Ming Chiao Tung University, Hsinchu, Taiwan, in 2021, where she is currently pursuing the Ph.D. degree with the Department of Electrical and Control Engineering.

Her research interests encompass hardware security, cryptographic algorithm architectures and countermeasures, physical unclonable functions, and side-channel analysis.

Ying-Cheng Su received the B.S. degree from the Department of Electronic Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 2017, and the M.S. degree from the Institute of Electrical and Computer Engineering, National Chiao Tung University in 2019.

He is working with mediatek, Hsinchu. His research areas are Advanced Encryption Standard architecture design, VLSI design and architecture, and side-channel analysis.

Paul C.-P. Chao (Fellow, IEEE) received the Ph.D. degree from Michigan State University, East Lansing, MI, USA, in 1997.

He was with Chrysler Corp., Detroit, MI, USA, before he joined National Yang-Ming Chiao-Tung University, Hsinchu, Taiwan, where he is currently a University Distinguished Professor with the Electronics and Electrical Engineering Department, and a Distinguished Lecturer for IEEE Sensors Council from 2018 to 2020. He has published more than 450 peer-reviewed papers (books, journal papers, conferences, and reports) and 38 patents. His research interests focus on sensors, actuators, and their interface circuitry.

Dr. Chao was the recipient of the 1999 Arch T. Colwell Merit Award from the Society of Automotive Engineering, Detroit; the 2019 Technical Achievement Award, IEEE Sensors Council; the 2017 Presidential Outstanding Professor of Engineering in Nation (Taiwan) (awarded by the President of the nation in the Presidential House of Taiwan); two 2017 Future Technology Awards (Taiwan Oscar Invention Award) from the Ministry of Science and Technology (MOST), Taiwan Government; the 2018 Outstanding Professor of Electrical Engineering in Nation (Taiwan), National Association of Electrical Engineering, Taiwan; the Outstanding research Award of MOST, 2018; the Technical Achievement Award, Sensors Council, IEEE, USA, in 2019; the Distinguished Institutional Award, American Society of Mechanical Engineering (ASME), USA, in 2019; and the Best Paper Award of the ASME ISPS Conference 2019. He also received the Award of the 2020 Best Topical Editor, IEEE SENSORS JOURNAL. He has served as the University Associate Vice President for NYCU for Academic Affairs from 2009 to 2010, and Research and Development in 2015. He is currently the Topical Editor of the IEEE SENSORS JOURNAL and IEEE INTERNET OF THINGS JOURNAL and the Associate Editor-in-Chief of the IEEE SELECTED TOPICS IN SENSORS. He is a Fellow of ASME.