Small Footprint Mix-Column Serial for PHOTON and LED Lightweight Cryptography

Yasir Amer Abbas
Department of Computer Engineering
University of Diyala
Diyala, Iraq
yasiramerabbas@gmail.com

Razali Jidin
College of Engineering
Universiti Tenaga Nasional
Selangor, Malaysia
Razali@uniten.edu.my

Norziana Jamil
Institute of Informatics and Computing in Energy
Universiti Tenaga Nasional
Selangor, Malaysia
Norziana@uniten.edu.my

Muhammad Reza Z'aba
Faculty of Computer Science and Information Technology
University of Malaya
Kuala Lumpur, Malaysia
reza.zaba@um.edu.my

Saad Al-Azawi
Department of Electronic Engineering
College of Engineering, University of Diyala
Diyala, Iraq
saad.alazawi@engineering.uodiyala.edu.iq

***Abstract*—Lightweight cryptography such as PHOTON or LED has a transform named as Mix-Column-Serial (MCS). Within the MCS, matrix manipulations use Galois polynomial multiplications that require lengthy steps of logical operations. This paper proposes the use of a look-up table with comparators to replace the lengthy steps. As PHOTON's Galois matrix multiplication produces identical results for pairs of column-row and row-columns, with comparators, the table size is reduced to half. The tables and comparators have been implemented on FPGAs. FPGA's synthesized results of the newly proposed MCS in the form table with comparators are superior in terms of throughput and area compared to other MCS hardware implementations found in literatures.**

Keywords*— *Lightweight cryptography, MixColumns, PHOTON, LED, FPGA

I. Introduction

Confidentiality can be achieved by encrypting important messages, while integrity is possible by employing hash algorithms. Message or plaintext and keys are inputs to an encryption algorithm to produce a cipher-text. Both the hash and encryption are parts of knowledge in the domain of cryptography. Recent research works on cryptography are the lightweight version that targeted for resource constraint devices. The lightweight version cryptographies are cipher algorithms that suitable for hardware implementations [1][2].

Research on lightweight cryptography are quite recent [3][4]. Research works on lightweight for constraint devices such as RFID works in [5] that have adopted block cipher PRESENT for hash function. In [6] [7], the focus have been lightweight hash functions that based on a SPONGE (SPNG) construction [8], resulting to two hash functions: QUARK and PHOTON algorithms. The existing research works on lightweight cryptography focus mainly on the less complex computations, to reduce area and power by having smaller data width and key sizes. The key sizes and number of round computations influence the strength of a cipher algorithm. Lightweight encryption or hash has algorithms that suit implementation on hardware such as field programmable logic arrays (FPGA). Examples of FPGA implemented hash functions optimized for small footprint and high throughput are SHA-3 [9] [10] and Keccak [11]. In [12] , different hardware architectures have been deliberated to minimize areas, while achieving high speed, however, compromises being made for throughput and memory footprint. Less intensive computing ciphers such as LED encryption block cipher [13] and PHOTON hash function families [7] have been proposed in 2011. Both LED and PHOTON have four transformations including one known as mix-columns-serial (MCS). The MCS consist of 4-bit arithmetic operation that include multiplication of matrix distance separation (MDS) [14].

In this paper a new Mix-Column are designed uses a simple table that contains all results of Galois multiplication, followed by the use of simple comparator sets. The proposed Mix-Column has superior throughput compared to earlier previous designs described, for LED and PHOTON implementation targeted for Spartan-3 and Artix-7 FPGA.

Our paper is organized as follows; in Section 2 the related work was discuss. Subsequently, in Section 3 the PHOTON and LED proposed architecture for matrix multiplication of Galois field. In the Section 4 our implementation is described, and then the results are presented and compared to FPGA implementations for different algorithms. Finally, in Section 5 this paper is concluded.

II. Related Work

Both algorithms are based on the well-established SPONGE structure [15] and supports different parameters to achieve different levels of security. Guo et al. have proposed the lightweight family of hash function PHOTON [7], targeted for ASICs. There are five types of PHOTON hash function families with digest sizes of 80, 128, 160, 224 and 256 bits.

978-1-5386-6696-8/18/$31.00 ©2018 IEEE

The PHOTON internal permutation has twelve round computations, almost similar to AES. PHOTON's internal state is represented as a (d × d) matrix, where d is 4,5,6,7 and 8 with element of 5-bit cells. Each round consists of four transformations: Add-Constant, Sub-Cells, Shift-Rows and Mix-Column-Serial. The work in [12] has investigates three PHOTON architectures targeted for low cost FPGA implementation or compact design for the five different flavours of PHOTON hash. The first architecture performs one round computation per clock cycle, while the second type is based on the architecture presented as in [7] for ASIC implementation. The third architecture realizes PHOTON computation based on shift registers such as SRL16 of one FPGA's resources. The third approach is better than the second one because it provides lower area design and better throughput performance. However, the design of mix-column has been based on shift register that requires a number of clock cycles to produce mix-column results. Many hashing algorithm use Mix-Column-Serial or simply known as Mix-Column transform with various implementation architectures on FPGA.

In this paper, a new Mix-Column uses a simple table that contains all results of Galois multiplication, followed by the use of simple comparators to select one of multiplication results. The new design addresses the most costly operation of hash function design that is multiplication. The proposed design replaces the lengthy steps of Galois polynomial matrix multiplication with the look-up table and comparators. The size of the table can be reduced to half since PHOTON-MCS matrix multiplication produces similar products for column-row and row-column pairs. The small size table with comparators improves the performance of Mix-Columns within both the PHOTON hash function and LED block cipher in terms of speed, cost and power consumption mmaintaining the Integrity of the Specifications

III. PHOTON AND LED : PROPOSED ARCHITECTURE FOR MATRIX MULTIPLICATION OF GALOIS FIELD

The transforms within LED or PHOTON as shown in Fig.1 are Add-Constant (AC), Sub-Cells (SC), Shift-Rows (SR), and Mix-Columns (MC). In general, Add-Constant performs addition of fixed values to the cells of the internal state or input, while Sub-Cell applies an 4-bit substitute value to each of them. Shift-Row rotates the position of the cells in each of the rows, and Mix-Column linearly mixes all the columns independently.

The original Mix-Column is a transformation based on Galois Field (GF) multiplication. Each 4 bits of a column is replaced with another value that is a function of all 4 bits in a given column. The Mix-Columns transformation operates on the state column-by-column, treating each column as a four-term polynomial. The columns are considered to be polynomials over GF (2^4) and modulo operation with $X^4+ X+1$ polynomial. For multiplication of matrices in Mix-Column, it is polynomial multiplication between two (4-bit) numbers that produces the output result of 8-bits. To satisfy Galois field, if the results are 4-bit (0 to 3), then the results can be taken directly, whereas if the results are larger than the polynomial order, then the result need to be XORed with 10011 repeatedly until the results become 4-bits or less.

The proposed approach to realize Mix-Column is to use a look-up table with comparator sets, instead of combinations of multipliers, XORs and irreducible polynomials operation, to obtain the results of any two matrix multiplication. The comparator reduces the number of mathematical and logic operations and hence provides high speed, low power consumption, acceptable area, high throughput and low complexity. The steps of the proposed MCS's comparators are as follows:

1. For any number from (0-F) multiplied by zero, the result will be zero. This step is performed by only one comparator operation.

2. For any number from (0-F) multiplied by one, the result will be the first operand number directly. This step is also performed by using one comparator operation only.

3. Multiplication of number (2) by (0-F) requires only 14 comparator operations because the multiplications of (2) by 0 and 1 have been performed by steps 1 and 2 above.

4. As above, number (3) multiplied by (0-F) requires 13 comparator operations because the multiplication of (3) by 0, 1 and 2 have been performed in steps 1,2 and 3 above, and so on.

5. Finally, multiplication of number (F) by (0-F) requires only one comparator operation because multiplications by (0-E) have been performed in previously.

Next, two matrix of Mix-Column is converted to an array vector depending on the location of every element. The comparators replace the two 4-bit matrix elements that to be multiplied. The comparators select the appropriate look-up table contents. Five outputs from the table are summed (logical XOR) to form an element of product matrix.

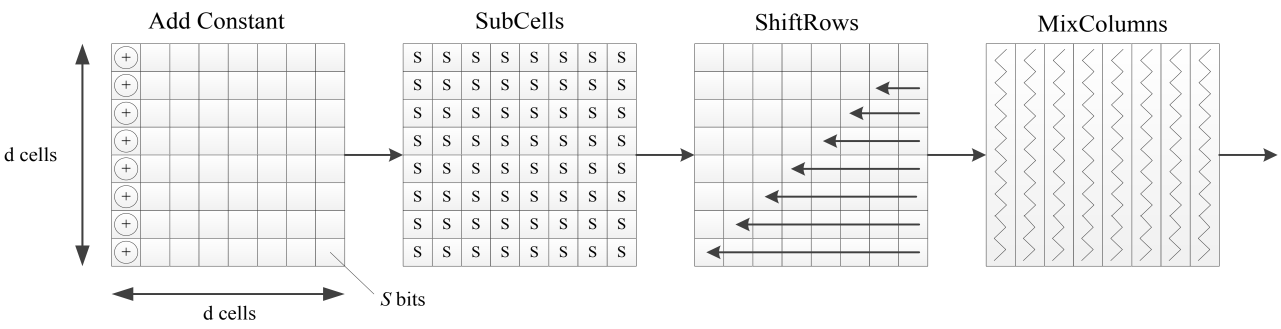


Fig. 1. Four Main Functions within LED and PHOTON algorithms [7]

The look-up table with comparators is to replace MCS's multiplication of two matrixes, as depicted in Fig 2. Therefore total comparators to realize multiplication of two 5×5 matrix with each 4-bit element is: [1+1+14+13+......) × 25 = 3750 comparators. For example to find C4 value there is five multiplication operation and four addition, each multiplication requires 30 comparators, therefore (30 × 5 × 25)= 3750 comparators.

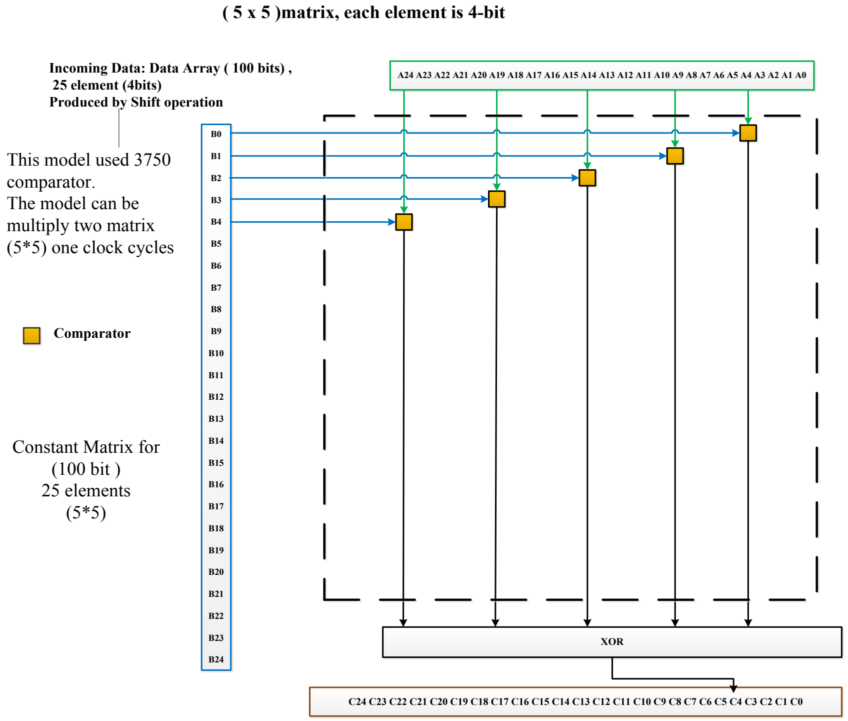
If to implement multiplication of 5x5 matrix within the Mix-Column, take one element, for example one product such as (A4.B4), using conventional approach, sixteen AND gates, nine XOR and three Irreducible Division are needed. Therefore, C4 requires five times of (sixteen AND gates, nine XOR and three Irreducible Division). For the newly proposed 4-bits multiplication method, C4 requires five comparator sets only, instead of eighty AND gates, forty-nine XOR and fifteen Irreducible Division.

A: Input (coming from Shift-Row) B: Constant

$$\begin{pmatrix} A4 & A3 & A2 & A1 & A0 \\ A9 & A8 & A7 & A6 & A5 \\ A14 & A13 & A12 & A11 & A10 \\ A19 & A18 & A17 & A16 & A15 \\ A24 & A23 & A22 & A21 & A20 \end{pmatrix} * \begin{pmatrix} B4 & B3 & B2 & B1 & B0 \\ B9 & B8 & B7 & B6 & B5 \\ B14 & B13 & B12 & B11 & B10 \\ B19 & B18 & B17 & B16 & B15 \\ B24 & B23 & B22 & B21 & B20 \end{pmatrix} =$$

C: Output Mix-Columns

$$\begin{pmatrix} C4 & C3 & C2 & C1 & C0 \\ C9 & C8 & C7 & C6 & C5 \\ C14 & C13 & C12 & C11 & C10 \\ C19 & C18 & C17 & C16 & C15 \\ C24 & C23 & C22 & C21 & C20 \end{pmatrix}$$


(5 x 5)matrix, each element is 4-bit
Incoming Data: Data Array (100 bits) ,
25 element (4bits)
Produced by Shift operation
This model used 3750 comparator.
The model can be multiply two matrix (5*5) one clock cycles
Comparator
Constant Matrix for (100 bit) 25 elements (5*5)
A24 A23 A22 A21 A20 A19 A18 A17 A16 A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
B0
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
B19
B20
B21
B22
B23
B24
XOR
C24 C23 C22 C21 C20 C19 C18 C17 C16 C15 C14 C13 C12 C11 C10 C9 C8 C7 C6 C5 C4 C3 C2 C1 C0

Where:

A_0 to A_{25} is incoming data produced by shift operation; every element is 4-bit.

B_0 to B_{25} is Constant data; every element is 4-bit.

C_0 to C_{25} is Results data produced after Mix-Columns operation; every element is 4-bit.

Fig. 2. New MixColumns Architecture for LED and PHOTON

IV. SYNTHESIS RESULTS OF PROPOSED MIX-COLUMN ARCHITECTURE

The new Mix-Column has been designed with loop unrolling data-path, targeted for implementation on FPGA. The proposed MCS architecture starts with conversion of the two matrixes to two arrays of data. The two matrixes: the first matrix is the output of Shift-Row, and the second matrix is minimum distance separation (MDS) or constant matrix. The size of every element in these two arrays is 4bits. The RTL top-level design for Mix-Column is shown in the **Error! Reference source not found.**3, utilizing VHDL as programming language.

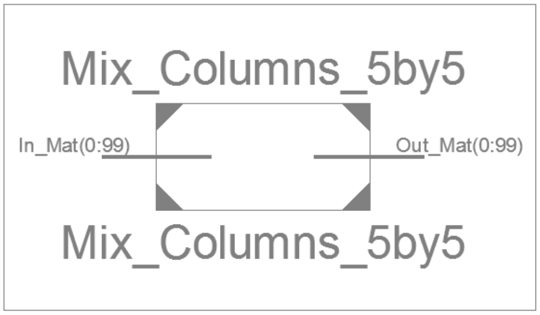


Fig. 3. RTL Top Design for Mix-Column-Serial (MCS)

The detail of architecture of the new Mix-Column is illustrated by Fig. 2. Table I shows the slice number for proposed design with different matrix sizes of PHOTON capacity.

The proposed MCS consists of set of comparators to achieve 4-bit Galois field multiplication. This comparator replaces the complex 4-bit binary multiplication. The number of slices for the proposed comparator is 12 slices only. The Mix-Column is configurable for four matrix sizes (5, 6, 7 and 8 elements) that to handle variable data-path width such as 100, 144, 196 and 256 bits.

TABLE I: THE SLICE NUMBER FOR THE PROPOSED MIXCOLUMNS WITH SPARTAN-3

Proposed	Matrix Size	Bit Size	Slices
PHOTON-80/20/16	5×5	100	165
PHOTON-128/16/16	6×6	144	282
PHOTON-160/36/36	7×7	196	478
PHOTON-224/32/32	8×8	256	840

The new Mix-Column produces results in one clock cycle. The synthesis of new Mix-Column for both LED and PHOTON lightweight cryptography algorithms yield higher efficiency compared to previous works as listed in Table II.

TABLE II: COMPARISON OF SYNTHESIS RESULTS OF PROPOSED MIX-COLUMN

MixColumns	**FPGA Board**	**Slices**	**Max. Freq. (MHz)**	**Thro/put (Mbps)**	**Efficiency (Mbps/Slice)**
Proposed PHOTON-80	Spartan3	165	93.13	9313	56.44
PHOTON-80 [12]	Spartan3	112	-	6.57	0.055
Proposed PHOTON-80	Artix7	66	499	49900	756.06
PHOTON-80 [12]	Artix7	58	-	18.33	0.32
Proposed LED-64	Spartan3	79	95.82	6132	77.62
LED-64 [12]	Spartan3	77	-	9.93	0.13
Proposed LED-64	Artix7	40	532	34060	851
LED-64 [12]	Artix7	40	-	22.93	0.57

CONCLUSION

The new Mix-Column uses a simple table that contains all results of Galois multiplication, followed by the use of simple comparator sets. The comparators are to select stored multiplication results. Results show that proposed design of Mix-Column is able to produce product matrix in one clock cycle. The proposed Mix-Column has superior throughput compared to earlier designs described in literatures, for PHOTON implementation targeted for Spartan-3 and Artix-7 FPGA. The simulation results show significant improvements of hardware efficiency for all data-path structures. Moreover, results show further improvement gain in throughput and efficiency by 344 % and 234 %, respectively, compared with the previous designs of PHOTON-80/20/16 with implementations on Spartan-3. A future work for the proposed Mix-Columns using Galois filed multiplication can implement with AES or SHA3 algorithms.

ACKNOWLEDGMENT

The authors would like to extend their gratitude to Ministry of Higher Education Malaysia (MOHE) for their FRGS grant no: FRGS/1/2015/ICT06/UNITEN/02/1 and to the Innovation and Research Management Centre (iRMC) of Universiti Tenaga Nasional (UNITEN) for their support. Also, authors would like to express their great thanks to College of Engineering, University of Diyala, Iraq for their great support.

REFERENCES

[1] C. Paar and J. Pelzl, Understanding Cryptography A Textbook for Student and Practitioners. Springer, 2010.

[2] Y. A. Abbas, R. Jidin, N. Jamil, M. R. Z'aba, M. E. Rusli, and B. Tariq, "Implementation of PRINCE algorithm in FPGA," in Proceedings of the 6th International Conference on Information Technology and Multimedia, 2014, pp. 1–4.

[3] M. Katagi and S. Moriai, "Lightweight Cryptography for the Internet of Things," Sony Corp., pp. 7–10, 2008.

[4] Y. A. Abbas, R. Jidin, N. Jamil, and M. R. Z'aba, "Securing Electrical Substation's Wireless Messaging with a Lightweight Crypto-Algorithm IP Core," in IEEE International Conference Power & Energy (PECON), 2014, pp. 159–163.

[5] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and G. Horst, "Hash Functions and RFID Tags : Mind the Gap," in In Cryptographic Hardware and Embedded Systems–CHES 2008, Springer Berlin Heidelberg, 2008, pp. 283–299.

[6] J. Aumasson, L. Henzen, and W. Meier, "Quark : a lightweight hash," J. Cryptol., vol. 26, no. 2, pp. 313–339, 2013.

[7] J. Guo, T. Peyrin, and A. Poschmann, "The PHOTON Family of Lightweight Hash Functions," pp. 222–239, 2011.

[8] G. Bertoni, J. Daemen, and G. Van Assche, "On the Indifferentiability of the Sponge Construction," in Advances in Cryptology – EUROCRYPT 2008, Springer, 2008, pp. 181–197.

[9] B. Jungk, "Evaluation Of Compact FPGA Implementations For All SHA-3 Finalists," in The Third SHA-3 Candidate Conference, 2012.

[10] J. Kaps, P. Yalla, and K. K. Surapathi, "Lightweight Implementations of SHA-3 Finalists on FPGAs," in The Third SHA-3 Candidate Conference., 2012, no. 60, pp. 1–17.

[11] E. B. Kavun and T. Yalcin, "A Lightweight Implementation of Keccak Hash Function for Radio-Frequency Identification Applications," pp. 258–269, 2010.

[12] N. N. Anandakumar, T. Peyrin, and A. Poschmann, "A Very Compact FPGA Implementation of LED and PHOTON," Prog. Cryptology--INDOCRYPT 2014, pp. 304–321, 2014.

[13] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED Block Cipher," Cryptogr. Hardw. Embed. Syst. 2011, pp. 326–341, 2011.

[14] D. Augot and M. Finiasz, "Exhaustive search for small dimension recursive MDS diffusion layers for block ciphers and hash functions," IEEE Int. Symp. Inf. Theory - Proc., pp. 1551–1555, 2013.

[15] A. Bogdanov, M. Knezevic, G. Leander, and D. Toz, "SPONGENT : A Lightweight Hash Function," in Cryptographic Hardware and Embedded Systems–CHES 2011, 2011, pp. 312–325.