Low Area and Low Power Implementation for CAESAR Authenticated Ciphers

Amr Abbas[1], Hassan Mostafa[2,3], Ahmed Nader Mohieldin[2]
[1]IC Verification Solutions, Mentor Graphics, a Siemens Business, Cairo, Egypt
[2]Electronics and Communications Engineering Department, Cairo University, Giza, Egypt
[3]Nanotechnology Program at Zewail City of Science and Technology, Cairo, Egypt
{amr_abbas@mentor.com, hmostafa@uwaterloo.ca, anader2000@yahoo.com}

***Abstract*—Authenticated Encryption (AE) and Authenticated Encryption with Associated Data (AEAD) play a significant role in cryptography as they simultaneously provide confidentiality, integrity, and authenticity assurances on the data. The Competition for Authenticated Encryption, Security, Applicability, and Robustness (CAESAR) seeks optimal authenticated ciphers based on multiple criteria, including security, performance, area, and energy-efficiency. In this paper, low area and low power implementations of selected ciphers from the CAESAR candidates namely NORX, Tiaoxin, SILC, and COLM are provided. A reduction in area with an average of 43% and a reduction in dynamic power with an average of 54% are achieved compared to their corresponding high-speed architectures. Moreover, throughput (TP) in (Mbps) decreases by an average of 68% and throughput-to-area (TP/A) in (Mbps/Slices) decreases by an average of 48%.**

Keywords: Authenticated Cipher, CAESAR, FPGA, Lightweight, Power, Energy

I. Introduction

Internet of Things (IoT) devices often requires cryptographic protections due to transactions of sensitive data. The need for Authenticated encryption emerged from the observation that securely combining separate confidentiality and authentication block cipher operation modes could be error-prone and difficult. Authenticated encryption was designed as a single primitive that is easy for developers to use. It provides all the necessary cryptographic services of confidentiality, integrity, and authentication.

Authenticated encryption ciphers take a message (M), an associated data (AD), a public message number (Npub), and an optional secret message number (Nsec) as an input and generate resulting ciphertext (C) , Tag (Tag) and optional encrypted (Nsec). Integrity of data and authenticity of sender are ensured by a keyed-hash computation which occurs on all blocks of (Npub), (AD) and (M). The result of these computations is forwarded to the recipient as a Tag, as shown in Figure. 1. In authenticated decryption, the recipient receives original (AD) and (Npub), along with (C) and (Tag), and uses Key to decrypt (C) to (M). The authenticated decryption recreates a Tag (Tag'), and releases the ciphertext if and only if Tag = Tag', then authentication and integrity of the transaction are assured, otherwise the decrypted ciphertext is not released.

The CAESAR Competition for Authenticated Encryption Security, Applicability, and Robustness was announced in order to encourage the design of AE algorithms. The contest started off with 57 candidates in round 1, then only 29 candidates qualified to round 2, and finally, in round 3, 15 candidates were selected. The Cryptographic Engineering Research Group (CERG) at George Mason University (GMU), USA, runs and maintains the online platform ATHENa [13] aimed at automated evaluation of hardware cryptographic cores targeting Field Programmable Gate Arrays (FPGAs), Systems on Chip, and Application Specific Integrated Circuits (ASICs). One of their on-going projects is the comparison of FPGA implementations of the CAESAR competition candidates. They have also provided high-speed round-based implementations of round 2 and round 3 candidates. The most recent benchmarking results are published in [14], where the authors provided a summary of available implementations for round 3 candidates that are either designed by the CERG research group or other members of the cryptographic community.

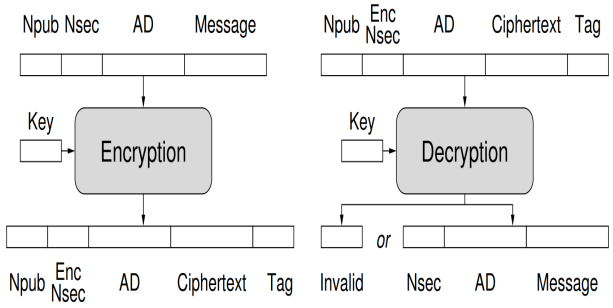

Fig. 1: Input and Output of an Authenticated Cipher [17]

Lightweight applications such as smart card, Radio Frequency IDentification (RFID), etc. demand low area and low memory footprint. AEAD schemes suitable for implementation in wearables additionally require that the power consumed is as minimum as possible. The CEASAR submitted candidates are optimized for high speed (HS), however, lightweight (LW) and energy efficient implementations are addressed in [6], [7], and [9]. In the proposed work round 3 candidates are analyzed from the point of view of their capability for low area and low power implementation through resources sharing. Optimized implementations for NORX, Tiaoxin, SILC, and COLM are provided that achieve reduced area and power consumption. Each optimized implementation and high-speed

implementation pair are benchmarked in Virtex-7 FPGA and compared in terms of area measured in Slices, throughput (TP) in (Mbps), and throughput to area (TP/A) in (Mbps/Slices). Also, the proposed work is compared to the implementations proposed in [7].

The paper is organized as follows: In Section II the previously related work is reviewed, then Section III describes the optimized implementations. Following that Section IV presents the implementations results and compares the results with high-speed implementations. Finally Section V concludes the paper and Section VI summarize the future work.

II. Related Work

Certain CAESAR candidates can be realized using low area implementations. An example in [6] where low area implementation of Ascon is presented which uses 2.57 Kilo-Gate Equivalent (KGE) in 90 nm ASIC technology, however, this version is not compliant with the CAESAR Hardware Application Programming Interface (HW API). In [10], the authors proposed low area implementation of AEGIS-128 by sharing of resources which requires 18 KGE.

There were attempts to provide dedicated lightweight authenticated encryption schemes. An example Hummingbird-2, which required 2.2 kGE in ASIC [11]. Later, AES-Based LW Authenticated Encryption was presented which require an area of 2.5 kGE and use the standard AES cryptographic primitive [12].

The majority of HW submissions of CAESAR are implemented using the CAESAR HW Development Package v1.0 [15] then a new version of the CAESAR HW Development Package v2.0 supporting lightweight (LW) implementations [16] was released. In [9] authors present LW implementations of CAESAR candidates Ketje Sr, Ascon-128, and Ascon-128a. They demonstrate that the use of a prototype version of the LW Development Package v2.0 significantly reduces the overhead of interface modules compared to the previous CAESAR HW Development Package v1.0. In [7] authors improved upon the HS implementations of ACORN, NORX, CLOC, and SILC ciphers by designing true LW implementations. Their design methodology consists of two aspects:

- Use of the LW CAESAR HW Development Package v2.0, with I/O bus widths of 8, 16, or 32 bits.
- Use of internal data paths for cryptographic primitives and authenticated cipher layer operations, which are matched to their corresponding I/O bus widths.

III. Low Power and Low Area Implementations

The optimization methodology depends on resource sharing as the addressed Ciphers (NORX, Tiaoxin, SILC, and COLM) use resource duplication in their High Speed implementations. The CAESAR HW Development Package v1.0 is used in the proposed work.

A. NORX

NORX[2] has a unique parallel architecture based on monkey duplex construction, where the degree of parallelism and tag size can be changed arbitrarily. The scheme is based on Addition-Rotation-XOR(ARX) instead of modular addition.

The pseudo code for the NORX core permutation F is given in Figure 2. A single NORX round F processes the state S by first transforming its columns with the function G using function Col(S), and then transforming its diagonals using function Diag(S).

The high-speed NORX hardware implementation duplicates the G function 8 times. The round operation is done in 2 steps, at the first step, 4 G functions operate on the columns, and at the second step, the other 4 G functions operate on the diagonals.

In order to optimize NORX for low area, only one G function is used so that the Round operation is processed in 8 cycles instead of 1 cycle. A register is added which is shifted every clock cycle from the 8 cycles to prepare the data for the G function. A counter is added to control the flow of data to and from the G function. The optimization removes 7 instances of the G function.

Algorithm: $\mathsf{F}^l(S)$

1. **for** $i \in \{0,\ldots,l-1\}$ **do**
2. $\quad S \leftarrow \mathsf{diag}(\mathsf{col}(S))$
3. **end**
4. **return** S

Algorithm: $\mathsf{G}(a,b,c,d)$

1. $a \leftarrow \mathsf{H}(a,b)$
2. $d \leftarrow (a \oplus d) \ggg r_0$
3. $c \leftarrow \mathsf{H}(c,d)$
4. $b \leftarrow (b \oplus c) \ggg r_1$
5. $a \leftarrow \mathsf{H}(a,b)$
6. $d \leftarrow (a \oplus d) \ggg r_2$
7. $c \leftarrow \mathsf{H}(c,d)$
8. $b \leftarrow (b \oplus c) \ggg r_3$
9. **return** a,b,c,d

Algorithm: $\mathsf{col}(S)$

1. $(s_0, s_4, s_8, s_{12}) \leftarrow \mathsf{G}(s_0, s_4, s_8, s_{12})$
2. $(s_1, s_5, s_9, s_{13}) \leftarrow \mathsf{G}(s_1, s_5, s_9, s_{13})$
3. $(s_2, s_6, s_{10}, s_{14}) \leftarrow \mathsf{G}(s_2, s_6, s_{10}, s_{14})$
4. $(s_3, s_7, s_{11}, s_{15}) \leftarrow \mathsf{G}(s_3, s_7, s_{11}, s_{15})$
5. **return** S

Algorithm: $\mathsf{diag}(S)$

1. $(s_0, s_5, s_{10}, s_{15}) \leftarrow \mathsf{G}(s_0, s_5, s_{10}, s_{15})$
2. $(s_1, s_6, s_{11}, s_{12}) \leftarrow \mathsf{G}(s_1, s_6, s_{11}, s_{12})$
3. $(s_2, s_7, s_8, s_{13}) \leftarrow \mathsf{G}(s_2, s_7, s_8, s_{13})$
4. $(s_3, s_4, s_9, s_{14}) \leftarrow \mathsf{G}(s_3, s_4, s_9, s_{14})$
5. **return** S

Algorithm: $\mathsf{H}(x,y)$

1. **return** $(x \oplus y) \oplus ((x \wedge y) \ll 1)$

Fig. 2: The NORX permutation function [2]

B. Tiaoxin-346

Tiaoxin 346[3] is a nonce-based authenticated encryption scheme, The internal state consists of 13 words of 16 bytes each. The 13 words are divided into three groups of 3, 4 and 6 words each. The state update function for Tiaoxin-346 absorbs a message block of 32 bytes and produces a new internal state, as illustrated in Figure 3.

One complete round of encryption uses 6 keyed AES calls. The high-speed Tiaoxin-346 hardware implementation duplicates AES 6 times. In order to optimize Tiaoxin-346 for low area, only one AES is used. The round operation is processed in 6 cycles instead of 1 cycle. Multiplexes are added to control data to the AES, latches to save data from AES and a counter to control the flow of data to and from the AES function. The optimization removes 5 instances of AES.

C. COLM

COLM[5] is a block cipher based on Encrypt-Linear mix-Encrypt mode, designed with the goal to achieve online misuse

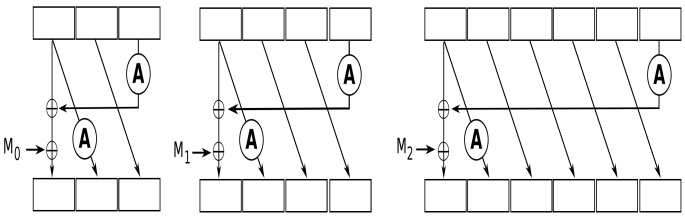

Fig. 3: The round function in Tiaoxin 346. Circled A stands for one AES round

resistance, to be fully parallelizable, and to be secure against blockwise adaptive adversaries.

The authenticated encryption for complete message block is shown in Figure 4. COLM consists of two-layer parallelizable encryption. COLM mixes the output of the first encryption layer to generate the input to the second encryption layer, using linear mixing function. The high-speed COLM implementation instantiates two instances of AES to implement the two layers of encryption. In order to optimize COLM for low area, only one instance of AES is used to perform the two encryption layers. A Finite state machine and Multiplexers are added to control the data flow to the AES. The optimized encryption operation is processed in twice the clock cycles of the non-optimized one and the same applies for the decryption operation.

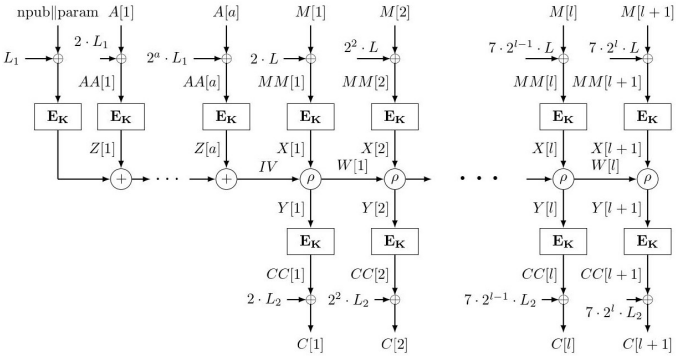

Fig. 4: COLM authenticated encryption for complete message block. E_{K}. denotes the block cipher AES-128 [5]

D. SILC

SILC [4] is a block cipher mode of operation for authenticated encryption. Its design goal is to optimize the HW implementation cost of CLOC [4]. In other words, SILC is a lighter version of CLOC. CLOC aims at being provably secure and optimizing the implementation overhead beyond the block cipher, the precomputation complexity, and the memory requirement. SILC also maintains the provable security based on the pseudo-randomness of the underlying block cipher. SILC is suitable for use within constrained hardware devices. SILC can be implemented based on the AES-128 block cipher for a 16-byte block length.

The four functions used in SILC are shown in Figure 5, HASH, ENC, DEC, and HASH, are all sequential. However, the block cipher calls in ENC and PRF can be done in parallel. The high-speed SILC hardware implementation does the block cipher calls for ENC and PRF in parallel. In order to optimize SILC for low area the block cipher calls in ENC and PRF are done sequentially. One AES round is used instead of two AES rounds, and as a result, one round operation is done in 2 cycles instead of 1 cycle.

Algorithm SILC-$\mathcal{E}_K(N,A,M)$	Algorithm SILC-$\mathcal{D}_K(N,A,C,T)$
1. $V \leftarrow \mathsf{HASH}_K(N,A)$	1. $V \leftarrow \mathsf{HASH}_K(N,A)$
2. $C \leftarrow \mathsf{ENC}_K(V,M)$	2. $T^* \leftarrow \mathsf{PRF}_K(V,C)$
3. $T \leftarrow \mathsf{PRF}_K(V,C)$	3. if $T \neq T^*$ then return $\perp$
4. return (C,T)	4. $M \leftarrow \mathsf{DEC}_K(V,C)$
	5. return M

Fig. 5: the encryption and the decryption algorithms of SILC [4]

IV. Results

To evaluate the hardware performance of the proposed optimized implementations, pairs of corresponding publicly-available HS implementations [8] (donated by High-Speed Implementations) and proposed Optimized implementations (denoted by Optimized Implementations) are benchmarked in the Virtex-7 FPGA (xc7vx485tffg1157-1). Results are shown in Table I. The results show that the proposed optimized implementations achieve an area reduction for NORX, Tiaoxin, SILC and COLM with 65%, 40%, 35% and 33% respectively, and a Dynamic Power consumption reduction by 88%, 66%, 22% and 39% respectively. As a cost, throughput (TP) decreases for NORX, Tiaoxin, SILC and COLM by 87.5%, 83%, 50% and 50% respectively, and throughput-to-area (TP/A) decreases by 64%, 72%, 30% and 25% respectively. The reduction in TP and TP/A ratio is expected as latency and throughput are sacrificed for area reduction.

For NORX and SILC the proposed optimized implementations are compared to work proposed in [7]. In [7] virtex-6 FPGA is used for implementation, while virtex-7 FPGA is used in this research so a comparison is done between Area reduction, Dynamic Power reduction and throughput-to-area change achieved by proposed work and the work in [7]. The comparison is summarized in table 2. For NORX proposed implementation has higher area reduction with 65% compared to 53.4% in [7] while for proposed implementation the throughput-to-area (TP/A) has decreased with 64% while it increased with 25.5 % in [7]. For SILC proposed implementation has lower area reduction with 35% compared to 69 % in [7] while proposed implementation has less reduction in throughput-to-area (TP/A) with 30% compared to 65% in [7].

V. CONCLUSIONS

In this paper, low area and low power implementations for four candidates (NORX, Tiaoxin-346, SILC, COLM) of

TABLE I: Results of Implementations of Ciphers in Virtex-7 FPGA

Algorithms	Area [Slices]	Reduction [%]	Dynamic Power [mW]	Reduction [%]	Freq [MHz]	TP [Gb/Sec]	Reduction [%]	TP/Area [Mbps /Slices]	Reduction [%]
High-speed Implementations									
NORX	928	-	447	-	250	24	-	25.86	-
Tiaoxin	1649	-	342	-	434	111	-	67.3	-
SILC	623	-	108	-	285	3.6	-	6.26	-
COLM	1566	-	173	-	250	2.9	-	1.85	-
Optimized Implementations									
NORX	326	65	53	88	250	3	87.5	9.2	64
Tiaoxin	994	40	116	66	434	18.5	83	18.6	72
SILC	410	35	84	22	285	1.8	50	4.39	30
COLM	1054	33	106	39	250	1.45	50	1.38	25

TABLE II: Comparison of Results to Work proposed in [7]

Algorithms	Area Reduction [%]	Dynamic Power Reduction [%]	TP/Area Change [%]
Work Proposed in [7]			
NORX	53.3	82	+25.5
SILC	69.1	29	-65
Optimized Implementation			
NORX	65	88	-64
SILC	35	22	-30

CAESAR Round 3 are proposed. The optimized implementations and the corresponding high-speed implementations are benchmarked in the Virtex-7 FPGA. A reduction in area with an average of 43% and a reduction in dynamic power with an average of 54% are achieved compared to their corresponding high-speed architectures. As a cost, throughput (TP) decreases by an average of 68% and throughput-to-area (TP/A) decreases by an average of 48%.

VI. AREAS FOR FUTURE RESEARCH

Future research could include additional ciphers of the CEASAR candidates. It could also include combining the use of reduced internal data path widths, and the LW CAESAR Development Package (work proposed in [7]) along with the work proposed in this research to achieve more area reduction.

VII. ACKNOWLEDGMENT

This work was partially funded by Mentor Graphics, and ONE Lab at Cairo University and Zewail City of Science and Technology.

REFERENCES

[1] "CAESAR competition for authenticated encryption," 2012, http://competitions.cr.yp.to/caesar.html.

[2] J. Aumasson, P. Jovanovic, and S. Neves, "NORX," [Online]. Available: https://competitions.cr.yp.to/round3/norxv30.pdf.

[3] I. Nikolic, "TIAOXIN," [Online]. Available: https://competitions.cr.yp.to/round1/tiaoxinv1.pdf.

[4] T. Iwata, K. Minematsu, J. Guo, S. Morioka, and E. Kobayashi, "CLOC and SILC," [Online]. Available: https://competitions.cr.yp.to/round3/clocsilcv3.pdf.

[5] E. Andreevam, A. Bogdanov, N. Datta, A. Luykx, B. Mennink, M. Nandi, and E. Tischhauser, "CLOM," [Online]. Available: https://competitions.cr.yp.to/round2/colm.pdf.

[6] H. Gro, E. Wenger, C. Dobraunig, and C. Ehrenhofer, " Suit up! Made-to-Measure Hardware Implementations of ASCON," 2015 Euromicro Conference on Digital System Design, Funchal, Portugal, Aug. 2015.

[7] F. Farahmand, W. Diehl, A. Abdulgadir, J. Kaps and K. Gaj, "Improved Lightweight Implementations of CAESAR Authenticated Ciphers," Proceedings of the 26th IEEE International Symposium on Field-Programmable Custom Computing Machines,FCCM 2018, Boulder, CO, USA, Jun 2018.

[8] CERG, "Hardware Benchmarking of CAESAR Candidates," Aug 2017, [Online]. Available: https://cryptography.gmu.edu/athena/index.php?id=CAESAR.

[9] P. Yalla and J. P. Kaps, "Evaluation of the CAESAR Hardware API for Lightweight Implementations," 2017 International Conference on ReConFigurable Computing and FPGAs, ReConFig 2017, Cancun, Mexico, Dec. 4-6, 2017.

[10] D. Bhattacharjee and A. Chattopadhyay, "Efficient Hardware Accelerator for AEGIS-128 Authenticated Encryption," 10th International Conference, Inscrypt 2014, Beijing, China, December 13-15, 2014.

[11] D. Engels, O. Saarinen, P. Schweitzer and E. Smith, "The Hummingbird-2 Lightweight Authenticated Encryption Algorithm," RFID. Security and Privacy: 7th International Workshop, RFIDSec '11, Amherst, Massachusetts, USA,June 2011.

[12] A. Bogdanov, F. Mendel, F. Regazzoni, V. Rijmen, and E. Tischhauser, "ALE: AES-Based Lightweight Authenticated Encryption," 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013

[13] Automated Tool for Hardware Evaluation (ATHENa). [Online]. Available: https://cryptography.gmu.edu/athena

[14] Homsirikamol, E. Farahmand, F. Diehl, and W. Gaj, "Benchmarking of Round 3 CAESAR Candidates in Hardware: Methodology, Designs and Results," [Online]. Available: https://cryptography.gmu.edu/athena/presentations/CAESAR_R3_HW_Benchmarking.pdf.

[15] E. Homsirikamol, W. Diehl, A. Ferozpuri, F. Farahmand, and K. Gaj, "Implementers Guide to Hardware Implementations Compliant with the CAESAR Hardware API, v1.0," [Online]. Available: https://cryptography.gmu.edu/athena/CAESAR_HW_API/CAESAR_HW_Implementers_Guide_v1.0.pdf.

[16] E. Homsirikamol, W. Diehl, A. Ferozpuri, F. Farahmand, and K. Gaj, "Implementers Guide to Hardware Implementations Compliant with the CAESAR Hardware API, v2.0," [Online]. Available: https://cryptography.gmu.edu/athena/CAESAR_HW_API/CAESAR_HW_Implementers_Guide_v2.0.pdf.

[17] E. Homsirikamol, W. Diehl, A. Ferozpuri, F. Farahmand, M.U. Sharif, and K. Gaj, "A Universal Hardware API for Authenticated Ciphers," 2015 International Conference on Reconfigurable Computing and FPGAs, ReConFig 2015, Mayan Riviera, Mexico, Dec. 7-9, 2015

[18] K. Khateb, M. Ahmed, A. K. ELdin, M. AbdelGhany, and H. Mostafa, "Dynamically Recongurable Power Efcient Security for Internet of Things Devices", IEEE International Conference on Modern Circuits and Systems Technologies (MOCAST2018), Thessaloniki, Greece, pp. 1-4, 2018.

[19] M. Bahnasawi, A., K. Ibrahim, A. Mohamed, M. Khalifa, A. Moustafa, K. Abelmonim, Y. ismail, and H. Mostafa, "ASIC-Oriented Comparative Review of Hardware Security Algorithms for the Internet of Things Applications", IEEE International Conference on Microelectronics (ICM 2016), Cairo, Egypt, pp. 285-288, 2016.