Efficient Hardware Accelerator for NORX Authenticated Encryption

Sachin Kumar, Jawad Haj-Yahya and Anupam Chattopadhyay
School of Computer Science and Engineering, Nanyang Technology University, Singapore

***Abstract*—Authenticated encryption with associated data (AEAD) plays a significant role in cryptography due to its ability to provide integrity, confidentiality and authenticity at the same time. There is an unceasing demand of high-performance and area-efficient AEAD ciphers due to the emergence of security at the edge of computing fabric, such as, sensors and smartphone devices. Currently, a worldwide contest, titled CAESAR, is being held to decide on a set of AEAD ciphers, which are distinguished by their security, runtime performance, energy-efficiency and low area budget. In this paper, we focus on optimizing the hardware architecture of NORX by applying a pipeline technique. Our pre-layout results using commercial ASIC TSMC 65 technology library show that optimized NORX is 40.81% faster, 18.01% smaller, and improved the throughput per area by 76.9% when compared with state-of-the-art NORX implementation.**

I. Introduction

Authenticated-Encryption with Associated-Data (AEAD) scheme protects both privacy and authenticity of the message when it is transformed into ciphertext where there may be additional information, such as a packet header, that travels alongside the ciphertext and must get authenticated with it. The need for AEAD emerged from the observation that securely combining a confidentiality mode with an authentication mode could be error prone and difficult [1], [2]. Number of practical attacks introduced into production protocols and applications by incorrect implementation, or lack, of authentication (including SSL/TLS)[3]. Moreover, developing two separate algorithms for encryption and authentication can increase implementation complexity of the cipher in software as well as in hardware.

An AEAD scheme typically consists of two tasks. The first one is encryption $\mathcal{E}_K(AD, M)$ which takes as input a shared key K, public associated data AD and the message to be encrypted M and returns a tagged ciphertext C. The second one is decryption/verification $\mathcal{D}_K(AD, C)$, which either returns an invalid symbol $\perp$ if the received ciphertext, associated data and the authentication data do not match, or the decrypted message M, otherwise. There are three main typical approaches which are adopted for AEAD: Encrypt-then-MAC (EtM), Encrypt-and-MAC (E&M) and MAC-then-Encrypt (MtE) as shown in Fig. 1. In the first approach, the plaintext is first encrypted and message authentication code (MAC) is computed based on the resulting ciphertext. This approach can be used to reach highest definition of security in authenticated encryption(AE) provided that MAC should be strictly unforgeable while in the second approach, MAC is produced with the help of plain text and the plaintext is encrypted without the MAC. In the MAC-then-Encrypt (MtE) approach, a MAC is computed from the plaintext. The MAC and plaintext both are encrypted to produce ciphertext. The ciphertext along with its encrypted MAC are sent together. Its mostly used in SSL/TLS [4].

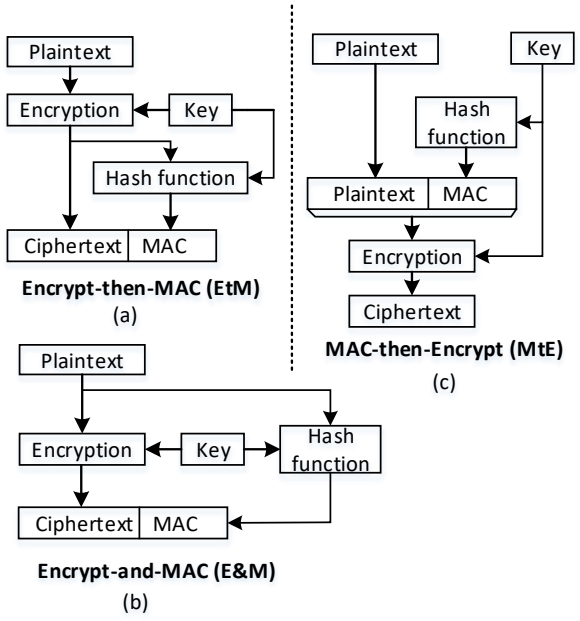

Fig. 1: Approaches to Authenticated Encryption with Associated Data

The most widely used authenticated scheme is AES-GCM [5] which is based on the Advanced Encryption Standard (AES). Even most efficient hardware implementation of AES-GCM may not be suitable in modern world of computing and data movement due to the bottleneck caused by slow multiplication of Galois field. The security and performance requirement in many modern applications such as smart devices, sensors etc., are difficult to accomplish by the current AE standard. Therefore, there is a need to have faster, smaller and lesser power-hungry AE cipher. With the great interest and importance of AE, a worldwide contest, named as CAESAR [1], competition was announced in 2014 [6]. This competition mainly signifies the current need for practical, secure and

[1]CAESAR stands for "Competition for Authenticated Encryption: Security, Applicability, and Robustness".

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

efficient AEAD schemes which also aims for finding authenticated encryption schemes that can offer better performance over AES-GCM. In the survey presented in [7], the round two candidates of the CAESAR competition were categorized into five families on the basis of their base constructions: block cipher-based, stream cipher-based, key-less permutations, hash-function-based and dedicated schemes. AEAD ciphers based on block-cipher allows block-level parallelism while using the underlying block cipher, such as the Offset Code Book mode OCB [8], [9], [10], the Synthetic Counter-in-Tweak mode (SCT) [11] and the Offset Two-Round mode (OTR) [12]. In September 2016, the CAESAR competition committee announced the selection of 15 AEAD schemes as candidates for round 3 of the CAESAR competition [6] wherein some AE algorithms are based on AES primitives such as AES-OTR, SILC, CLOC etc. while some of them are based on sponge based construction such as ASCON, NORX etc.

The experimental study for the evaluation of their hardware performance of AEAD schemes selected in the third round of CAESAR competition is also equally important. Basic implementation of all candidates are done on various platforms with different interfaces. Furthermore, performance of the several designs is better on some platforms, e.g., Field Programmable Gate Array (FPGA). However, FPGA boards are mainly used for verification of the design with the help of programmable gates but it does not provide actual performance metrics of the design which can only be achieved by implementing the design in Application-Specific Integrated Circuits (ASIC). In addition, all the available hardware implementations of the CAESAR competition candidates on the ATHENa hardware evaluation website [13] are fully sequential implementations, i.e. to start processing a new block, all the previous blocks have to be finished. These implementations do not take full advantage of the specific characteristics of the schemes based on the aforementioned modes.

The sponge-based cryptographic primitives, due to their relatively hardware-friendly design and proven security results, are gaining popularity. A prominent example of this is Secure Hash Algorithm (SHA-3), which is standardized by NIST. In this paper we focus on achieving efficient hardware implementation of NORX, a sponge-based AE scheme in the third round of CAESAR contest. The underlying primitives of NORX is the function G, which requires only bitwise logical AND, XOR, and shift operations, making it a strong contender as CAESAR winning candidate. However, no efficient hardware architecture for NORX has been proposed so far.

II. Authenticated Encryption scheme based on sponge construction

The sponge function and duplex construction are being widely used to develop many cryptographic algorithms including authenticated encryption schemes. The duplex construction is used to develop AE algorithms submitted to CAESAR such as ASCON, NORX, Ketje etc. [6]. In duplex construction function, the fixed permutation F are determined by the following two parameters: bitrate r and capacity c [14] [15]. Input/output length or the state size (S) are computed by adding the parameters bitrate r and capacity c (i.e. $S = r + c$). For a fixed state size (S), there is always a trade-off between security and speed because of giving different values for bit-rate and capacity. For instance; higher bitrate makes the algorithm faster with lower security and vice versa. In the duplex construction, input blocks (plaintext) are given into the bitrate part of the permutation F. If an input block is smaller than r bits, it gets padded to achieve the full r bits. After processing by the permutation F, r-bit output blocks (ciphertext) are squeezed out. In the process, the capacity (c) part of the state size (S) is never directly manipulated, neither for absorbing nor for squeezing.

Authentication encryption with associated data (AEAD) can be developed by using the duplex construction as depicted in the Fig. 2. First, the concatenation of a secret key K and a nonce are given into the initial state. The permutation F is applied on the initial state. The following steps show the duplex construction operation on n blocks of plaintext P_i, computing the ciphertext C_i correspondingly, where $i = 0, \cdots, n$. For the plaintext P_i, the ciphertext C_i is computed by XOR-ing the plaintext with the bitrate part squeezed out from the state. The permutation F is kept calling until the last block of plaintext. Besides plaintext blocks, public associated data blocks can be processed similarly by absorbing them without encryption. The encryption is completed by getting a r-bit authentication tag T. It can be noticed that the sponge-based authenticated encryption schemes is developed by using duplex construction but their security largely depends on the internal structure of permutation F which is usually implemented as a sequence of elementary operations called as rounds. It can be assumed that permutation F with more rounds makes cryptanalysis more complex and the relevant AEAD becomes more secure but with the more hardware cost.

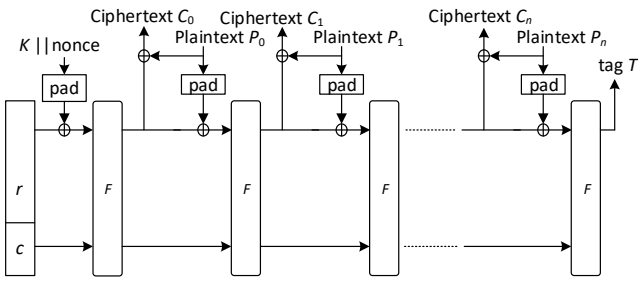

Fig. 2: Duplex construction based Authenticated encryption scheme.

III. NORX cipher and its architecture

NORX has an unique parallel architecture based on monkey duplex construction [15] along with its two versions i.e. either NORX-32 or NORX-64. This AEAD scheme comes with an associated data supporting arbitrary parallelism. In addition,

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

the NORX algorithm is developed with the help of Addition-Rotation-XOR(ARX) primitives instead of modular addition. This cipher was optimized to be efficient in both software as well as hardware with a single-instruction multiple-data (SIMD)-friendly core and no secret-dependent memory lock-ups. The underlying permutation F is designed by referencing ChaCha stream cipher where in the integer addition is replaced by a simple bit-wise XOR operation i.e. $(a \oplus b) \oplus (a \wedge b) \ll 1$ which leads to improve its hardware efficiency.

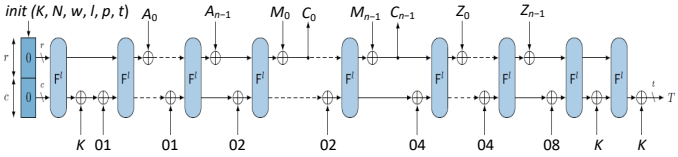

Fig. 3: Authenticated Encryption with associated data procedure of NORX [6].

In the Fig. 3, the core algorithm F of NORX is a permutation of $S = r + c$ bits, where S is called the width or state size, r the rate (or block length), and c the capacity. F is called a round and F^{l} denotes its $l-$fold iteration. The state is viewed as a concatenation of 16 words, i.e. $S = s_0||...||s_{15}$, out of which $s_0, ..., s_{11}$ are called the rate words where data blocks are injected and $s_{12}, .., s_{15}$ are called the capacity words which remain untouched. Conceptually, the 16 state words are arranged in a 4×4 matrix:

$$S = \begin{bmatrix} s_0 & s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 & s_7 \\ s_8 & s_9 & s_{10} & s_{11} \\ s_{12} & s_{13} & s_{14} & s_{15} \end{bmatrix}$$

The pseudocode for the NORX core permutation F^{l} is given in Algorithm 1. 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 $Diag(S)$ as depicted in Fig. 4 (A). The G function uses cyclic rotations and non-linear operation interchangeably to update its four input words (due to space-constraint, function G is not included).

IV. Optimized hardware architecture of NORX

The state-of-the-art NORX hardware implementation which is reported in [6] has duplication of resources, the operations $S \leftarrow Col(S)$ and $S \leftarrow diag(S)$ at the F^{l} function that is shown in Algorithm 1 duplicate the G function eight times, the operations are done in two phases, at the first phase four G functions operates on the four columns of the matrix, once the result is ready then the second phase starts, where additional four G functions operates on the four diagonals of the matrix as shown in Fig. 4 (A).

In order to optimize NORX, the original algorithm of $F^{\mathrm{l}}(S)$ has been modified as presented in Algorithm 2, where this algorithm is sharing the same resources of the four G functions to apply them on both columns and diagonals using the

Algorithm 1 Algorithm of given NORX (not including implementation of G).

function $F^{\mathrm{l}}(S)$
  for $i = 1$ to l **do**
    $S \leftarrow Col(S)$
    $S \leftarrow Diag(S)$
    return S
  end for
  return S
end function
function Col(S)
  $(s_0, s_4, s_8, s_{12}) \leftarrow G(s_0, s_4, s_8, s_{12})$
  $(s_1, s_5, s_9, s_{13}) \leftarrow G(s_1, s_5, s_9, s_{13})$
  $(s_2, s_6, s_{10}, s_{14}) \leftarrow G(s_2, s_6, s_{10}, s_{14})$
  $(s_3, s_7, s_{11}, s_{15}) \leftarrow G(s_3, s_7, s_{11}, s_{15})$
  return S
end function
function Diag(S)
  $(s_0, s_5, s_{10}, s_{15}) \leftarrow G(s_0, s_5, s_{10}, s_{15})$
  $(s_1, s_6, s_{11}, s_{12}) \leftarrow G(s_1, s_6, s_{11}, s_{12})$
  $(s_2, s_7, s_8, s_{13}) \leftarrow G(s_2, s_7, s_8, s_{13})$
  $(s_3, s_4, s_9, s_{14}) \leftarrow G(s_3, s_4, s_9, s_{14})$
  return S
end function

Algorithm 2 Algorithm of the optimized NORX(not including implementation of G)

function $F^{\mathrm{l}}(S)$
  T ▷ Temporary matrix T, includes transformation of S
  for $i = 1$ to l **do**
    $S \leftarrow ColDiag(S)$
    $T[0, 4, 8, 12] \leftarrow S[0, 5, 10, 15]$
    $T[1, 5, 9, 13] \leftarrow S[1, 6, 11, 12]$
    $T[2, 6, 10, 14] \leftarrow S[2, 7, 8, 13]$
    $T[3, 7, 11, 15] \leftarrow S[3, 4, 9, 14]$
    $T \leftarrow ColDiag(T)$
    $S[0, 5, 10, 15] \leftarrow T[0, 4, 8, 12]$
    $S[1, 6, 11, 12] \leftarrow T[1, 5, 9, 13]$
    $S[2, 7, 8, 13] \leftarrow T[2, 6, 10, 14]$
    $S[3, 4, 9, 14] \leftarrow T[3, 7, 11, 15]$
  end for
  return S
end function
function ColDiag(S)
  $(s_0, s_4, s_8, s_{12}) \leftarrow G(s_0, s_4, s_8, s_{12})$
  $(s_1, s_5, s_9, s_{13}) \leftarrow G(s_1, s_5, s_9, s_{13})$
  $(s_2, s_6, s_{10}, s_{14}) \leftarrow G(s_2, s_6, s_{10}, s_{14})$
  $(s_3, s_7, s_{11}, s_{15}) \leftarrow G(s_3, s_7, s_{11}, s_{15})$
  return S
end function

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

functions $ColDiag(S)$ that is called twice, at the second call we transform the state matrix S using the temporary state matrix. The proposed hardware architecture based on Algorithm 2 is shown in Fig. 4 (B). The optimization removes *four* instances of the G function by converting the implementation in sequential, a buffer was added to collect the intermediate results of the status matrix. A small logic was added that rearrange the status matrix that is received from the first stage of calculation (applied on column). It also prepares the data in correct order for the diagonals calculation stage. Moreover, a Finite State Machine (FSM) logic is added in order to control the flow. In order to account for the additional delay due to the insertion of the pipe stage, few counters that controls the main pipeline of the NORX were modified.

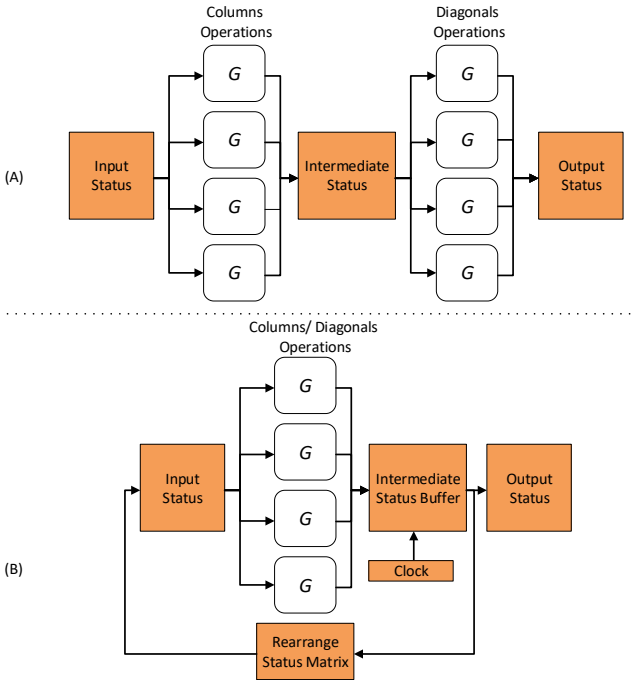

Fig. 4: NORX hardware architecture (A) Before optimization (B) After the proposed optimization.

V. Experimental Results and Comparison

To evaluate the hardware performance of proposed hardware architecture of NORX cipher (denoted by Optimized NORX), the Optimized NORX is compared against the given NORX architecture [6]. Both designs are described in VHDL language and functionally verified by given test vectors using pre-layout simulation tool. In what follows, the designs are synthesized and mapped to TSMC 65nm standard cell library by using Synopsys Design Compiler version J-2014.09. There are two cases. In the first case, the designs are optimized for minimum achievable delay. In the second case, the designs are constrained for achieving minimum hardware area at minimum frequency. The areas are reported in μm^2 as well as in kilo gate equivalent (KGE) which is computed by dividing the measured area by the area of two input NAND gate (for this library, area of nand gate is 1.44 μm^2). The synthesis results for both cases are shown in Table I and Table II.

TABLE I: Synthesis results of Optimized NORX and NORX at minimum delay.

Metrics	NORX	Optimized NORX	Improvement
Delay (ns)	2.23	1.32	40.8%
Frequency (MHZ)	448.43	757.57	68.9%
Throughput (Gbps)	57.40	83.11	44.8%
Area (μm^2)	123180.48	100988.64	18.01%
KGE	85.54	70.13	18.01%
Throughput/Area (Gbps/KGE)	0.67	1.185	76.9%

Table I shows that Optimized NORX is 40.8% faster and achieved 44.8% gain in throughput and saves the hardware area by 18.01% when compared with given NORX. The elimination of four G function significantly improved the efficiency (throughput per area metric) of Optimized NORX by 76.9% over the existing NORX design.

TABLE II: Synthesis area of Optimized NORX and NORX at 448.43 Mhz.

AEAD Ciphers	Throughput (Gbps)	Area (μm^2)	KGE	Throughput /Area
NORX	57.40	121675.68	84.50	0.67
Optimized NORX	49.20	77057.28	53.51	0.92
Improvement		36.7%	36.7%	37.3%

Since there is always a trade-off between area and delay, area can be further reduced by performing the simulation at $448.43Mhz$ (slowest between both designs). In Table II, the experimental result shows that although the throughput is lower for the Optimized NORX, the area is reduced by 36.7% and as well as the efficiency of the optimized NORX (throughput per area) improved by 37.3% when compared with existing NORX architecture. Moreover, when compared with AES-based AEAD ciphers of CAESAR as shown in Table III, the synthesis results shows that Optimized NORX is a promising candidate to win CAESAR competition [6] due to its excellent throughput and area-efficiency results.

TABLE III: ASIC synthesis results comparison of Optimized NORX with AES-based ciphers from CAESAR competition.

AEAD ciphers	Frequency (Mhz)	Throughput (Gbps)	Efficiency (Gbps/KGE)
AEGIS_GMU	550.54	8.65	0.07
AEZ_GMU	581.40	2.98	0.04
CLOC_GMU	588.24	6.84	0.07
JAMBU_AES	495.05	3.17	0.12
OCB_GMU	460.83	4.92	0.05
SILC_GMU	500.00	6.40	0.14
COLM	505.05	5.88	0.05
Deoxys	549.45	2.43	0.05
Optimized NORX	**757.57**	**83.11**	**1.185**

VI. Conclusion

In this paper, a novel low area and high-speed hardware architecture for NORX cipher is proposed. The efficiency of the optimized NORX is improved by eliminating four G functions with the help of pipeline technique. Based on the pre-layout synthesis results, we show that the optimized

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

NORX significantly improves on the basic implementation available in literature, and outperforms all other CAESAR candidates in terms of throughput and throughput-per-area metrics.

ACKNOWLEDGEMENT

This research project is funded by the National Research Foundation Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

REFERENCES

[1] M. Bellare, P. Rogaway, and D. Wagner, "A conventional authenticated-encryption mode," *manuscript, April*, 2003.

[2] T. Kohno, J. Viega, and D. Whiting, "The cwc authenticated encryption (associated data) mode," *ePrint Archives*, 2003.

[3] D. J. Bernstein, "Failures of secret-key cryptography," *Invited talk at FSE at 20th International Workshop on Fast Software Encryption*, Singapore, 2013.

[4] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," *Journal of Cryptology*, vol. 21, no. 4, pp. 469–491, Oct 2008.

[5] M. J. Dworkin, "Sp 800-38d. recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac," Gaithersburg, MD, United States, Tech. Rep., 2007.

[6] CAESAR Competition, "CAESAR submissions," https://competitions.cr.yp.to/caesar-submissions.html, 2016.

[7] F. Abed, C. Forler, and S. Lucks, "General classification of the authenticated encryption schemes for the CAESAR competition," *Computer Science Review*, 2016.

[8] P. Rogaway, M. Bellare, and J. Black, "OCB: A block-cipher mode of operation for efficient authenticated encryption," *ACM Transactions on Information and System Security (TISSEC)*, vol. 6, no. 3, pp. 365–403, 2003.

[9] P. Rogaway, "Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC," in *International Conference on the Theory and Application of Cryptology and Information Security*, Jeju Island, Korea, 2004, pp. 16–31.

[10] T. Krovetz and P. Rogaway, "The software performance of authenticated-encryption modes," in *International Workshop on Fast Software Encryption*, Lyngby, Denmark, 2011, pp. 306–327.

[11] T. Peyrin and Y. Seurin, "Counter-in-tweak: authenticated encryption modes for tweakable block ciphers," in *Annual International Cryptology Conference*, CA, USA, 2016.

[12] K. Minematsu, "Parallelizable rate-1 authenticated encryption from pseudorandom functions," in *EUROCRYPT 2014*, Copenhagen, Denmark, 2014, pp. 275–292.

[13] George Mason University, "ATHENa: Automated Tools for Hardware EvaluatioN," https://cryptography.gmu.edu/athena/, 2017.

[14] A. Dwiedi, M. Klouček, P. Morawiecki, I. Nikolić, J. Pieprzyk, and S. Wójtowicz, "Sat-based cryptanalysis of authenticated ciphers from the caesar competition," in *14th International Joint Conference on e-Business and Telecommunications (ICETE 2017)*, Madrid, Spain, 2017, pp. 237–246.

[15] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, *Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications*. Springer Berlin Heidelberg, 2012, pp. 320–337.