Unified Hardware for High-Throughput AES-Based Authenticated Encryptions

Shotaro Sawataishi, Rei Ueno, *Member, IEEE*, and Naofumi Homma, *Senior Member, IEEE*

***Abstract*—This brief presents an efficient unified hardware for up-to-date authenticated encryptions with associated data (AEADs). Although some major AEADs share several fundamental components (e.g., advanced encryption standard (AES), block chaining, and XOR-Encryption-XOR (XEX) scheme), each AEAD is equipped with a unique mode of operation and/or subfunctions, which makes it difficult to integrate various AEADs in a hardware efficiently. The proposed hardware in this brief efficiently unifies the fundamental components to perform a set of AEADs with minimal area and power overheads. The proposed configurable datapath is adapted to a set of peripheral operations (e.g., block chaining and XEX), dictated by the given AEAD algorithm. In this brief, we also demonstrate the validity of the proposed hardware through an experimental design adapted to four AES-based AEADs. Consequently, we confirm that the proposed hardware can perform the four AEADs with quite smaller area than the sum of the each dedicated AEAD hardware, comparable throughput and power consumption. In addition, we confirmed that the proposed hardware is superior to software implementation on general-purpose processor in terms of both throughput and power consumption.**

***Index Terms*—Authenticated encryption, AES, cryptographic hardware implementation.**

I. Introduction

In recent years, there has been an increasing demand for AEADs, to ensure both the confidentiality and authenticity of messages. Many AEADs have been developed specifically to achieve certain performance goals, target device suitability, and security levels in international competitions for AEADs, such as CAESAR and NIST LWC [1], [2]. Thus, various AEADs can be employed not only in the Internet security applications such as TLS (Transport Layer Security) and PGP (Pretty Good Privacy), but also for Internet of Things (IoT) applications in the present and future.

In practice, the hardware implementations of AEADs are sometimes mandatory for many high-end transaction servers and resource-constrained embedded devices, to facilitate a higher throughput and/or lower area/power/energy footprint. On the other hand, some devices and fog servers also require interoperability for processing AEADs, as they can be connected to multiple networks, which means that the device should be able to efficiently perform a set of different AEAD procedures. However, each AEAD has its own mode of operation and a subcomponent set-up specific to itself. Therefore, it has been difficult to integrate and perform the peripheral operations of many AEADs around the underlying pseudo-random function (e.g., AES) using conventional datapaths.

To address the problem, the present paper suggests and designs a high throughput AEAD hardware architecture, capable of carrying out various AEAD procedures in a unified manner, whilst using AES as a cryptographic primitive. The proposed hardware employs a newly-designed reconfigurable core, referred to as the peripheral-datapath-reconfigurable (PDR)-AES core. The basic task of the PDR-AES core is to realize a reconfigurable datapath, that can perform, using some essential components, the common modes/subfunctions utilized in major AEADs. The proposed datapath is composed of an AES encryption/decryption core, an up-counter, configurable connectors for an XOR-Encryption-XOR (XEX) scheme, feedback/feedforward mechanisms for the ciphertext/message blocks, and a unified Galois field (GF) accumulator, which covers a set of AES-based AEADs, including the de facto standard AEAD (e.g., AES-GCM), as well as CAESAR and NIST LWC candidates. In particular, we design a novel high-throughput area/power-efficient pipelined AES encryption/decryption core, based on the unified hardware proposed in a recent work [3].

In this brief, the proposed hardware is validated through an experimental design which demonstrates four major types of AES-based AEADs on ASIC, whereas the proposed architecture potentially supports other types of AES-based AEADs. We compare the performance of the proposed hardware with the conventional dedicated hardware of each of the AEADs. As a result of our experiments, we can confirm that the proposed hardware performs the four AEADs with an equivalent throughput and quite smaller area than the sum of the dedicated hardware for each of the AEADs.

II. Brief Description of AEAD

The security goal of an AEAD is to ensure the confidentiality of plaintext and the authenticity of both plaintext (ciphertext) and associated data. The sender generates a ciphertext from the plaintext using a secret key, and further generates an authentication tag from the associated data and ciphertext (or plaintext). The associated data, tag, and ciphertext are then

Manuscript received March 1, 2020; revised July 6, 2020; accepted July 26, 2020. Date of publication July 31, 2020; date of current version September 3, 2020. This work was supported by JSPS KAKENHI under Grant 17H00729, Grant 19H21526, and Grant 20K19765. This brief was recommended by Associate Editor N. Maghari. *(Corresponding author: Rei Ueno.)*

The authors are with the Research Institute of Electrical Communication, Tohoku University, Sendai 980-8577, Japan (e-mail: shotaro@riec.tohoku.ac.jp; ueno@riec.tohoku.ac.jp; homma@riec.tohoku.ac.jp).

Digital Object Identifier 10.1109/TCSII.2020.3013415

sent to the receiver. The receiver obtains the plaintext from ciphertext using the secret key and also regenerates the authentication tag using the same procedure as the sender. Then, the receiver can detect any tampering of the associated data and ciphertext, by comparing the received and regenerated tags.

Currently, AES-GCM [4] is one of the most widely used encryption methods and is the de facto standard AEAD. AES-GCM is constructed on the basis of a generic composition of AES in the counter mode and a Wegman-Carter message authentication code (MAC) scheme with a polynomial hash function, referred to as GHASH. One major feature of AES-GCM is that it can be implemented with a high throughput and efficiency, especially when provided with effective hardware support.

Following the publication and deployment of AES-GCM, many AEAD algorithms have been put forwards and developed over the years, aiming to improve the efficiency of the process and to pursue different performance goals and/or preferable features to AEAD (e.g., reduction of the number of AES encryption calls required, inverse-freeness, nonce-misuse-resistance, small state-size and parallelizability). The AES with offset codebook mode (AES-OCB) [5] is given as the first rate-1 AEAD which required the smallest number of AES calls, compared to the existing (non-rate-1) AEADs. In addition, AES-OCB exploits a block-wise parallelism (e.g., multi-core and pipelining), through which it can achieve a higher throughput. For processing AEADs in AES-OCB, the parallelizable MAC using an AES (AES-PMAC)-like scheme is employed. The scheme is efficiently implemented using only mask-value addition, AES encryption, and the accumulation of AES outputs, whilst preserving the parallelizability.

The designs of many subsequent AEADs have been inspired by AES-GCM and/or AES-OCB. For example, AES with an offset two-round mode (AES-OTR) [6] addresses one of major drawbacks of AES-OCB. While AES-OCB decryption requires the inverse function of the AES (i.e., AES decryption), AES-OTR achieves inverse-freeness in addition to rate-1 and parallelizability, by using a two-round Feistel network with mask-value addition. Many other AEADs such as Pyjamask [7] and SKINNY-AEAD [8] follow the construction of the OCB mode of operation, due to its high performance and desirable features. Thus, many AEADs, including the aforementioned ones, as well as candidates of the CAESAR and NIST LWC competitions, may sometimes show similarities.

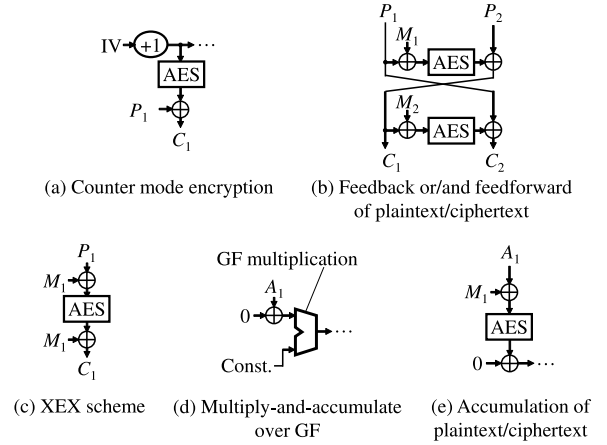

Fig. 1. Fundamental components of AES-based AEAD.

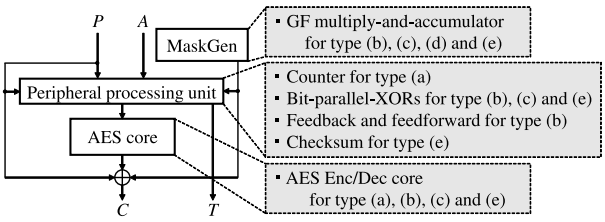

Fig. 2. Basic concept of proposed unified architecture.

III. Unified Hardware for AEADs

A. Concept of Unification

The architecture being proposed here exploits the similarity found in many AEAD constructions. We first extract the common features, components, and subfunctions employed in the peripheral operations of many AEADs.

In this brief, we focus on the fact that several AEADs based on AES employ the five schemes as building blocks. Fig. 1 illustrates the block diagram structure of these subcomponents, where AES denotes the AES encryption with a secret key. First, (a) the counter mode encryption is widely used in a number of the AEADs constructed from a generic composition such as AES-GCM or AES-CCM. Second, (b) feedback or/and feedforward mechanisms constitute several of the peripheral processes in AEADs, such as the block chaining modes (e.g., cipher block chaining (CBC), cipher-based message authentication code (CMAC), and counter with CBC-MAC (CCM)), the output feedback mode (OFB), and the Feistel network in OTR. Third, (c) the XEX scheme model (and its generalized scheme) is regarded as one of the most promising schemes for constructing rate-1 AEADs, including AES-OTR. Fourth, (d) multiply-and-accumulate over GF using a GF multiplier performs GHASH in GCM and generates mask-values for rate-1 AEADs. Finally, (e) the accumulation of plaintext/ciphertext corresponds to the processing of ciphertext in AES-PMAC and the checksum in rate-1 AEADs.

Fig. 2 shows the conceptual block diagram of the proposed reconfigurable datapath, for unifying AEAD processing. In designing the five aforementioned subcomponents, the proposed reconfigurable datapath implements three functionalities: (i) AES core, (ii) MaskGen and (iii) Peripheral processing unit. (i) AES core is used for blockwise AES encryption/decryption process that is used in aforementioned components (a), (b), (c) and (e). (ii) MaskGen generates mask values required for components (b), (c), (d), and (e) by multiplication over Galois field. (iii) Peripheral processing unit supports various peripheral process used in (i) AES core. (iii) Peripheral processing unit consists of counter, bit-parallel XOR gates, feedback and feedforward scheme and accumulation module, which used in components (a), (b), (c) and (e).

The proposed architecture has scalability because various instantiations according to design goals (e.g., high-throughput

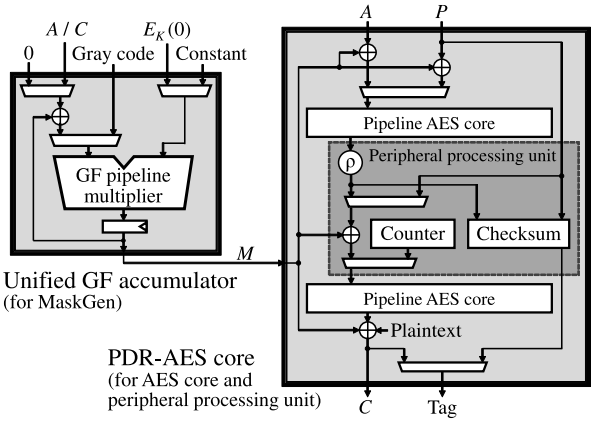

Fig. 3. High-throughput unified hardware.

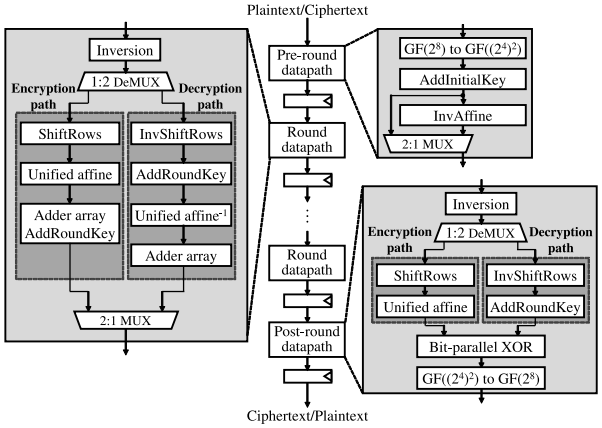

Fig. 4. Unrolled-pipeline AES encryption/decryption core.

or lightweight) are possible depending on the implementation method of aforementioned three functionalities.

B. High-Throughput Unified Hardware

This section presents a design of high-throughput unified hardware, as one of the instantiations of the aforementioned proposed concept for unification. The goal of our unified hardware is to perform various AEAD encryption/decryption procedures with a high throughput and efficiency. This constitutes one of the most desirable features in applications where multiple AEAD processing is required.

Fig. 3 shows an overall view of the unified hardware incorporating the aforementioned three functionalities. We expand the conceptual circuit shown in Fig. 2 to form the datapath, using two pipeline AES encryption/decryption cores. Functionality (i) AES core and (iii) peripheral processing unit are grouped and implemented as the peripheral-datapath-reconfigurable (PDR)-AES core. Such a dual-core architecture can perform AEADs based on a generic composition, as this architecture can simultaneously encipher the plaintext and generate the tag. In addition, such a dual-core is also an effective way for parallelizable AEADs to enhance both throughput and efficiency [9], [10]. Furthermore, non-rate-1 AEADs, which performs AES encryptions more than once for one plaintext block (e.g., AES-COLM [11]), can achieve high throughput by parallel processing with two AES cores implemented in series. In addition to this, a Checksum block is independently implemented, to perform the accumulations of plaintext and ciphertext in parallel. A linear mixing function • •is also included the peripheral operations which are not described in Section III-A. The latency and area of the linear mixing function • •are negligible compared to other critical blocks, such as the AES encryption/decryption core and GF multiplier.

Finally, the unified GF multiply-and-accumulator is used for computing the GHASH and mask-values. To support constant and sequential multiplication procedures, the GF multiply-and-accumulator utilizes the feedback of the multiplier output and a reconfigurable datapath at the input of the multiplier, to select operands appropriate for the operated AEAD. The feedback of the multiplier output is inputted to the multiplier after addition of the associated data for the multiply-and-accumulation, but it is also possible to construct a simple feedback structure by selecting 0 instead of the associated data as the value to be added. Simple feedback is used to realize a sequential constant multiplication, which is necessary for the mask generation for some AEADs, such as AES-OTR.

TABLE I
PERFORMANCE EVALUATION OF NAIVE AND PROPOSED ARCHITECTURES

	Platform	AEAD to be executed	Freq. [MHz]	Thru. [Gbps]	Area	Efficiency
Naive	FPGA	GCM	170.7	5.5	116,468	47
		OCB/COLM/OTR		21.9		188
	ASIC	Four modes	581.4	74.4	936,404	79
Proposed	FPGA	GCM	154.5	4.9	52,266	95
		OCB/COLM/OTR		19.8		378
	ASIC	Four modes	568.2	72.7	392,541	185

TABLE II
COMPARISON OF INDIVIDUAL AEAD CORES AND PROPOSED HARDWARE ON ASIC

Architecture	Freq. [MHz]	Thru. [Gbps]	Area [GE]	Efficiency [kbps/GE]
Ind. AES-GCM	632.9	81.0	172,977	468.3
Ind. AES-OCB	595.2	76.2	244,430	311.7
Ind. AES-COLM	584.8	74.9	280,757	266.6
Ind. AES-OTR	666.7	85.3	254,791	334.9
Proposed	568.2	72.7	392,541	185.3

The design of the AES encryption/decryption core has a significant impact on the overall performance of the system. To achieve the highest throughput and implementation efficiency, we employ an unrolled-pipeline AES architecture, which exploits the block-wise parallelism without deteriorating the operation frequency. Whereas we use a standard AES core with table-based S-box from [12] to derive our unrolled-pipeline architecture, for ASIC design, we focus on the most efficient AES encryption/decryption datapath [3] and extend it to achieve a higher throughput. Fig. 4 shows the block diagram of the proposed AES encryption/decryption core. The proposed core utilizes an unrolled-pipeline architecture, which is understood to be the most suitable for high-throughput implementation. More precisely, we unroll the round datapath and insert pipeline registers at the boundaries between rounds in the unrolled datapath. One major drawback of such an architecture is its large circuit area. However, the round datapath that we use manages to efficiently unify the tower-field

TABLE III
POWER ESTIMATION ON ASIC

Architecture	Power@500MHz [mW]			
	AES-GCM	AES-OCB	AES-COLM	AES-OTR
Individual	185	304	253	216
Proposed	189	330	285	254

inversion, which is one of the most area-consuming blocks. Accordingly, the area of our unrolled-pipeline AES encryption/decryption core is not as significant a problem as the AES core can perform various AEADs efficiently due to the unification. Thus, our core can achieve a high efficiency, owing to its high throughput as shown in Fig. 4.

C. Side-Channel Considerations

Whereas the AEAD focused in this brief is computationally secure against known cryptanalysis (or mathematical) attacks, it is important to discuss how to make our architecture resistant to physical attacks such as side-channel attack. Typically, attackers would attempt to retrieve the secret key from AES core(s) to break the confidentiality and authenticity provided by AEAD. Many side-channel-resistant AES implementations employ a masking countermeasure based on a masked inversion circuit such as [13]. Such a masking countermeasure can be easily implemented by replacing the inversion circuit with masked one and by duplicating the linear operations for a multi-party computation. Such a countermeasure induces non-negligible overhead of latency and area. However, these countermeasures would be easily integrated in our architecture because our architecture can employ any type of AES core, which means that we can easily exploit the tradeoff of latency/throughput and area by adopting an appropriate AES core (e.g., round-based and byte-serial ones) in order to tolerate the overhead incured by countermeasure.

IV. PERFORMANCE EVALUATION

For validation of our scheme, we implemented the proposed hardware using a Synopsys Design Compiler (I-2013.12-SP5) and Nangate 45 nm Open Cell Library. In addition, we synthesized the proposed hardware using a Vivado 2019.1 with Artix-7 (xc7a200tfbg484-3) as the target device. The individual implementations of inverse-free AEADs (e.g., AES-GCM and AES-OTR) adopt an AES encryption core that is derived by eliminating the decryption path of the AES encryption/decryption core described in Section III-B. Although the proposed hardware can perform a wide range of AES-based AEADs, we confirm here the operations of four major AEADs, namely, AES-GCM, AES-OCB, AESCOLM, and AES-OTR. In this brief, we evaluate the proposed hardware with a focus on these major AEADs owing to their presence.

Table I shows the synthesis result of the conventional straightforward hardware (denoted by Naive) and the proposed hardware, where Area indicates the number of LUTs and gate equivalents (GE) for FPGA and ASIC, respectively, and Efficiency indicates the throughput/area efficiency. Straightforward hardware internally employs four individual hardware of each of AEAD and select AEAD to be executed. Here, for AES-GCM and the proposed hardware, the GF multiplier is pipelined to make the delay almost equal to that of AES, so the throughput of these hardware are smaller than that of the other hardware. The throughput of proposed hardware is slightly (2.3% and 9.5% for ASIC and FPGA, respectively) smaller than straightforward hardware due to the path selectors for switching modes of operation. On the other hand, in terms of the circuit area, the proposed hardware is 58.1% and 55.2% smaller than the straightforward hardware for ASIC and FPGA, respectively. As a result, the proposed hardware can achieve 133% and 102% higher efficiency than the straightforward hardware on ASIC and FPGA, respectively. From the result, we confirm that the proposed hardware can support various AES-based AEADs including AES-GCM, AES-OCB, AES-COLM and AES-OTR with a comparable throughput and quite smaller area than the straightforward hardware.

TABLE IV
COMPARISON OF INDIVIDUAL AEAD CORES AND PROPOSED HARDWARE ON FPGA

Architecture	AEAD to be executed	Freq. [MHz]	Thru. [Gbps]	Area [LUTs]
Individual	AES-GCM	170.8	5.5	18,807
	AES-OCB	185.4	23.7	32,607
	AES-COLM	248.3	31.8	41,523
	AES-OTR	250.9	32.1	23,009
Proposed	AES-GCM	154.5	4.9	52,266
	AES-OCB/COLM/OTR		19.8	

Table II shows the synthesis result of the proposed hardware on ASIC. AES-COLM hardware is has the lowest maximum frequency among individual hardware because its critical path runs through linear mixing function in addition to AES round function. In contrast, the proposed hardware has a selector placed in front of the pipeline AES core, so the maximum operating frequency is slightly (2.8%) lower than that of AES-COLM. The throughput of the proposed hardware follows the same trend as frequency. In other words, the maximum operating frequency of the proposed hardware is given just by the subcomponent required in the supported AEAD, which means that the unification of AEAD hardware does not incur any overhead in terms of latency. On the other hand, the proposed hardware is implemented with only 40%–127% area overhead compared with the four individual hardware.

Table III shows the power consumption estimation, where we back-annotated the switching activity via timing simulation at 500 MHz using the above logic synthesis result. The power consumption estimation of the proposed hardware was obtained for each of the four AEADs. The power consumption of the proposed hardware is 2.2%–17.6% larger than that of individual hardware. However, such an increase in power consumption can be considered negligible because the proposed hardware implements more components than the individual hardware to support multiple AEADs. These result shows that the proposed hardware can be implemented on ASIC with small area overhead, a throughput and power consumption comparable to the individual hardware.

We also compare the performance of the proposed hardware with that of the individual hardware on FPGA. Table IV

TABLE V
COMPARISON OF PROPOSED HARDWARE WITH SOFTWARE EXECUTION FOR EACH AEAD

Platform	AEAD to be executed	Freq. [MHz]	Thru. [Gbps]	Power [mW]	Efficiency [Mbps/mW]
Intel Xeon E3-1220	AES-GCM	3.0e+3	26.1	8.0e+4	0.33
	AES-OCB		36.9		0.46
	AES-COLM		21.1		0.26
	AES-OTR		38.1		0.48
Proposed (ASIC)	AES-GCM	500	64.0	189	339
	AES-OCB			330	194
	AES-COLM			285	225
	AES-OTR			254	252

shows the synthesis results of these hardware. In Table IV, the throughput of the proposed hardware is different for AES-GCM and other AEADs and is 9.6%–38.4% lower than that of the individual hardware. On the other hand, the proposed hardware is implemented with 25.9%–178% area overhead compared with the four individual hardware. From the result, we confirm that the proposed hardware can be implemented with a comparable throughput and small area overhead with the individual hardware on FPGA as well as on ASIC.

Finally, Table V shows a performance comparison of software on a general-purpose processor and the proposed hardware on ASIC. We adopted Intel Xeon E3-1220 v5 as a general-purpose processor. Throughput and efficiency of Intel Xeon were calculated based on values provided by eBACS [14], the cryptosystem evaluation project. The values are obtained by actual measurement using the evaluation toolkit SUPERCOP (version 20191221) [15], in which AES-NI is used for each AEAD processing. Both hardware and software implementations use sufficiently long plaintext and associated data as input. The throughput of the proposed hardware is 68%–204% faster than that of the software implementation. Furthermore, the power consumption of the proposed hardware is 0.2%–0.4% of that of software implementation. These results show that the proposed hardware can be implemented very efficiently compared with software implementation in terms of both processing speed and power consumption.

Thus, we can confirm the proposed method has superiority in any implementation platform. Although we confirm the operation of four major AEADs by the proposed hardware and evaluate the proposed hardware with focusing on the four AEADs in this brief, our architecture can be applied to many other AES-based AEADs. This indicates that, if we should support a wider variety of AES-based AEADs, the proposed hardware should be more advantageous, because the proposed architecture can perform a wide variety of AEADs with the minimal number of components.

V. CONCLUSION

In this brief, we presented a design of unified AEAD hardware architecture, based on reconfigurable peripheral datapaths. The main idea of the proposed hardware is to use the similarity of many AEADs to construct a new reconfigurable datapath, which is capable of supporting their peripheral operations with the minimal number of subcomponents. In addition, we apply our concept to the design of high-throughput AES-based unified AEAD hardware. We then validate the proposed hardware through an experimental implementation. We compare the performance of the proposed hardware with the hardware implementations dedicated to AES-GCM, -OCB, -COLM, and -OTR. Based on the results, we confirm that our hardware can support the four AEADs with high efficiency in comparison with the conventional hardware. In addition, we confirmed that the proposed hardware is superior to software implementation on general-purpose processor.

We evaluated the performance of proposed architecture using the logic synthesis and gate-level timing simulation, where the effect of place and route (including wire delay and parasitic elements) were not evaluated. A detailed evaluation after the place and route would remain in future work. However, our architecutre would be advantageous even after the place and route because the proposed architecture is smaller in area and do not have any extra global wires that have an impact on the critical path. In addition, the extension of our unified hardware to other performance goals and other AEAD schemes also remains to be investigated in future works.

REFERENCES

[1] D. J. Bernstein. (Jan. 2014). *Cryptographic Competitions*. [Online]. Available: https://competitions.cr.yp.to/index.html

[2] *Lightweight Cryptography*, NIST, Gaithersburg, MD, USA, Oct. 2019. [Online]. Available: https://csrc.nist.gov/Projects/lightweight-cryptography

[3] R. Ueno, S. Morioka, N. Homma, and T. Aoki, "A high throughput/gate AES hardware architecture by compressing encryption and decryption datapaths," in *Cryptographic Hardware and Embedded Systems—CHES 2016*, Heidelberg, Germany: Springer, 2016, pp. 538–558.

[4] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*, NIST Spec. Publ., Gaithersburg, MD, USA, 2007.

[5] T. Krovetz and P. Rogaway. (Sep. 2016). *OCB (V1.1)*. [Online]. Available: https://competitions.cr.yp.to/round3/ocbv11.pdf

[6] K. Minematsu. (Sep. 2016). *AES-OTR V3.1*. [Online]. Available: https://competitions.cr.yp.to/round3/aesotrv31.pdf

[7] D. Goudarzi *et al.* (2019). *Pyjamask V1.0*. https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/Pyjamask-spec.pdf

[8] C. Beierle *et al.* (2019). *SKINNY-AEAD and SKINNY-Hash V1.1*.. [Online]. Available: https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/SKINNY-spec-round2.pdf

[9] Y. Hori, A. Satoh, H. Sakane, and K. Toda, "Bitstream encryption and authentication with AES-GCM in dynamically reconfigurable systems," in *Proc. Int. Conf. Field Program. Logic Appl. (FPL)*, 2008, pp. 23–28.

[10] R. Ueno, N. Homma, T. Iida, and K. Minematsu, "High throughput/gate FN-based hardware architectures for AES-OTR," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Sapporo, Japan, 2019, pp. 1–4.

[11] E. Andreeva *et al.*, (Sep. 2016). *COLM V1*. [Online]. Available: https://competitions.cr.yp.to/round3/colmv1.pdf

[12] *Computer Structures Laboratory*, Tohoku Univ., Sendai, Japan, Jan. 2020. [Online]. Available: http://www.aoki.ecei.tohoku.ac.jp/index.html

[13] E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen, "A side-channel analysis resistant description of the AES S-box," in *Fast Software Encryption—FSE*. Heidelberg, Germany: Springer, 2005, pp. 413–423.

[14] *eBACS: ECRYPT Benchmarking of Cryptographic Systems*, VAMPIRE, Dec. 2019. [Online]. Available:https://bench.cr.yp.to/results-aead.html

[15] *SUPERCOP*, VAMPIRE, Dec. 2019. [Online]. Available: https://bench.cr.yp.to/supercop.html