NISTIR 8369

Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process

Meltem Sönmez Turan
Kerry McKay
Donghoon Chang
Çağdaş Çalık
Lawrence Bassham
Jinkeon Kang
John Kelsey

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8369

National Institute of
Standards and Technology
U.S. Department of Commerce

NISTIR 8369

Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process

Meltem Sönmez Turan
Kerry McKay
Donghoon Chang
Çağdaş Çalık
Lawrence Bassham
Jinkeon Kang
John Kelsey

Computer Security Division
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8369

July 2021

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
James K. Olthoff, Performing the Non-Exclusive Functions and Duties of the Under Secretary of Commerce for Standards and Technology & Director, National Institute of Standards and Technology

National Institute of Standards and Technology
Interagency or Internal Report 8369
Natl. Inst. Stand. Technol. Interag. Intern. Rep. 8369, 81 pages (July 2021)

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8369

Abstract

The National Institute of Standards and Technology (NIST) initiated a public standardization process to select one or more Authenticated Encryption with Associated Data (AEAD) and hashing schemes suitable for constrained environments. In February 2019, 57 candidates were submitted to NIST for consideration. Among these, 56 were accepted as first-round candidates in April 2019. After four months, NIST selected 32 of the candidates for the second round. In March 2021, NIST announced 10 finalists to move forward to the final round of the selection process. The finalists are ASCON, Elephant, GIFT-COFB, Grain-128AEAD, ISAP, PHOTON-Beetle, Romulus, SPARKLE, TinyJAMBU, and Xoodyak. This report describes the evaluation criteria and selection process, which is based on public feedback and internal review of the second-round candidates.

Keywords

authenticated encryption · cryptography · hash functions · lightweight cryptography

Acknowledgments

NIST thanks the second-round submission teams, who developed and designed the second-round candidates, and the cryptographic community, who analyzed the candidates, shared their comments through the `lwc-forum`, and published papers on various technical aspects of the candidates.

NIST also thanks the developers, who provided optimized implementations of the candidates as well as the hardware and software benchmarking initiatives, for their contribution to the understanding of the performance characteristics of the algorithms on various target platforms.

Specifically, NIST thanks all those who contributed to the following projects: (*i*) FPGA benchmarking by K. Mohajerani, R. Haeussler, R. Nagpal, F. Farahmand, A. Abdulgadir, J.-P. Kaps, and K. Gaj; (*ii*) ASIC benchmarking by M. Aagaard and N. Zidarič; (*iii*) ASIC benchmarking by M. Khairallah, T. Peyrin, and A. Chattopadhyay; (*iv*) Microcontroller benchmarking by S. Renner, E. Pozzobon, and J. Mottok; (*v*) Microcontroller benchmarking by R. Weatherley; (*vi*) RISC-V benchmarking by F. Campos, L. Jellema, M. Lemmen, L. Müller, D. Sprenkels, and B. Viguier; (*vii*) RISC-V benchmarking by G. Nisanci, R. Atay, M. K. Pehlivanoglu, E. B. Kavun, and T. Yalçın; and (*viii*) eBACS (ECRYPT Benchmarking of Cryptographic Systems) benchmarking by D. J. Bernstein and T. Lange.

The authors of this report acknowledge and appreciate contributions from their colleagues at NIST – Lily Chen, Andrew Regenscheid, Sara Kerman, Noah Waller, Isabel Van Wyk, Ray Perlner, Luís Brandão, Sheryl Taylor, Dustin Moody, and Michael J. Fagan – who provided technical and administrative support and participated in meetings to discuss the selection of the finalists.

Contents

List of Tables

List of Figures

Acronyms

AD	Associated Data
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
ARX	Addition-Rotation-Xor
ASIC	Application Specific Integrated Circuit
CAESAR	Competition for Authenticated Encryption: Security, Applicability, and Robustness
CBC	Cipher Block Chaining
CCA	Chosen Ciphertext Attack
CCAm	CCA with nonce misuse-resilience
CI	Ciphertext Integrity
CIM	Ciphertext Integrity with Misuse-resistance
CTR	Counter
eBACS	ECRYPT Benchmarking of Cryptographic System
eSTREAM	ECRYPT STREAM cipher project
FOBOS	Flexible Open source workBench fOr Side-channel analysis
FPGA	Field Programmable Gate Array
GCM	Galois/Counter Mode
GE	Gate Equivalents
INT-RUP	INTegrity under RUP
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
LE	Logic Elements
LFSR	Linear Feedback Shift Register
LUT	Look-Up Table

MAC	Message Authentication Code
MCU	Microcontroller Unit
MD	Merkle-Damgård
MDS	Maximum Distance Separable
MILP	Mixed-Integer Linear Programming
MITM	Man-In-The-Middle
MMO	Matyas-Meyer-Oseas
NIST	National Institute of Standards and Technology
NISTIR	NIST Internal Report
OCB	Offset CodeBook
OFB	Output FeedBack
RAM	Random Access Memory
RFID	Radio Frequency Identification
RISC	Reduced Instruction Set Computer
ROM	Read Only Memory
RUP	Releasing Unverified Plaintext
SFS	Semi-Free-Start
SHA	Secure Hash Algorithm
TBC	Tweakable Block Cipher
XOF	eXtendable-Output Function
XOR	eXclusive OR

1. Introduction

The National Institute of Standards and Technology (NIST) initiated the Lightweight Cryptography Standardization Process to solicit, evaluate, and standardize one or more Authenticated Encryption with Associated Data (AEAD) and hashing schemes that are suitable for use in constrained environments, such as in radio frequency identification (RFID) tags and sensor networks, where the performance of current NIST cryptography standards is not acceptable or is deficient.

NIST received 57 submissions in response to its call for algorithms; and in April 2019, NIST announced 56 first-round candidates. In August 2019, NIST announced 32 second-round candidates and published NIST Internal Report (NISTIR) 8268 [1] to explain the evaluation criteria and selection of the second-round candidates. Later, NIST published the second-round packages of the candidates, which included corrections to typographical errors, bug fixes, and additional supplementary content such as optimized implementations and new security analysis. At this stage, the designers were not allowed to tweak their designs.

In August 2020, NIST invited the submitters of the second-round candidates to provide a short update on their algorithms, specifically on (1) new proofs/arguments that support the security claims, (2) new software and hardware implementations (including ones that protect against side-channel attacks), (3) new third-party analysis and its implications, (4) platforms and metrics in which the candidate performs better than current NIST standards, (5) target applications and use cases for which the candidate is optimized, (6) planned tweak proposals if the submission is accepted as a finalist, and any other relevant information. In September 2020, NIST received status updates from 27 (out of 32) teams [2–28].

During the second round of the process, NIST hosted the third and fourth lightweight cryptography workshops to discuss various aspects of the second-round candidates and obtain valuable feedback for the selection of the finalists. The timeline of the standardization process is summarized in Table 1. In March 2021, NIST announced the finalists of the standardization process, namely:

- ASCON
- Elephant
- GIFT-COFB
- Grain-128AEAD
- ISAP
- PHOTON-Beetle
- Romulus
- SPARKLE
- TinyJAMBU
- Xoodyak

The purpose of this report is to provide a public record of the second round of the standardization process. The report describes the evaluation criteria and selection of the finalists. In Section 2 explains the evaluation criteria for the second round. Section 3 provides the classification of second-round candidates and a discussion for each candidate. Section 4 lists the software and hardware benchmarking initiatives. Finally, Section 5 explains the selection of the finalists. Appendix A provides additional information about NIST's internal software benchmarking results.

Table 1. Timeline of the NIST Lightweight Cryptography Standardization Process

Date	*Event*
July 2015	First Lightweight Cryptography Workshop at NIST
October 2016	Second Lightweight Cryptography Workshop at NIST
March 2017	NISTIR 8114 Report on Lightweight Cryptography [29]
April 2017	(draft) Profiles for Lightweight Cryptography Standardization Process [30]
August 2018	Federal Register Notice
	Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process [31]
February 2019	Submission deadline
April 2019	Announcement of the first-round candidates
August 2019	Announcement of the second-round candidates
October 2019	NISTIR 8268, Status Report on the First Round of the NIST Lightweight Cryptography Standardization Process [1]
November 2019	Third Lightweight Cryptography Workshop at NIST
September 2020	Submission deadline for optional status updates
October 2020	Fourth Lightweight Cryptography Workshop (virtual)
March 2021	Announcement of the finalists

2. Evaluation Criteria

The evaluation criteria for the standardization process were published in August 2018 [31]. These criteria were further discussed and clarified during the NIST Lightweight Cryptography workshops. This section summarizes the evaluation criteria used during the second round of the standardization process.

The cryptographic *security* of the candidates is the most important criterion. Similar to the selection of the second-round candidates, the submissions with significant third-party analysis or that based their security claims on well-understood design principles and security proofs were favored during the selection of finalists. NIST studied the feedback received from the cryptographic community and the documents provided by the submitters in addition to its own internal analysis. The security evaluations of the candidates are summarized in Section 3.2.

The second criterion is the performance of the candidates in applications using constrained devices (i.e., the *hardware and software performance* of candidates in constrained environments). Candidates were evaluated and compared in terms of various performance and cost metrics, and those that performed significantly better than the current NIST standards (especially AES-GCM [32] and SHA-2 [33]) were favored during the selection. NIST considered various public software and hardware benchmarking efforts as well as internal software benchmarking comparisons (see Section 4 and Appendix A).

Side-channel resistance of the candidates was another criterion. Although there were not many comprehensive comparisons of the candidates, NIST considered the claims from the submission documents and related papers from the literature. More information is provided in Section 3.3.1.

Although not explicitly required by the submission call, there were some additional properties considered during the selection when multiple candidates had similar security and performance evaluations, including *nonce-misuse security* (see Section 3.3.2), *releasing unverified plaintext (RUP) security* (see Section 3.3.3), the *impact of state recovery* (see Section 3.3.4), and *post-quantum security* of the candidates (see Section 3.3.5). NIST also considered tweak plans, the diversity of the candidates, and their suitability for addition to NIST's portfolio of cryptographic standards.

3. Second-Round Candidates

3.1 Classification of the Second-Round Candidates

This section provides a classification of the second-round candidates [34–65] based on their underlying primitives and modes of operation. Similar comparisons are available in [66, 67].

The submission packages were allowed to include a family of algorithms supporting different parameters, such as key and nonce sizes. To allow for fair comparison, each family was required to include a *primary* variant that met minimum criteria. Some of the candidate families include more than one variant for AEAD and hashing, and the total number of variants is 89 for AEAD and 19 for hashing. Out of 32 second-round candidates, 11 of them have single variants. On average, candidates have 2.7 AEAD variants and 1.5 hash variants. The comparisons provided in this section are mostly based on the primary variants.

The underlying primitives of the second-round candidates are summarized in Table 2. Block ciphers or tweakable block ciphers are usually used by candidates aiming for AEAD-only functionality, whereas permutations are generally used when both AEAD and hashing functionalities are targeted.

There are many candidates that share a particular primitive or primitives that are very closely related (e.g., some changes to the primitive are present, but previous security analysis of the primitive may still apply). For example, the modes presented in COMET and SAEAES are instantiated using the block cipher AES, whereas the candidates ESTATE and mixFeed use slightly modified variants of AES. The well-analyzed block cipher GIFT is also used in multiple candidates, namely ESTATE, GIFT-COFB, HyENA, LOTUS-AEAD, LOCUS-AEAD, and SUNDAE-GIFT. Similarly, the tweakable block cipher SKINNY is used in ForkAE, Romulus, SKINNY-AEAD and SKINNY-HASH. Among permutation-based candidates, the ASCON permutation and its variants are used in ASCON, DryGASCON, and

Table 2. Underlying primitives of the second-round candidates

Candidates providing AEAD-only functionality	
Permutation	Elephant, ISAP, Oribatida, SPIX, SpoC, Spook[3], WAGE
Block Cipher	COMET, GIFT-COFB, HyENA, mixFeed, Pyjamask, SAEAES, SUNDAE-GIFT, TinyJAMBU[1]
Tweakable Block Cipher	ESTATE, ForkAE, LOTUS-AEAD and LOCUS-AEAD, Romulus, Spook
Stream Cipher	Grain-128AEAD
Candidates providing AEAD and hashing functionalities	
Permutation	ACE, ASCON, DryGASCON, Gimli, KNOT, ORANGE, PHOTON-Beetle, SPARKLE, Subterranean 2.0, Xoodyak
Block Cipher	SATURNIN[2]
Tweakable Block Cipher	SKINNY-AEAD and SKINNY-HASH

[1] The underlying primitive of TinyJAMBU is given as keyed permutation in [63].
[2] The underlying primitive of SATURNIN [55] can also be considered a tweakable block cipher due to the use of additional 9-bit parameter for domain separation.
[3] Spook design uses both a permutation and a tweakable block cipher.

ISAP; the KECCAK permutation is used in Elephant and ISAP; the PHOTON permutation is used in ORANGE and PHOTON-Beetle; and the sLiSCP-light permutation is used in SPIX and SpoC.

Table 3 classifies the AEAD modes of operation of the primary variants as sequential and parallel modes. For sequential modes, sponge constructions and their variants are commonly used in addition to modes based on block ciphers and tweakable block ciphers, whereas parallel modes are typically based on (tweakable) block ciphers.

Twelve of the 32 second-round candidates offer hashing functionality. Except for SATURNIN, all of these candidates are based on the sponge construction or its variants [68] for hashing (see Table 4). SATURNIN uses an Matyas-Meyer-Oseas (MMO)-based compression function [69] instantiated with a tweakable block cipher, and its hash algorithm is constructed using a Merkle-Damgård (MD) construction [70, 71].

3.2 Evaluation of the Second-Round Candidates

This section includes an evaluation of the second-round candidates based on the evaluation criteria described in Section 2. This section provides a summary of the design and security analysis for each candidate, as well as tweak plans. Performance summaries can be found in Section 4.

Table 3. AEAD mode-of-operation classification of primary variants

Sequential modes	
Classical sponge with public permutation	DryGASCON, Gimli, KNOT, Subterranean 2.0, Xoodyak
Classical sponge with stronger initialization and finalization	ACE, ASCON, SPIX, Spook, WAGE
Classical sponge with keyed permutation	SAEAES, TinyJAMBU
Modified sponge with public permutation	ORANGE, Oribatida, PHOTON-Beetle SPARKLE, SpoC
Block cipher or tweakable-block-cipher based feedback (rate 1)	COMET, GIFT-COFB, HyENA, mixFeed, Romulus
Encrypt-then-MAC	ISAP, SATURNIN
MAC-then-Encrypt	ESTATE, SUNDAE-GIFT
Stream cipher based	Grain-128AEAD
Parallel modes	
ΘCB3-based (rate 1)	SKINNY-AEAD
OCB3-based (rate 1)	Pyjamask
Enc-then-MAC	Elephant
Others	ForkAE, LOTUS-AEAD

3.2.1 ACE

ACE, designed by Aagaard et al. [34], is a permutation-based AEAD and hashing scheme. The AEAD scheme uses a unified sponge duplex mode, whereas the hash function is based on the sponge construction. The ACE permutation is based on a 5-block generalized version of sLiSCP-light [72]. Nonlinearity is provided by a round-reduced unkeyed instance of Simeck-64, denoted `SB-64` [73].

Variants. The variants of the ACE family are listed below.

AEAD variants	*Key*	*Nonce*	*Tag*	*# Steps*	**Hash variants**	*Digest size*	*# Steps*
ACE-$\mathcal{AE}$-128	128	128	128	16	**ACE-$\mathcal{H}$-256**	256	16

Security Analysis. Liu et al. [74] constructed two 8-step impossible differentials for the ACE permutation and showed that there are no impossible differential distinguishers longer than nine steps. Various analyses [75–78] on Simeck64/128 describe attacks exceeding the

Table 4. Hashing modes of the 12 candidates with hashing functionalities

Hashing modes	***Candidates (digest sizes)***
Sponge construction	ACE (256), ASCON (256), DryGASCON (256,512), Gimli (256), KNOT (256, 384, 512), ORANGE (256), PHOTON-Beetle (256), SKINNY-HASH (256), SPARKLE (256, 384), Subterranean 2.0 (256), Xoodyak (256)
MD construction based on MMO mode	SATURNIN (256)

eight rounds of SB-64. The implications of these analyses on the ACE permutation need more investigation.

Tweak plan. No tweak plan has been proposed [2].

3.2.2 ASCON

ASCON, designed by Dobraunig et al. [35], is a permutation-based AEAD and hashing scheme. ASCON-AEAD is based on the monkeyDuplex construction [79] with additional key additions during initialization and finalization. ASCON-Hash is based on the duplex sponge construction [80]. The main component of the ASCON family is a 320-bit permutation instantiated with different constants and number of rounds for different variants. ASCON was selected as the primary choice for lightweight authenticated encryption in the final portfolio of the CAESAR competition (2014–2019).

Variants. The variants of the ASCON family are listed below. The number of rounds for the AEAD variants is represented as a triplet a, b, and c, where a is the number of rounds during initialization, b is the number of rounds during AD or message processing, and c is the number of rounds during finalization. The primary AEAD and hash variants appear in bold face.

AEAD variants	*Key*	*Nonce*	*Tag*	*# Rounds*	**Hash variants**	*Digest size*	*# Rounds*
ASCON-128	128	128	128	12, 6, 12	**ASCON-Hash**	256	12
ASCON-128a	128	128	128	12, 8, 12	ASCON-Xof	256	12
ASCON-80pq	160	128	128	12, 6, 12			

Security Analysis. ASCON has received significant third-party analysis (e.g., [35, 81–91]). The existing security analyses are summarized in [92], and a summary is provided in Table 5. Additionally, Ramezanpour et al. [93] evaluated ASCON's vulnerability to both passive and active side-channel attacks.

Table 5. Summary of attacks on ASCON family

Target	*Attack type*	*Method*	*Rounds*[1]	*Data*	*Time*	*Memory*	*Nonce misuse*	*Ref.*
ASCON-128 ASCON-128a	Key recovery	Cube	7,$\star$,$\star$	$2^{77.2}$	$2^{103.92}$	-	No	[85]
		Cube	7,5,$\star$	2^{50}	2^{98}	2^{41}	Yes	[86]
		Conditional cube	6,$\star$,$\star$	2^{40}	2^{40}	-	No	[85]
	Forgery	Cube	$\star$,$\star$,6	2^{33}	-	-	Yes	[86]
		Cube	$\star$,$\star$,5	2^{17}	-	-	Yes	[86]
ASCON-128	State recovery	Cube	$\star$,5,$\star$	2^{18}	2^{66}	2^{54}	Yes	[86]
		SAT-solver	$\star$,2,$\star$	-	practical		No	[87]
ASCON-128a	State recovery	Cube	$\star$,5,$\star$	2^{18}	2^{64}	2^{38}	Yes	[86]
ASCON Permutation	Distinguishing	Zero-sum	12	-	2^{130}	-	-	[84]
		Zero-sum	11	-	2^{85}	-	-	[35]
		Integral	11	-	2^{315}	-	-	[82]
ASCON-Hash[2] ASCON-Xof[2]	Preimage	Algebraic	6,6,6	-	$2^{63.3}$	-	-	[90]
		Cube	2,2,2	-	2^{39}	-	-	[90]
ASCON-Hash ASCON-Xof	SFS Collision	Differential	2,2,2	-	practical	-	-	[90]

[1] The symbol $\star$ denotes arbitrary number of rounds.
[2] The variant has 64-bit hash output.

Tweak plan. No tweak plan has been proposed [3].

3.2.3 COMET

COMET (COunter Mode Encryption with authentication Tag), designed by Gueron et al. [36], is a single-pass, inverse-free block-cipher mode of operation that uses re-keying techniques to minimize the state size and the number of operations. The mode is instantiated using the block ciphers AES-128 [94], CHAM-128 [95], and Speck-64 [96].

Variants. COMET supports 64-bit and 128-bit block sizes, and the variants of the COMET family are listed below.

AEAD variants	*Key*	*Nonce*	*Tag*
COMET-128_AES-128/128	128	128	128
COMET-128_CHAM-128/128	128	128	128
COMET-64_CHAM-64/128	128	120	64
COMET-64_Speck-64/128	128	120	64

Security Analysis. COMET designers provided a proof in the ideal cipher model showing that COMET-128 is secure when the data complexity is less than 2^{64} bytes and the offline computation complexity, including block cipher evaluations, is less than 2^{119} [97]. Khairallah [98] showed the existence of 2^{64} weak keys and demonstrated an existential forgery attack with weak keys using 2^{64} online queries. Bernstein et al. [99] presented two observations on COMET: the first observation uses a long message to detect the use of weak keys; and the second focuses on the resistance of COMET against slide attacks.

Tweak plan. Although the results presented in [98, 99] do not invalidate the security claims of COMET, the designers considered the following tweaks [4]: (1) updating the place where control bits are XORed to save additional n-bit memory and (2) updating the `permute` function that modifies the key to increase the data complexities of the available attacks. The designers provided a security proof for the mode of this updated proposal [100].

3.2.4 DryGASCON

DryGASCON, designed by Riou [37], is an AEAD and hashing scheme that aims to prevent many physical attacks at the algorithmic level. DryGASCON is based on the DrySponge construction, which is derived from the duplex sponge construction [80]. The underlying GASCON permutations are constructed by tweaking the ASCON permutation [35].

Variants. The variants of the DryGASCON family are listed below. In the table, s represents the size of the GASCON permutation, and the number of rounds for AEAD variants is represented as a tuple a and b, where a is the number of rounds during the diversification phase (i.e., nonce-processing phase) and b is the number of rounds used in remaining phases.

AEAD variants	*Key*	*Nonce*	*Tag*	*s*	*# Rounds*	**Hash variants**	*Digest size*	*s*	*# Rounds*
DryGASCON128	128	128	128	320	11, 7	**DryGASCON128**	256	320	7
DryGASCON256	256	128	256	576	12, 8	DryGASCON256	512	576	8

Security Analysis. The designer of DryGASCON argues that the GASCON permutations have security properties similar to the ASCON permutation [37]. Tezcan reported a 3.5 round truncated differential with a probability of one [91].

Tweak plan. The designer has described three possible tweaks [5]: (1) adding an XOF mode; (2) uniformization of the key profiles to allow for mapping all keys from the small key profile to keys in the fast and full profile to achieve "upward" interoperability; and (3) allowing precomputation over AD by swapping the order of computation between nonce and AD, which involves removing support for the "static data" feature.

3.2.5 Elephant

Elephant, designed by Beyne et al. [38], is a permutation-based AEAD scheme that follows a nonce-based encrypt-then-MAC construction with a counter mode and a variant of the

Wegman-Carter-Shoup MAC function. Elephant is the only second-round candidate that uses a parallel mode with a permutation. The mode is instantiated using the permutations of Spongent [101] and KECCAK [102].

Variants. The variants of the Elephant family are listed below.

AEAD variants	*Key*	*Nonce*	*Tag*	*Permutation*	*# Rounds*
Dumbo	128	96	64	160-bit Spongent	80
Jumbo	128	96	64	176-bit Spongent	90
Delirium	128	96	128	200-bit KECCAK	18

Security Analysis. The designers of Elephant [38] provided privacy and authenticity proofs of the mode of Elephant in the nonce-respecting scenario under the ideal permutation assumption. As confirmed by the designers of Elephant [6], the mode of Elephant does not provide authenticity in the nonce-misuse scenario. Zhou et al. [103] presented an interpolation attack on 8-round (out of 18) Delirium in the nonce-respecting scenario with 2^{70} data complexity and $2^{98.3}$ XOR operations and 2^{70} memory complexity. Sun et al. [104] presented a zero-sum distinguishing attack on 176-bit Spongent permutation of 21-round (out of 90 rounds) with 2^{159} time complexity. Bogdanov et al. [105] presented differential distinguishers on 160-bit Spongent permutation of 40-round (out of 80) with probability 2^{-160} and 176-bit Spongent permutation of 44-round (out of 90) with probability 2^{-176}. Bogdanov et al. [105] presented linear distinguishers on 160-bit Spongent permutation of 80-round (out of 80) with correlation probability 2^{-160} and 176-bit Spongent permutation of 90-round (out of 90) with probability 2^{-180}. Zhang and Liu [106] presented a truncated-differential distinguisher on 176-bit Spongent permutation of 46-round (out of 90) with probability $2^{-174.415}$. In [107, 108], the security of Elephant against key recovery attacks were analyzed in quantum settings.

Tweak plan. The designers described two possible tweaks related to the mode [6, 109]: (1) changing the index pair of every block, which is used to generate a masking key for that block computation, and (2) changing from a Wegman-Carter-Shoup style authenticator to a protected counter sum style authenticator to provide authenticity even in nonce-misuse.

3.2.6 ESTATE

ESTATE, designed by Chakraborti et al. [39], is an AEAD scheme based on tweakable block ciphers. The mode of ESTATE uses a nonce-based MAC-then-Encrypt construction, where the encryption (based on the Output Feeback (OFB) mode) is done by XORing a message and a keystream, and the authentication/verification is based on a tweakable variant of FCBC [110] by using different tweak values for the last block processing, depending on whether the last block of input is full or partial. The ESTATE mode, inspired by SUNDAE [111], uses a tweakable block cipher to minimize the number of underlying primitive calls and remove field multiplications from the mode level. The ESTATE mode provides privacy and authenticity even in nonce-misuse and RUP attack models [112, 113].

Variants. The variants of the ESTATE family are listed below. ESTATE is instantiated using three new tweakable block ciphers with 4-bit tweak: (1) TweAES-128 (adds the tweak at every second round of AES-128); (2) TweGIFT-128 (adds the tweak at every fifth round of GIFT-128); and (3) TweAES-128-6 (adds the tweak at every second round of the first six rounds of AES-128 and includes the MixColumns operations in the last round).

AEAD variants	*Key*	*Nonce*	*Tag*
ESTATE_TweAES-128	128	128	128
ESTATE_TweGIFT-128	128	128	128
sESTATE_TweAES-128-6	128	128	128

Security Analysis. The security of the new block ciphers TweAES and TweGIFT relies on the available analysis on AES and GIFT. The designers of ESTATE [39, 113] reported that they were unable to find any vulnerability of the tweak injection after conducting in-depth self-analysis. In [112, 113], privacy, authenticity, and INT-RUP security proofs of ESTATE in the nonce-misuse scenario were provided. In [108], the security of sESTATE against the key recovery attack was analyzed in quantum settings.

Tweak plan. No tweak plan has been proposed except for two bug fixes in the reference implementation [7].

3.2.7 ForkAE

ForkAE, designed by Andreeva et al. [40], is an AEAD scheme that uses the two modes of operations – SAEF (Sequential AEAD from a Forkcipher) and PAEF (Parallel AEAD from a Forkcipher) – and the forkcipher ForkSkinny. A Forkcipher [114–116] enables the evaluation of a tweakable block cipher on the same message and two different keys with an amortized computational cost to optimize short message handling.

Variants. The variants of the ForkAE family are listed below.

AEAD variants	*Key*	*Nonce*	*Tag*	*Tweakey*	*Block size*
PAEF-ForkSkinny-128-288	128	104	128	288	128
PAEF-ForkSkinny-128-192	128	48	128	192	128
PAEF-ForkSkinny-128-256	128	112	128	256	128
PAEF-ForkSkinny-64-192	128	48	64	192	64
SAEF-ForkSkinny-128-192	128	56	128	192	128
SAEF-ForkSkinny-128-256	128	120	128	256	128

Security Analysis. Although ForkSkinny is a new forkcipher design, the candidate benefits from the extensive analysis on the tweakable block cipher Skinny. Bariant et al. [117] showed that attacks on Skinny can be extended to one extra round for most ForkSkinny variants and up to three rounds for ForkSkinny-128-256. Privacy and authenticity proofs of PAEF and SAEF in the nonce-respecting scenario were provided in [40]. The RUP-security of SAEF was provided in [118]. The online nonce-misuse security of SAEF was given in [119]. The existing security analyses on ForkSkinny and other new modes based on a forkcipher were summarized in [120].

Tweak plan. The designers described the following tweaks [8]: (1) reducing the number of rounds in ForkSkinny; (2) including new instances of RPAEF (Reduced Parallel AEAD from a forkcipher) mode, which are optimized to handle long queries; and (3) possibly including new CTR-like modes that achieve beyond-birthday-bound security and improve efficiency for encryption-only queries.

3.2.8 GIFT-COFB

GIFT-COFB, designed by Banik et al. [41], is a rate-1 AEAD scheme based on the 128-bit block cipher GIFT-128 [121, 122]. The mode of GIFT-COFB is slightly different from the original COmbined FeedBack (COFB) mode introduced in [123, 124]. When a 128-bit block cipher is used, the original COFB mode uses a 64-bit nonce and a feedback with no entropy loss. Alternatively, GIFT-COFB uses a 128-bit nonce and hardware-efficient feedback with 1-bit entropy loss and handles empty data by changing its padding method.

Variants. The single variant of the GIFT-COFB family is listed below.

AEAD variants	*Key*	*Nonce*	*Tag*
GIFT-COFB	128	128	128

Security Analysis. The underlying block cipher GIFT has received a large number of third-party analyses and still has a high security margin (see Table 6). An extended list of third-party analyses on GIFT-128 is provided in [9].

Tweak plan. The designers do not plan to tweak their submission [9].

3.2.9 Gimli

Gimli, designed by Bernstein et al. [42], is a permutation-based AEAD and hashing scheme. The AEAD scheme Gimli-Cipher is based on a duplex mode of operation, and Gimli-Hash is based on a sponge mode of operation, both having 128-bit rate and 256-bit capacity. The scheme is based on the 384-bit Gimli permutation with 24 rounds, where the state is a 3×4 matrix of 32-bit words. The Gimli permutation comprises three layers: (1) the nonlinear

Table 6. A non-exhaustive summary of attacks on GIFT-128

Attack	*Rounds*	*Setting*	*Data*	*Time*	*Memory*	*Ref.*
Differential	22/40	single key	2^{114}	2^{114}	2^{53}	[125]
Differential	26/40	single key	$2^{124.4}$	$2^{124.4}$	2^{109}	[126]
Boomerang	21/40	related key	$2^{126.6}$	$2^{126.6}$	–	[127]
Boomerang	22/40	related key	$2^{112.6}$	$2^{112.6}$	2^{52}	[128]
Rectangle	23/40	related key	$2^{121.3}$	$2^{126.9}$	$2^{121.6}$	[128]

layer with the SP-box on 96 bits is applied to every column of each round; (2) the linear layer with one of two swap operations (small-swap or big-swap), is applied only to the first row in every second round; and (3) a 32-bit round constant is added to the state in every fourth round.

Variants. The variants of the Gimli family are listed below. Additionally, the designers proposed Gimli-XOF by including the output length in the Gimli-Hash initialization.

AEAD variants	*Key*	*Nonce*	*Tag*	*# Rounds*	**Hash variants**	*Digest size*	*# Rounds*
Gimli-Cipher	256	128	128	24	**Gimli-Hash**	256	24

Security Analysis. Since the Gimli permutation was first introduced in 2017 [129], several in-depth analyses [89, 130–135] have been conducted. For each scheme, the best-known attacks are shown in Table 7. Since the small-swap and big-swap operations are applied only to the first row in every second round, the Gimli permutation has slow diffusion, which led to a larger number of rounds being analyzed especially by collision and distinguishing attacks.

Tweak plan. The designers did not plan to tweak their submission [10].

3.2.10 Grain-128AEAD

Grain-128AEAD, designed by Hell et al. [43], is a bit-oriented feedback shift register based AEAD scheme optimized for hardware implementations. In 2008, Grain v1 was selected as a finalist in the hardware profile of the eSTREAM portfolio [136]. The standard ISO/IEC 29167-13:2015 [137] includes Grain-128a for RFID systems.

Variants. The AEAD variant of the Grain-128AEAD family is listed below.

AEAD variants	*Key*	*Nonce*	*Tag*
Grain-128AEAD	128	96	64

Table 7. Summary of attacks on Gimli

Target	*Attack Type*	*Rounds*	*Data*	*Time*	*Memory*	*Ref.*
Gimli-Cipher (AEAD)	State-recovery in nonce-respecting scenario	9	4	2^{192}	2^{190}	[134]
		5	4	2^{128}	2^{126}	[134]
Gimli-Hash	Collision (Quantum)	14	-	2^{74}	negl.	[135]
	Collision	12	-	2^{106}	negl.	[135]
	SFS Collision (Quantum)	20	-	2^{74}	2^{64}	[135]
	SFS Collision	18	-	2^{106}	2^{64}	[135]
	Preimage	5	-	2^{96}	$2^{65.6}$	[133]
	2nd Preimage	3	-	1	negl.	[132]
Gimli-Permutation	Distinguishing	28*	-	2^{64}	negl.	[135]
		24 (Full)	-	2^{52}	negl.	[133]
		24 (Full)	-	2^{64}	negl.	[135]
Gimli-XOF-128	Preimage	9	-	2^{104}	2^{70}	[133]

* Represents 1-round shifted version of Gimli-permutation.

Security Analysis. Earlier versions of the Grain family have been investigated by a large number of third-party analyses (e.g., [138–143]). Although the design of Grain-128AEAD is slightly different from earlier versions, some of the third-party analyses are still applicable. The best attack was by Hao et al. [144], which presented distinguishing attacks up to 189 rounds with 2^{96} time complexity and a key-recovery attack for 190 rounds with 2^{123} time complexity. Additionally, Chang and Turan [145] analyzed the complexity of key recovery of Grain-128AEAD from the internal state under different scenarios.

Tweak plan. No tweak plan was proposed [11].

3.2.11 HyENA

HyENA (Hybrid feedback-based ENcryption with Authentication), designed by Chakraborti et al. [44], is a single-pass, inverse-free block-cipher mode of operation in which block cipher input is obtained from both the ciphertext feedback and the plaintext feedback. The candidate is instantiated with the block cipher GIFT-128 [122].

Variants. The AEAD variant of the HyENA family is listed below.

AEAD variants	*Key*	*Nonce*	*Tag*
HyENA	128	96	128

Security Analysis. Mege [146] presented a practical forgery attack due to problems in domain separation of full versus incomplete message blocks. Although designers proposed a change to address the issue before the attack was published [147], NIST considered this change as a design tweak rather than a typo correction and did not accept it during that stage of the process. The designers provided a security proof of the modified mode of HyENA [148, 149].

Tweak plan. The designers considered the following design tweaks [12]: (1) updating the masking operation, (2) simplifying the initialization to improve the hardware implementation area, and (3) removing the swap operation in the finalization.

3.2.12 ISAP

ISAP, designed by Dobraunig et al. [45], is a permutation-based AEAD scheme designed to provide, from the algorithmic level, security against a wider range of implementation attacks, such as differential fault attacks, statistical fault attacks, statistical ineffective fault attacks, and differential power analysis. The mode of ISAP is a nonce-based encrypt-then-MAC construction, where the encryption is done by XORing a message and a keystream, and the authentication/verification is based on a hash-then-MAC paradigm.

Variants. The variants of the ISAP family are listed below. The rate is represented as a tuple a, and b, where a represents the size of the rate for the nonce processing in the rekeying function IsapRk and b represents the size of the rate for all other phases. The number of rounds is represented using a 4-tuple s_H, s_B, s_E, s_K that shows the number of rounds of the permutation used during the authentication phase, the nonce processing phase, encryption and decryption phases, and generating session keys in the rekeying function phase, respectively.

AEAD variants	*Key*	*Nonce*	*Tag*	*Permutation*	*Rate*	*#Rounds*
ISAP-K-128a	128	128	128	400-bit KECCAK	144,1	16,1,8,8
ISAP-A-128a	128	128	128	320-bit ASCON	64,1	12,1,6,12
ISAP-K-128	128	128	128	400-bit KECCAK	144,1	20,12,12,12
ISAP-A-128	128	128	128	320-bit ASCON	64,1	12,12,12,12

Security Analysis. Formal security proofs of ISAP mode were provided in leakage-resilient settings [150–152]. ISAP variants are instantiated using well-analyzed permutations. The distinguishing attacks on the ASCON permutation were summarized in Table 5. A zero-sum distinguisher on the 12-round 400-bit KECCAK permutation with a complexity 2^{82} was presented in [153]. In the authentication phase, the key is used only at the end so the attacker can try to find a collision on a state for a forgery attack. However, in cases of ISAP-K-128a and ISAP-K-128, $s_H = 16$ or 20 provides a sufficient security margin because no collision attack on KECCAK variants with more than six rounds has been reported [154]. The values of s_H, s_B, s_E, and s_K were chosen by designers to provide sufficient security margins against known cube or cube-like attacks [45]. The implementation security of ISAP is summarized in [155].

Tweak plan. The designers considered changing the recommendation order as follows [13]: ISAP-A-128a (primary), ISAP-K-128a, ISAP-A-128, and ISAP-K-128. This change is motivated by the significantly better performance of the ASCON permutation on 32-bit devices and the noticeably lower area requirements of ASCON-permutation-based ISAP in hardware.

3.2.13 KNOT

KNOT, designed by Zhang et al. [46], is a permutation-based AEAD and hashing scheme. KNOT-AEAD is based on the monkeyDuplex construction [79], and KNOT-Hash is based on an extended sponge construction with a different squeezing bitrate. The structure of the KNOT permutation is similar to the 64-bit block cipher RECTANGLE [156].

Variants. The variants of the KNOT family are listed below. There are four members of the KNOT-AEAD(k, b, r) family with k-bit key, b-bit state, and r-bit rate. Similarly, there are four members of the KNOT-Hash(n, b, r, r') family with n-bit hash output, b-bit state, r-bit absorbing rate, and r'-bit squeezing rate. The number of rounds is represented using a 3-tuple a, b and c that shows the number of rounds of the KNOT permutations during initialization, AD and message processing, and finalization, respectively.

AEAD variants	*Nonce*	*Tag*	*#Rounds*	**Hash variants**	*Digest size*	*# Rounds*
KNOT-AEAD(128, 256, 64)	128	128	52, 28, 32	**KNOT-Hash(256, 256, 32, 128)**	256	68
KNOT-AEAD(128, 384, 192)	128	128	76, 28, 32	KNOT-Hash(256, 384, 128, 128)	256	80
KNOT-AEAD(192, 384, 96)	192	192	76, 40, 44	KNOT-Hash(384, 384, 48, 192)	384	104
KNOT-AEAD(256, 512, 128)	256	256	100, 52, 56	KNOT-Hash(512, 512, 64, 256)	512	140

Security Analysis. Ding et al. [157] constructed a 48-round differential trail with probability 2^{-252}, a 52-round differential trail with probability 2^{-274}, a 45-round linear trail with the square of correlation 2^{-256}, and a 51-round linear trail with the square of correlation 2^{-292}. Using the tool from [157], Zhang et al. [158] showed that the probability of the best 14-round useful differential trail during the initialization phase (the full number of rounds of the initialization is 52) is $2^{-62.2}$, the probability of the best 12-round useful differential trail during the encryption phase (the full number of rounds of the encryption is 28) is $2^{-62.2}$, and the probability of the best 13-round useful differential trail during the finalization phase (the full number of rounds of the finalization is 32) is $2^{-61.4}$.

Rohit [159] suggested that the preimage resistance may be lower than claimed. Raghav included details and asked if the interpretation was correct. The KNOT team [160] responded with several arguments and believed that the preimage attack Raghav proposed could not work. Raghav [161] conceded that the computations were not correct.

Tweak plan. The designers considered two possible tweaks [14]: (1) for security against multi-key attacks, in case of the primary member, increasing the number of rounds during the plaintext processing from 28 to 36 and the number of rounds during the finalization processing from 32 to 40 and (2) for supporting hashing on top of AEAD more seamlessly, using the same LFSRs to generate round constants of the underlying permutations.

3.2.14 LOTUS-AEAD and LOCUS-AEAD

LOTUS-AEAD and LOCUS-AEAD, designed by Chakraborti et al. [47], are AEAD modes that rely on a tweakable block cipher to allow for domain separation. These modes have nonce-based rekeying. The submission package names two variants that are instantiated with `TweGIFT-64`, a tweakable variant of `GIFT-64-128` [121]. `GIFT-64-128` has a 64-bit state and 128-bit key and 28 rounds, and a modified version – `TweGIFT-64` – is proposed that additionally has a 4-bit tweak. The round structure of `TweGIFT-64` is an augmented version of `GIFT-64-128` that adds a step for injecting bits into the state that are derived from the tweak every four rounds.

Variants. The variants of the LOTUS-AEAD and LOCUS-AEAD family are listed below. LOCUS-AEAD requires the inverse cipher for decryption, whereas LOTUS-AEAD is inverse-free.

AEAD variants	*TBC*	*# Rounds*	*Key*	*Nonce*	*Tag*
LOTUS-AEAD	`TweGIFT-64`	28	128	128	64
LOCUS-AEAD	`TweGIFT-64`	28	128	128	64

Security Analysis. `GIFT-64-128` is a widely-analyzed block cipher (e.g., see [121, 125, 127, 128, 162–165]). The submission team provided analysis of four rounds of `TweGIFT-64` that they identify as the *4-round core*, where the tweak addition occurs in the middle of the core. Both LOTUS-AEAD and LOCUS-AEAD modes claim to be INT-RUP secure. The submission team provided a security proof to support this claim that uses the ideal tweakable cipher model [166, 167].

Tweak plan. No tweak plan has been proposed.

3.2.15 mixFeed

Minimally Xored Feedback (mixFeed), designed by Chakraborty and Nandi [48], is a block-cipher based AEAD scheme. The scheme is single-pass and inverse-free, using nonce-dependent keys to limit the damage of leakage and the re-keying technique to minimize mode overhead. The submission instantiates the mode with the AES′128/128 block cipher. AES′ is largely identical to AES [94] with the exception that a MixColumns operation is performed in the final round. Therefore, all 10 rounds are structurally identical, allowing for more compact implementation than AES at the cost of performing more operations.

Variants. The single variant of the mixFeed family is listed below.

AEAD variants	*Key*	*Nonce*	*Tag*
mixFeed	128	120	128

Security Analysis. During the first round, Khairallah reported a forgery attack in the nonce-misuse scenario [168]. The authors agreed with the analysis and no longer claim nonce-misuse resistance for mixFeed. Khairallah also provided analysis of weak keys in mixFeed [98], which was extended by Leurent and Pernot [169]. The time and data complexity of these analyses does not violate the security claims. Security proofs of mixFeedare given in the ideal cipher model [170].

Tweak plan. No tweak plan has been proposed.

3.2.16 ORANGE

ORANGE (Optimum RAte spoNGE construction), designed by Chakraborty and Nandi [49], is a rate-1 sponge-variant design that uses PHOTON$_{256}$ [171], the 256-bit PHOTON permutation with 12 rounds. The AEAD scheme is called ORANGE-Zest, and the hash function is called ORANGISH. ORANGE-Zest uses a sponge variant with full state absorption by holding a dynamic secret state to mask part of the ciphertext.

Variants. The variants of the ORANGE family are listed below.

AEAD variants	*Key*	*Nonce*	*Tag*	**Hash variants**	*Digest size*
ORANGE-Zest	128	128	128	**ORANGISH**	256

Security Analysis. The PHOTON$_{256}$ permutation has received much analysis, summarized in Section 3.2.18 (see Table 8). Dobraunig et al. [172, 173] proposed a practical forgery attack that uses two encryptions of the same message block when no associated data (AD) is used. The designers of ORANGE presented a security proof of a modified mode of ORANGE [174]. Dobraunig et al. [172] stated that the attack did not seem to apply to the modified version. Sarkar et al. [175] observed that in the proof of the modified mode there is no mention of the limit of the size of the tag in the proof of [174] and showed that it is easy to forge and recover the key for the modified mode when the tag size is the same as the size of the block. Therefore, for the security of the modified mode, it is crucial to limit the size of tag so that it is infeasible to recover a state right before the tag generation.

Tweak plan. The designers have confirmed the attack presented in [173] and proposed a fix in their second-round submission of ORANGE to avoid the attack by changing the input of the function in the absence of AD. However, since NIST did not accept any tweaks during the second round of the process, this suggestion was not officially accepted by NIST. The ORANGE designers did not provide an official status update.

3.2.17 Oribatida

Oribatida, designed by Bhattacharjee et al. [50], is a permutation-based AEAD scheme based on a sponge-like structure with ciphertext-masking. The main motivation of the mode design is to provide privacy and authenticity in nonce-respecting and INT-RUP security [176]. Its underlying permutation is called SimP and is designed by borrowing two components of the Simon block cipher [96]: its key-update function and its state update function.

Variants. The variants of the Oribatida family are listed below. The number of steps for AD processing and for other parts of processing are 2 and 4, respectively. Oribatida-256-64 (primary) and Oribatida-192-96, where Oribatida-n-s has n-bit state and s-bit secret masking value. Each s-bit masking value is derived from the c-bit capacity part of its previous state, and each ciphertext block is generated by XORing a masking value and the rate part of state.

AEAD variants	*Key*	*Nonce*	*Tag*	*# Rounds of Step*	*# Steps*
Oribatida-256-64	128	128	128	34	2/4
Oribatida-192-96	128	64	96	26	2/4

Security Analysis. Rohit and Sarkar [177] reported a key recovery attack on Oribatida-192-96 that makes two online queries and has time complexity 2^{96}. Note that Oribatida-192-96 is not the primary variant. The designers provided privacy and authenticity proofs in the nonce-respecting case as well as an INT-RUP security proof [178].

Tweak plan. The designers considered two possible tweaks [15]: (1) masking the tag with a secret masking value to hide the output of the permutation in order to prevent a key recovery attack on Oribatida-192-96 and improve the nAE-security in the nonce-respecting case from 89 bits to 112 bits and (2) updating the definition of the end-of-type (EOT) control bits t_2 of the domain constants to simplify the description. The designers provided a security proof of the mode of this tweak proposal [179].

3.2.18 PHOTON-Beetle

PHOTON-Beetle, designed by Bao et al. [51], uses PHOTON$_{256}$ [171] – the 256-bit PHOTON permutation with 12 rounds – as the underlying permutation for all the members of PHOTON-Beetle. PHOTON-Beetle-AEADs are based on a sponge-like AEAD mode with a combined feedback (inspired by COFB mode [123]), and PHOTON-Beetle-Hash is based on a sponge structure.

Variants. The variants of the PHOTON-Beetle family are listed below. In the *Rate* column, a/b indicates a-bit absorbing rate and b-bit squeezing rate.

AEAD variants	*Key*	*Nonce*	*Tag*	*Rate*	**Hash variants**	*Digest size*	*Rate*
PHOTON-Beetle-AEAD[128]	128	128	128	128/128	**PHOTON-Beetle-Hash[32]**	256	32/128
PHOTON-Beetle-AEAD[32]	128	128	128	32/128			

Security Analysis. PHOTON [171] was introduced in 2011 and is a part of ISO/IEC 29192-5:2016 [180]. The PHOTON$_{256}$ permutation has received significant analysis (see Table 8). Security proofs of the Beetle mode are provided in [181–184]. Dobraunig and Mennink [185] pointed out an incorrect security bound in the submission and described a key recovery attack with empty message and AD that takes 2^{124} primitive queries. The complexity is too high for it to be a threat under the NIST security requirements, but they recommended the submitters update their security claims.

Table 8. Summary of distinguishing attacks on PHOTON$_{256}$

Method	***Rounds***	***Time***	***Memory***	***Ref.***
Zero-sum Partitions	12 /12	2^{184}	-	[186]
Statistical Integral	10/12	$2^{96.59}$	$2^{70.46}$	[187]
Rebound-like	9/12	2^{184}	2^{32}	[188]
multiple limited-birthday	8/12	$2^{10.8}$	2^{8}	[189]
Rebound-like	8/12	2^{16}	2^{8}	[171]

Tweak plan. No tweak plan has been proposed [16].

3.2.19 Pyjamask

Pyjamask, designed by Goudarzi et al. [52], is an AEAD scheme instantiated with the new block cipher Pyjamask operated in Offset Codebook (OCB) mode. OCB mode is an old, well-understood AEAD mode that can support computing many instances of the block cipher in parallel. However, the Pyjamask designers slightly altered the mode to support the use of a 96-bit block cipher.

A major design goal of the Pyjamask block ciphers is to be efficiently maskable. The cost of current masked implementations of ciphers is mostly determined by the number of AND gates that must be computed. Thus, the cipher supports a bitsliced implementation, the S-box has multiplicative complexity of one, and the key schedule is entirely linear.

Variants. The variants of the Pyjamask family are listed below.

AEAD variants	*Key*	*Nonce*	*Tag*
Pyjamask-128-AEAD	128	96	128
Pyjamask-96-AEAD	128	64	96

Security Analysis. Pyjamask's block ciphers were not published prior to becoming candidates and, thus, have seen relatively little third-party analysis. Additionally, their variant of OCB for 96-bit blocks is new and has not seen much independent analysis. Goudarzi et al. [190] gives computed bounds on differential and linear characteristics, which show a large security margin against these attacks. Dobraunig et al. [191] demonstrates higher order differential attacks on the full 96-bit version of the cipher, though the attack requires the full codebook from the cipher as well as 2^{115} work. The same paper shows an attack on the seven-round version of the cipher (out of 14 total rounds), which works even within OCB mode and requires 2^{41} chosen plaintexts and 2^{89} work.

Tweak plan. No tweak plan has been proposed.

3.2.20 Romulus

Romulus, designed by Iwata et al.[53], is an AEAD scheme based on the tweakable block cipher `SKINNY` [192]. Romulus consists of two families: a nonce-based AEAD Romulus-N and a nonce misuse-resistant AEAD Romulus-M. Romulus-N uses a rate-1 TBC-based combined feedback mode, and the mode of Romulus-M follows a MAC-then-encrypt approach.

Variants. The variants of the Romulus family are listed below.

AEAD variants	*Family*	*TBC*	*# Rounds*	*Key*	*Nonce*	*Tag*
Romulus-N1		SKINNY-128-384	56	128	128	128
Romulus-N2	Romulus-N	SKINNY-128-384	56	128	96	128
Romulus-N3		SKINNY-128-256	48	128	96	128
Romulus-M1		SKINNY-128-384	56	128	128	128
Romulus-M2	Romulus-M	SKINNY-128-384	56	128	96	128
Romulus-M3		SKINNY-128-256	48	128	96	128

Security Analysis. Proofs for privacy and authenticity of the mode of Romulus-N in the nonce-respecting scenario (with beyond-birthday-bound security) and the mode of Romulus-M in the nonce-misuse scenario were provided in [193–195]. In [196], INT-RUP security and the plaintext-awareness security for Romulus-M were provided. In [196], a new hashing mode and leakage-resilient AEAD modes based on a tweakable block cipher were described. In [197], rate-1 leakage-resilient AEAD based on the Romulus family was introduced along with security proofs in terms of CIML2, CCAmL1, and INT-RUP. Table 9 summarizes the best-known available key-recovery attacks in the single key and related-tweakey attack models. In [198, 199], new related-tweakey boomerang distinguishers on 20 rounds for Skinny-128-256 (with probability $2^{-85.77}$) and 24 rounds for Skinny-128-384 (with probability $2^{-86.09}$) were described.

Table 9. Summary of best key-recovery attacks on SKINNY-128-256 and SKINNY-128-384

Target	*Attack*	*Rounds*	*Setting*	*Data*	*Time*	*Memory*	*Ref.*
SKINNY-128-256	Impossible Diff	20/48	single key	$2^{92.1}$	$2^{245.72}$	$2^{147.1}$	[200]
	Impossible Diff	23/48	related-tweakey	$2^{124.47}$	$2^{251.47}$	2^{248}	[201]
	Impossible Diff	23/48	related-tweakey	$2^{124.41}$	$2^{243.41}$	$2^{155.41}$	[202]
	Rectangle	24/48	related-tweakey	$2^{125.21}$	$2^{209.85}$	$2^{125.54}$	[203]
SKINNY-128-384	Impossible Diff	22/56	single key	$2^{92.22}$	$2^{373.48}$	$2^{147.22}$	[200]
	MITM	22/56	single key	2^{96}	$2^{382.46}$	$2^{330.99}$	[204]
	MITM	22/56	single key	2^{96}	$2^{366.28}$	$2^{370.99}$	[205]
	Rectangle	28/56	related-tweakey	2^{122}	$2^{315.25}$	$2^{122.32}$	[165, 206]
	Rectangle	30/56	related-tweakey	$2^{125.29}$	$2^{361.68}$	$2^{125.8}$	[203]

Tweak plan. The designers considered the following tweaks [17]: (1) dropping the variants that are not based on SKINNY-128-384, (2) reducing the number of rounds of SKINNY-128-384 from 56 to 40, and (3) adding the new variant Romulus-H to provide hashing and XOR capabilities.

3.2.21 SAEAES

SAEAES, designed by Naito et al. [54], is an AEAD scheme that uses an instantiation of the SAEB (Small (Simple, Slim, Sponge-based) AEAD from Blockcipher) mode of operation [207] with the block cipher AES. SAEB is similar to the sponge-based design approach but uses a block cipher instead of a permutation. SAEB is an inverse-free, online mode of operation with the design goals of having minimum state size, being XOR-only and efficiently handling static AD.

Variants. Each member of the SAEAES family, SAEAES$k_r_1_\tau$, is parameterized by the key size k, the AD block length r_1, and the tag length τ. The nonce is fixed to 128 bits for all members. SAEAES family has the following ten members:

AEAD variants	*Key*	*Nonce*	*Tag*
SAEAES128_120_128	128	120	128
SAEAES128_120_64	128	120	64
SAEAES128_64_128	128	120	128
SAEAES128_64_64	128	120	64
SAEAES192_120_128	192	120	128
SAEAES192_64_128	192	120	128
SAEAES192_64_64	192	120	64
SAEAES256_120_128	256	120	128
SAEAES256_64_128	256	120	128
SAEAES256_64_64	256	120	64

Security Analysis. SAEAES benefits from the significant third-party analysis of AES, and a security proof of the SAEB mode is published in [207].

Tweak plan. No tweak plan has been proposed.

3.2.22 SATURNIN

SATURNIN, designed by Canteaut et al. [55], contains two AEAD variants and one hash function. The core primitive of all variants is the SATURNIN block cipher that has a 256-bit state, 256-bit key, and 9-bit parameter for domain separation. It was built with AES design principles and aims to provide security against quantum adversaries. SATURNIN-Hash is a Merkle-Damgård construction that uses the SATURNIN block cipher in Matyas-Meyer-Oseas (MMO) mode as the compression function.

Variants. The variants of the SATURNIN family are listed below. SATURNIN-CTR-Cascade is the primary AEAD variant that accepts a nonce with length up to 160 bits and a 256-bit key as inputs and produces a 256-bit tag. There is an additional variant, SATURNIN-Short, that is designed for messages less than 128 bits long with no AD. It outputs a single block of ciphertext, and verification is performed by decrypting the ciphertext and comparing the left half of the plaintext to the nonce. SATURNIN-CTR-Cascade is inverse-free, while SATURNIN-Short requires the inverse function to decrypt the ciphertext.

AEAD variants	*Key*	*Nonce*	*Tag*	**Hash variants**	*Digest size*
SATURNIN-CTR-Cascade	256	128	256	**SATURNIN-Hash**	256
SATURNIN-Short	256	128	256		

Security Analysis. Because SATURNIN shares a common structure with AES, previous analysis on AES can be applied. The SATURNIN team has provided a significant amount of analysis in the submission documentation. However, third-party cryptanalysis was not available.

Tweak plan. SATURNIN-CTR-Cascade is an encrypt-then-MAC design and, as such, does not rank as favorably as single-pass candidates in terms of throughput. To alleviate this issue, the team has proposed adding an additional variant for single-pass AEAD [18].

3.2.23 SKINNY-AEAD and SKINNY-HASH

Skinny, designed by Beierle et al. [56], is a tweakable block cipher based AEAD and hashing scheme. The main component of SKINNY-AEAD and SKINNY-HASH is the tweakable block cipher SKINNY. This tweakable block cipher was proposed in CRYPTO 2016 [192]. The design goal of SKINNY is to compete with the block cipher SIMON in terms of hardware and software performance while proving stronger security guarantees with regard to differential and linear attacks. SKINNY-AEAD uses a mode of operation following the general θCB3 framework [208], whereas SKINNY-HASH follows the sponge construction.

Variants. The variants of the SKINNY family are listed below. M5 and M6 are additional variants that do not follow NIST requirements. The table also includes the block size of the underlying function b, the rate size r, and the capacity size c.

AEAD variants	*TBC*	*Key*	*Nonce*	*Tag*	**Hash variants**	*TBC*	*Digest size*	b,r,c
SKINNY-AEAD-M1	SKINNY-128-384	128	128	128	SKINNY-**tk3-Hash**	SKINNY-128-384	256	384,128,256
SKINNY-AEAD-M2	SKINNY-128-384	128	96	128				
SKINNY-AEAD-M3	SKINNY-128-384	128	128	64				
SKINNY-AEAD-M4	SKINNY-128-384	128	96	64				
SKINNY-AEAD-M5	SKINNY-128-256	128	96	128	SKINNY-tk2-Hash	SKINNY-128-256	256	256,32,224
SKINNY-AEAD-M6	SKINNY-128-256	128	96	64				

Security Analysis. There has been a significant amount of third-party analysis on SKINNY (e.g., [201, 202, 209]) (see Table 9 for the key-recovery and distinguishing attacks). Zhao et al. [206] presented a 28-round (out of 56) related-tweakey rectangle attack on SKINNY-128-384 with time complexity $2^{315.25}$, data of 2^{122} chosen plaintexts, and memory $2^{122.32}$. Hadipour et al. [203] presented a 30-round attack on SKINNY-128-384 with time complexity $2^{361.68}$, data of $2^{125.29}$ chosen plaintexts, and memory $2^{125.8}$. Since most of the third-party analyses require high complexity and only work in the related-tweakey model, according to [210], the security margin of the candidate is adequate.

Tweak plan. The designers considered two tweaks [19]: (1) proposing five new variants – SKINNY-AEAD-M1+, SKINNY-AEAD-M2+, SKINNY-AEAD-M3+, and SKINNY-AEAD-M4+, SKINNY-tk3-Hash+ – where the number of rounds of SKINNY-128-384 is reduced from 56 to 40 and (2) dropping the variants based on SKINNY-128-256, namely SKINNY-AEAD-M5, SKINNY-AEAD-M6, and SKINNY-tk2-Hash.

3.2.24 SPARKLE

SPARKLE, designed by Beierle et al. [57], comprises the SCHWAEMM family of AEAD cipher and ESCH family of hash functions, both based on the SPARKLE permutation [57]. The permutation applies multiple distinct instances of Alzette, a 4-round 64-bit block cipher, to achieve nonlinearity. Alzette is a 64-bit S-box based on an Addition-Rotation-XOR (ARX) design operating on 32-bit words, making it particularly efficient in software. The SCHWAEMM family of AEAD cipher is based on the duplexed sponge construction with a combined feedback. The ESCH family of hash functions is based on the sponge construction.

Variants. The variants of the SPARKLE family are listed in the table below. Both primary algorithms rely on the 384-bit SPARKLE permutation. In the table, each variant uses the b-bit SPARKLE permutation with r-bit rate and c-bit capacity, where $b = r + c$. In the # *Steps* column for AEAD variants, x, y indicates that the SPARKLE permutation with y steps is used (1) in the initialization, (2) between the AD processing and the message processing, and (3) in the finalization; and the SPARKLE permutation with x steps is used in (1) the AD processing and (2) the message processing. In the # *Steps* column for Hash variants, x, y indicates that the SPARKLE permutation with y steps is used once to generate the first half of the hash output, and the SPARKLE permutation with x steps is used (1) in the absorption phase and (2) to generate the second half of the hash output.

AEAD variants	*Key*	*Nonce*	*Tag*	*b, r, c*	*# Steps*	**Hash variants**	*Digest size*	*b, r, c*	*# Steps*
SCHWAEMM128-128	128	128	128	256,128,128	7,10	**ESCH256**	256	384,128,256	7,11
SCHWAEMM192-192	192	192	192	384,192,192	7,11	ESCH384	384	512,128,384	8,12
SCHWAEMM256-128	128	256	128	384,256,128	7,11				
SCHWAEMM256-256	256	256	256	512,256,256	8,12				

Security Analysis. The security of SPARKLE against various attacks was analyzed by Beierle et al. [211]. Work has also shown that the probability of 7-round differential trails of Alzette is at most 2^{-24} [212]. Liu et al. [213] presented 4-round differential-linear and rotational differential-linear trails in Alzette that improved on previously published differential characteristics. In [20], the designers of SPARKLE reported that the probability of the best 7-round differential trails is 2^{-26}.

Tweak plan. The designers have indicated that they do not plan to tweak their submission [20].

3.2.25 SPIX

SPIX, designed by AlTawy et al. [58], is a permutation-based AEAD scheme based on the monkeyDuplex construction [79] with additional key additions during initialization and finalization to prevent a key-recovery attack and a forgery attack even after a state is recovered by an attacker during the encryption. SPIX operates on two instances (where steps $s \in \{9, 18\}$) of 256-bit sLiSCP-light permutations. The nonlinear layer applies a substitution box to two words of the state. The substitution-box function is an unkeyed 8-round variant of the Simeck-64 block cipher [73] that is denoted SB-64.

Variants. The single AEAD variant of the SPIX family is listed below. In the table, the number of steps corresponds to 18 steps during the initialization and finalization phases and 9 steps during the AD or message processing phases.

AEAD variants	*Key*	*Nonce*	*Tag*	*# Steps*
SPIX	128	128	128	18, 9

Security Analysis. SPIX and one of the SpoC variants use similar sLiSCP-light-[256] permutations. Differential characteristics [214–216] and distinguishers [217] on sLiSCP-light[256], summarized in Section 3.2.26, are also applicable to SPIX.

Tweak plan. No tweak was planned [21].

3.2.26 SpoC

Sponge with masked Capacity (SpoC), designed by AlTawy et al. [59], is a permutation-based AEAD scheme that aims to achieve a higher rate than the duplex sponge mode while also maintaining the same level of security. This is done by masking the capacity with data. The SpoC mode is instantiated with sLiSCP-light permutations similar to those of SPIX.

Variants. The variants of the SpoC family are listed below. SpoC-64, the primary variant, has 64-bit rate, 128-bit capacity, and 64-bit tag and uses an unkeyed 6-round variant of the

Simeck-64 block cipher [73]. A second variant, SpoC-128, has a 128-bit rate, 128-bit capacity, and 128-bit tag and uses an unkeyed 8-round variant of the Simeck-64 block cipher [73]. Both variants use a 128-bit nonce and 128-bit keys. Both permutations comprise 18 steps. In the table, b is the block size of the underlying function, r is the rate size, and c is the capacity size.

AEAD variants	*Key*	*Nonce*	*Tag*	*# Rounds*	*# Steps*	b, r, c
SpoC-128_sLiSCP-light-[256]	128	128	128	8	18	256, 128, 128
SpoC-64_sLiSCP-light-[192]	128	128	64	6	18	192, 64, 128

Security Analysis. Kraleva et al. [214–216] described differential characteristics of sLiSCP-light-[256] and sLiSCP-light-[192] that covered six, seven, or nine steps, as well as corresponding attacks on reduced-step SpoC-64 and SpoC-128 variants. A key recovery attack on full SpoC-64 was presented that does not violate the security claims due to large data requirements.

Hosoyamada et al. [217] described limited-birthday distinguishers over 15 steps of sLiSCP-light that can be used to mount attacks on 15-step and 16-step variants of the permutations. These attacks on the permutation do not pose an immediate threat to the SpoC submission, as they do not take the mode into account. The proofs of the mode of SpoC were provided in [182–184].

Tweak plan. The SpoC team did not plan to tweak their design [22].

3.2.27 Spook

Spook, designed by Bellizia et al. [60], is an AEAD based on the duplex mode of operation [80], where the Shadow-512 permutation or the Shadow-384 permutation is used with strengthened initialization and finalization phases using the tweakable block cipher Clyde-128.

Variants. The variants of the Spook family are listed below, where su and mu mean single-user and multi-user. For example, Spook[128, 512, su] uses Clyde-128 as its underlying tweakable block cipher and Shadow-512 permutation as its underlying permutation in the single user setting.

AEAD variants	User setting	*Key*	*Nonce*	*Tag*
Spook[128, 512, su]	single	128	128	128
Spook[128, 384, su]	single	128	128	128
Spook[128, 384, mu]	multi	256	128	128
Spook[128, 512, mu]	multi	256	128	128

Security Analysis. Derbez et al. [218] described practical distinguishers of the full six steps of Shadow-512 and Shadow-384 and also practical forgeries against Spook with 4-step Shadow in the nonce-misuse scenario.

Tweak plan. The designers considered two possible tweaks [23, 219] for improving the security margins of Spook against the collision attack [218] of Crypto 2020 against a reduced-round Shadow: (1) changing the diffusion layer with an efficient MDS matrix and (2) updating the round constants.

3.2.28 Subterranean 2.0

Subterranean 2.0, designed by Daemen et al. [61], is an AEAD and hashing scheme operating on a 257-bit state and is based on the monkeyDuplex construction [79].

Variants. The Subterranean 2.0 submission contains one AEAD variant, Subterranean-SAE, that applies Session Authenticated Encryption (SAE) to a Subterranean duplex object. The submission also specifies a single hash variant, Subterranean-XOF, where the XOF output length is fixed to 256 bits. Both Subterranean-SAE and Subterranean-XOF have an injection rate of 33 bits, extraction rate of 32 bits, and 224-bit capacity. In addition, the specification defines a Doubly-Extendable Cryptographic Keyed (deck) function, which can be used to derive fresh session keys.The variants of the Subterranean 2.0 family are listed below.

AEAD variants	*Key*	*Nonce*	*Tag*	**Hash variants**	*Digest size*
Subterranean-SAE	128	128	128	**Subterranean-XOF**	256

Security Analysis. Liu et al. [220] performed a security analysis of Subterranean-SAE using four types of conditional cube testers inspired by the idea of conditional cube attacks [221]. They presented a full-state-recovery with 2^{15} bytes of data in the nonce-misuse scenario and a key-recovery with time 2^{35} in the nonce-misuse scenario, distinguishing with 2^{35} bytes of data and time 2^{33} with 4 blank rounds in nonce-respecting scenario and key-recovery attack (data $2^{71.5}$ bytes of data and time 2^{122}) with 4 blank rounds in nonce-respecting scenario. Note that Subterranean 2.0 uses eight blank rounds. Later, Song et al. [222] proposed a better (in terms of time complexity and the number of nonce repetitions) full-state-recovery attack with 352 bytes of data in nonce-misuse by proposing a nested, one-round differential analysis, which exploits the output difference in two consecutive rounds. Song et al. [222] made MILP models and showed that in cases of Subterranean-deck and Subterranean-SAE, the range of the correlation of linear trails of three or four keystream blocks is $[2^{-96}, 2^{-49})$, and the range of the probability of differential trails for state collision is $[2^{-180}, 2^{-108})$.

Tweak plan. No tweak plan was proposed [24].

3.2.29 SUNDAE-GIFT

SUNDAE-GIFT, designed by Banik et al.[62], is a block-cipher based AEAD scheme instantiated with the block cipher GIFT-128 [121, 122]. Its mode follows the MAC-then-encrypt approach to provide privacy and authenticity even in the nonce-misuse setting. Encryption (based on Output Feedback (OFB) mode) is done by XORing a message and a key-stream; authentication/verification is based on a variant of CBC-MAC (inspired by GCBC [223]), which uses multiplication by 2 or 4 to provide domain separation for padded vs. unpadded versions of the AD and plaintext.

Variants. The variants of the SUNDAE-GIFT family are listed below.

AEAD variants	*Key*	*Nonce*	*Tag*
SUNDAE-GIFT-96	128	96	128
SUNDAE-GIFT-0	128	0	128
SUNDAE-GIFT-128	128	128	128
SUNDAE-GIFT-64	128	64	128

Security Analysis. Mege [224] reported an attack that reveals a block of plaintext with 2^{46} ciphertext blocks, 2^{49} bytes of memory, and 2^{45} memory lookups with attack success probability 2^{-39}. The attack does this by finding internal collisions. Peyrin [225] responded that this is a generic birthday attack, and the complexity supports the security claims made in the submission. Chang et al. [226] reported that SUNDAE is not INT-RUP-secure. The third-party analyses on GIFT-128 are summarized in Table 6 in Section 3.2.8.

Tweak plan. The designers did not plan to tweak their submission [25].

3.2.30 TinyJAMBU

TinyJAMBU, designed by Wu and Huang [63], is an AEAD scheme that is inspired by the third-round candidate of the CAESAR competition, JAMBU [227]. The main component of TinyJAMBU is a 128-bit keyed permutation without a key schedule that is based on a nonlinear feedback shift register, and the nonlinearity in each round is obtained using a single `NAND` operation.

Variants. The variants of the TinyJAMBU family are listed below.

AEAD variants	*Key*	*Nonce*	*Tag*	*State size*
TinyJAMBU-128	128	96	64	128
TinyJAMBU-192	192	96	64	128
TinyJAMBU-256	256	96	64	128

Security Analysis. In [63], the designers provided a security proof for the mode, analysis of the keyed-permutation, and security against forgery and key recovery attacks (including differential, linear, algebraic, and slide attacks). Saha et al. [228–230] showed that the security margin of TinyJAMBU is around 12 % due to the dependencies between the outputs of multiple AND gates.

Tweak plan. In order to provide a larger security margin, the designers considered increasing the number of rounds for the permutation that processes the nonce and the AD [26].

3.2.31 WAGE

WAGE, designed by Aagaard et al. [64, 231], is a permutation-based AEAD scheme. WAGE is based on the duplex mode of operation [80], where the rate is 64 bits and the capacity is 195 bits. The mode includes additional key additions during the initialization and finalization phases to prevent a key-recovery attack and a forgery attack even after a state is recovered by an attacker during the encryption. The design of the underlying 259-bit permutation of WAGE, called the WAGE permutation, is inspired by the initialization phase of the Welch-Gong (WG) cipher [232, 233]. The WAGE permutation has 111 rounds, where the state is described by 37 7-bit stages. At each round, six stages out of the 37 stages are updated and shifted with one stage through an LFSR operation with 10 tap positions; two applications of the Welch-Gong permutation (WGP); four applications of a 7-bit S-box (SB); and the addition of two 7-bit round constants. Note that WGP and SB are chosen for low-cost hardware implementation, the differential probabilities of WGP and SB are $2^{-4.42}$ and 2^{-4}, and the corresponding linear squared correlations are $2^{-5.08}$ and $2^{-5.35}$, respectively.

Variants. The single AEAD variant of the WAGE is listed below.

AEAD variants	*Key*	*Nonce*	*Tag*
WAGE-$\mathcal{AE}$-128	128	128	128

Security Analysis. Using the MILP model, the WAGE designers [64, 231] showed that the lower bounds on the minimum number of differentially/linearly active S-boxes are 59 for the case where there are no constraints on the positions of input and output differences and 72 for the case where the input and output differences restricted to only rate positions.

Tweak plan. The designers considered one possible tweak [27]: changing the number of rounds of the WAGE permutation from 111 to 74 in the initialization, associated data processing, and finalization phases.

3.2.32 Xoodyak

Xoodyak, designed by Daemen et al. [65], is a permutation-based AEAD and hashing scheme. Xoodyak is built from a fixed 384-bit permutation Xoodoo operated in Cyclist mode. The design approach of Xoodoo is closely related to that of the KECCAK permutation.

Variants. The variants of the Xoodyak family are listed below.

AEAD variants	*Key*	*Nonce*	*Tag*	**Hash variants**	*Digest size*
Xoodyakv1	128	128	128	**Xoodyakv1**	256

Security Analysis. The designers of Xoodoo provide bounds on differential and linear trails, as well as a preliminary security analysis on the permutation. Song and Guo [234] demonstrated a cube-like attack on Xoodoo-AE reduced to six (out of 12) rounds in 2^{89} time and 2^{55} memory. Zhou et al. [235] also presented a conditional cube attack on Xoodyak reduced to six (out of 12) rounds in a nonce-misuse setting, recovering the 128-bit key in $2^{43.8}$ time and negligible memory cost. Liu et al. [236] showed a zero-sum distinguisher on the full 12-round Xoodoo, with 2^{33} time complexity and Liu et al. [213] identified a 4-round rotational differential-linear distinguisher with correlation one on Xoodoo with a probability $2^{-117.81}$.

Tweak plan. The submitters have indicated that they plan to make a tweak that will improve speed for short messages [28].

3.3 Additional Considerations

This section describes additional considerations for the selection of the finalists.

3.3.1 Side-Channel Resistance

As stated in the submission call [31], resistance to side-channel attacks is a desired feature. This section provides a non-exhaustive overview of the results on the side-channel resistance of the candidates.

Belaïd et al. [237] introduced the tool `Tornado` that produces bitsliced masked implementations from a high-level description of a cryptographic primitive, and they evaluated 11 of the candidates that self-identified as being amendable to masking across a large range of masking orders.

Abdulgadir et al. [238] developed FOBOS2 – an improved version of FOBOS (Flexible Opensource workBench fOr Side-channel analysis) – to perform power measurements and side-channel resistance evaluations of the candidates. Results on SpoC, Spook, GIFT-COFB, ASCON, and AES-GCM are measured.

Coleman et al. [239] compared the side-channel protection cost of the ARX-based candidates COMET-CHAM and SCHWAEMM. Their results showed that, on average, the cost of protecting these ciphers against side-channel attacks is 32 % more in area and 38 % more in power compared to other candidates.

Guo et al. [240] proposed security definitions and constructions of authenticated encryption schemes in the presence of side-channel leakages and nonce misuse.

Bellizia et al. [241] considered the nonce misuse-resilient CCA security with a unique challenge nonce (called CCAm security) and the nonce misuse-resistant Ciphertext Integrity (called CIM security) in the two leakage models, L1 (encryption leakage only) and L2 (both encryption and decryption leakages). As shown in the below table, Bellizia et al. [241] considered the achievable securities of leveled implementations of some second-round candidates, where different parts of the investigated candidates have different security requirements against leakage.

Achievable security	***Candidates***
CCAL1, CIL1	DryGASCON, Gimli, Oribatida, PHOTON-Beetle
CCAmL1, CIML2	ACE, ASCON, SPIX, Spook, WAGE
CCAmL2, CIML2	ISAP

Mandal and Gong [242] investigated the Boolean circuit complexity of the core primitives of the candidates, specifically for privacy enhancing cryptographic techniques. The individual nonlinear counts of the candidates are also useful to study the cost of providing side-channel resistance for the candidates.

Adomnicai and Peyrin [243] studied the benefits of the fixslicing implementation strategy that aims to achieve constant-time software implementations, and showed how it impacted the software performance of GIFT-COFB, Romulus, SKINNY-AEAD, and SAEAES.

Kiaei et al. [244] examined execution time characteristics of several second-round candidates, and how these characteristics enable timing side channels. Additionally, the authors evaluated the feasibility of automatic constant-time code generation on ACE, ASCON, GIFT-COFB, and WAGE.

3.3.2 Nonce-Misuse Security

In the nonce-misuse scenario, the same nonce is used for two or more encryption queries. None of the rate-1 second-round candidates are integrity-secure in a nonce-misuse scenario [245], because a simple forgery attack is possible by finding a state collision. There are eight rate-1 candidate primary variants: COMET, GIFT-COFB, HyENA, mixFeed, ORANGE, Pyjamask, Romulus, and SKINNY-AEAD. The candidates ESTATE and SUNDAE-GIFT are both based on the MAC-then-encrypt approach and are designed to provide confidentiality and integrity in nonce-misuse scenarios.

The nonce-misuse security of designs following sponge-type construction, whether it is based on a public permutation or based on a block cipher with a fixed secret key or a keyed permutation, largely depends on the capacity size c.

In the case of an AEAD with a short tag size t, even when a nonce repetition is not allowed for its encryption queries, a nonce-misuse attack can be mounted through a successful decryption query with probability 2^{-t} [246]. For example, when $t = 64$, a forgery attack on any rate-1 candidate with decryption-query complexity 2^{64} is possible. However, a forgery attack requiring more than data complexity 2^{50} is beyond the NIST submission requirements [31].

Ashur et al. [247] introduced a concept of nonce misuse-resilient CCA security, which means that as long as the nonce used in the test query is fresh, the confidentiality of the test query must hold. Later, Bellizia et al. [241] and Guo et al. [240] called the nonce misuse-resilient CCA security "CCAm." Note that nonce misuse-resilience considers security with a unique challenge nonce, and nonce misuse-resistance considers the security without restriction on nonce-repetition. For example, Subterranean-SAE is insecure in terms of CCAm security [220]. This is because an efficient key-recovery attack on Subterranean-SAE is possible in a nonce-misuse scenario. Note that once a key is compromised, no meaningful security is guaranteed.

3.3.3 RUP Security

In the releasing unverified plaintext (RUP) scenario, even when tags are not valid, unverified plaintexts are returned through decryption queries [176]. When an AEAD provides integrity security in the RUP scenario, the AEAD is called INT-RUP-secure. None of the rate-1 second-round candidates are INT-RUP-secure [245]. SUNDAE-GIFT is not INT-RUP-secure [226], which shows that its nonce-misuse security does not guarantee its RUP security. ESTATE [113], LOTUS-AEAD [167], LOCUS-AEAD [167], and Oribatida [178] are designed to provide integrity in the RUP scenario.

In the case of a sponge-type AEAD design based on a permutation, except the exhaustive key search and a tag guessing, its INT-RUP security level from its mode level largely depends on the capacity size c, where the capacity part of each state is secret and is assumed to be unpredictable.

3.3.4 Impacts of State Recovery

In this section, we analyze the complexity of recovering the key from the knowledge of the internal state of the cipher. Though the secret state recovery is a strong assumption, some second-round candidates claim an additional feature that ensures limited damage even in case of state recovery.

ACE, ASCON, ISAP, SPIX, Spook, and WAGE are all based on sponge-type modes with keyed initialization and finalization phases making it difficult to recover a key from the knowledge of the secret state. Therefore, they are designed to provide security against key-recovery and forgery attacks even when an entire internal state during the encryption phase is disclosed to the attackers.

In cases of candidates based on a (tweakable) block cipher with key(s) or a keyed permutation, no meaningful security is guaranteed, because the disclosure of an entire internal state during the encryption phase may imply the disclosure of secret key as well.

In cases of sponge-like candidates based on a public permutation with no independent keyed finalization, once an entire internal state during the encryption phase is disclosed a key or an initial secret state can be recovered by reversing it, which enables attackers to freely perform encryption and decryption with any input. There are 10 such candidates: DryGASCON, Gimli, KNOT, ORANGE, Oribatida, PHOTON-Beetle, SPARKLE, SpoC, Subterranean 2.0 [220], and Xoodyak.

Chang and Turan [145] analyzed the complexity of key recovery of Grain-128AEAD from the internal state under different scenarios and showed that the secrecy of the key is not fully protected by reintroducing the key.

3.3.5 Post-Quantum Security

Although not formally required by the lightweight cryptography standardization process, an additional concern for new cryptography standards is their resistance to quantum threats. Shor [248] showed that a sufficiently powerful quantum computer can efficiently break commonly used public-key cryptosystems based on integer factorization, discrete logarithm, or elliptic-curve discrete logarithm problems. To address this issue, NIST is currently running a process to standardize one or more quantum-resistant public-key cryptographic algorithms [249].

Most symmetric cryptosystems are considered to be relatively secure against quantum threats. The best generic attack against symmetric ciphers is Grover's algorithm [250], which provides a quadratic speedup for exhaustive key search (or finding collisions in hash functions). To avoid the attack, variants with larger key sizes (or larger digest sizes) may be preferred. However, due to the requirement to run Grover's algorithm using sequential queries, the practical implications of the attack may be limited. Later studies have shown that it is also possible to exploit the internal structure of symmetric ciphers (e.g., Even-Mansour construction [251], 3-round Feistel cipher [252], mode of operations [253]) using superposition queries to a quantum cryptographic oracle. Hence, classical security proofs of cryptographic construction need to be revisited to consider the quantum threat. There are also new studies that extend these results to a quantum attacker limited to classical queries and offline quantum computations (e.g., [107, 254]), which can be more relevant in practice. Recently, in [108], the securities of sESTATE and Elephant against key recovery attacks were analyzed in both online and offline quantum settings.

Three of the second-round teams responded to the quantum security of their candidates in their submission document:

- ASCON [35] includes the variant Ascon-80pq specifically designed for quantum resistance supporting a key size of 160 bits;

- Gimli [42] argues that the selection of the 256-bit key is done to avoid quantum and multi-user attacks;
- SATURNIN [55] was designed specifically to resist quantum attacks with a 256-bit key and block size.

Additionally, there are six more candidates, namely DryGASCON, KNOT, SAEAES, SPARKLE, Spook and TinyJAMBU, with variants that support 256-bit keys.

4. Performance Benchmarking

This section summarizes the main software and hardware performance benchmarking initiatives that were considered during evaluation. While these efforts provided crucial information on the performance of the candidates, it is important to note their limitations. Results may not present a complete picture of a candidate's potential for optimization in any particular metric. Further, not all implementations are designed with the same assumptions or goals, and there are more diverse implementations for some candidates than others. More efficient implementations are likely possible for all candidates. As such, the results of these efforts were considered as a general guide and not a strict ranking.

4.1 Software Benchmarking

Software performance on microcontrollers is an important criterion for the evaluation of the candidates. Multiple benchmarking initiatives evaluated the performance of the candidates. These initiatives cover a wide range of target platforms from 8-bit microcontrollers with limited memory to 32-bit and 64-bit microcontrollers. The specifications of the microcontrollers used by the benchmarking initiatives are summarized in Table 10.

4.1.1 Microcontroller Benchmarking by NIST

NIST [255] internally evaluated the performance of the candidates on microcontrollers and compared them against the NIST standards AES-GCM and SHA-256. The implementations were collected from the submission packages, GitHub repositories of the candidates, and the repositories of other benchmarking initiatives. In total, 327 AEAD and 116 hash implementations were benchmarked on two 8-bit MCUs and four 32-bit MCUs. The performance metrics captured were code size and execution time, with various input lengths ranging from 8 bytes to 1024 bytes. The implementations were benchmarked under four different compiler optimization flags. Appendix A contains details and selected results for this effort. The complete set of results is provided on the project GitHub page [255].

The AEAD variants of KNOT and PHOTON-Beetle achieved the smallest code size on 8-bit platforms. On one of the 8-bit platforms (ATmega4809), the code sizes of GIFT-COFB and TinyJAMBU are also less than that of AES-GCM. ASCON, GIFT-COFB, Gimli, KNOT, and TinyJAMBU are smaller than AES-GCM on all 32-bit platforms, whereas Elephant, Pyjamask, SATURNIN, SPARKLE, SPIX, SUNDAE-GIFT, and Xoodyak have smaller

Table 10. Specifications of microcontrollers used in benchmarking initiatives

Initiative	***Microcontroller***	***Processor***	***Word size***	***Clock speed***	***Flash***	***RAM***
NIST [255]	ATmega328P	AVR	8-bit	16 MHz	32 KB	2 KB
	ATmega4809	AVR	8-bit	20 MHz	48 KB	6 KB
	SAM D21	ARM Cortex-M0+	32-bit	48 MHz	256 KB	32 KB
	nRF52840	ARM Cortex-M4	32-bit	64 MHz	1 MB	256 KB
	PIC32MX340F512H	MIPS32 M4K	32-bit	80 MHz	512 KB	32 KB
	ESP8266	Tensilica L106	32-bit	80 MHz	4 MB	80 KB
Renner et al. [256]	ATmega328P	AVR	8-bit	16 MHz	32 KB	2 KB
	STM32F103	ARM Cortex-M3	32-bit	72 MHz	64 KB	20 KB
	STM32F746ZG	ARM Cortex-M7	32-bit	216 MHz	1 MB	320 KB
	ESP32 WROOM	Tensilica Xtensa LX6	32-bit	240 MHz	4 MB	520 KB
	Kendryte K210	RISC-V (Dual Core)	64-bit	400 MHz	16 MB	8 MB
Weatherley [257]	ATmega2560	AVR	8-bit	16 MHz	256 KB	8 KB
	AT91SAM3X8E	ARM Cortex-M3	32-bit	84 MHz	512 KB	96 KB
	ESP32	Tensilica Xtensa LX6	32-bit	240 MHz	4 MB	520 KB

code sizes on some of those platforms. Overall, Gimli, KNOT, and TinyJAMBU have the smallest code size across all platforms with ASCON and Xoodyak performing particularly well on 32-bit platforms.

The hash variants of Gimli, KNOT, and SPARKLE achieve smaller code sizes than SHA-256 on all the tested platforms. ASCON outperforms SHA-256 except on SAM D21 and ESP8266, and Xoodyak outperforms SHA-256 except on nRF52840. PHOTON-Beetle achieves the smallest code size on 8-bit platforms while performing worse on 32-bit platforms.

4.1.2 Microcontroller Benchmarking by Renner et al.

Renner et al. [256, 258–260] developed a benchmarking framework to evaluate the performance of AEAD algorithms and evaluated the second-round candidates on five microcontrollers. The benchmarking provides execution time (average time to generate the test vectors), size of the compiled binary, and RAM usage (for only STM32F746ZG) as performance metrics. The results are published at [258].

Considering only the primary variants, the results can be summarized as follows:

- SCHWAEMM, TinyJAMBU, GIFT-COFB, KNOT, and HyENA were the fastest on the Arduino UNO, while KNOT, PHOTON-Beetle, TinyJAMBU, Xoodyak, and GIFT-COFB had the smallest code sizes.
- On a STM32F1 MCU, Xoodyak, ASCON, SCHWAEMM, TinyJAMBU, and GIFT-COFB had the fastest implementations; and KNOT, TinyJAMBU, Xoodyak, ACE,

and SCHWAEMM used the least ROM.

- The fastest five on the ESP32 MCU were TinyJAMBU, Xoodyak, SAEAES, KNOT, and GIFT-COFB, and the smallest were Grain-128AEAD, KNOT, TinyJAMBU, ASCON, and SCHWAEMM.

- TinyJAMBU, ASCON, Xoodyak, SCHWAEMM, and Gimli were fastest on the STM32F7 MCU; ASCON, KNOT, TinyJAMBU, Xoodyak, and SCHWAEMM used the least ROM; and TinyJAMBU, KNOT, ASCON, SCHWAEMM, and SpoC used the least RAM.

- ASCON, KNOT, SAEAES, TinyJAMBU, and Xoodyak were the fastest on the RISC-V platform; and ASCON, KNOT, TinyJAMBU, SCHWAEMM, and Xoodyak used the least ROM.

- Overall, ASCON, GIFT-COFB, KNOT, SCHWAEMM, TinyJAMBU, and Xoodyak stood out as top performers.

4.1.3 Microcontroller Benchmarking by Weatherley

Weatherley [257] developed optimized implementations of the second-round candidates and performed timing measurements on them. For each variant, an optimized C implementation and assembly implementations for AVR and ARM-v7m architectures were provided. AES-based variants of four submissions, namely COMET, ESTATE, mixFeed, and SAEAES were excluded. The benchmarks included timing results for encrypting and decrypting 16- and 128- byte messages. The speed metrics were given by comparing each AEAD algorithm with ChaChaPoly [261–263] and each hashing algorithm with BLAKE2s [264]. This work also provided the masked implementations of ASCON, GIFT-COFB, Gimli, KNOT, Pyjamask, SPIX, SpoC, Spook, TinyJAMBU, and Xoodyak and benchmarked their execution times for up to six shares.

Results can be summarized as follows:

- On the 8-bit AVR platform, COMET, SPARKLE, GIFT-COFB, HyENA, ASCON, TinyJAMBU, ORANGE, Spook, and Gimli were the top AEAD performers, and their performance is more than twice that of ChaChaPoly.

- ESTATE, SUNDAE-GIFT, Subterranean 2.0, Xoodyak, Pyjamask, Romulus, PHOTON-Beetle, KNOT, SKINNY-AEAD, SpoC, and SATURNIN are faster than ChaChaPoly by a factor less than two.

- Oribatida, Grain-128AEAD, WAGE, ForkAE, ACE, Elephant, and ISAP are slower than ChaChaPoly. SPARKLE, Gimli, SATURNIN, and Xoodyak performed hashing faster than BLAKE2s.

- DryGASCON, ASCON, ORANGE, PHOTON-Beetle, Subterranean 2.0, ACE, KNOT, and SKINNY-HASH were slower than BLAKE2s.

- On the 32-bit platforms, COMET, SPARKLE, Xoodyak, ASCON, and TinyJAMBU were faster than ChaChaPoly.
- There was no hash algorithm among the candidates that was faster than BLAKE2s on 32-bit platforms.

4.1.4 Additional Results

Campos et al. [265] compared the effectiveness of various optimization methods on six of the second-round candidates when implemented on RISC-V platforms. The focus of this work was not to compare the performance between candidates, but, rather, to compare different implementation optimization strategies of a small number of candidates. This was done by considering the time (in cycle counts) for encrypting 128 bytes of message with 128 bytes of AD, as well as hashing 128 bytes of messages.

Nisanci et al. [266] evaluated the size and speed of the reference implementations of the primary AEAD variants of the second-round candidates on 32-bit RISC-V architectures. The metrics used were code size and execution time for authenticated encryption/decryption of inputs of size 128-byte, 2KB, and 16KB. In all cases, compiling with the `-os` flag resulted in a significantly smaller executable than compiling with no optimization. Other flags also led to more compact implementations in some cases.

eBACS (ECRYPT Benchmarking of Cryptographic Systems) [267] is a repository of software performance benchmarking results on conventional processors. There are several smaller benchmarking projects, each focusing on a different type of primitive or functionality. Results from eBAEAD (ECRYPT Benchmarking of Authenticated Ciphers) and eBASH (ECRYPT Benchmarking of All Submitted Hashes) results were considered for server or hub devices. ASCON, Gimli, and Xoodyak implementations performed best overall for hashing, followed by SATURNIN and SPARKLE. For AEAD, ASCON and Xoodyak consistently outperform other candidates on 64-bit platforms when considering all tested input lengths. Gimli, KNOT, and SAEAES also have high speed implementations on some systems.

Santos and Großschädl [268] studied the execution time and code size of assembler implementations of permutations of ASCON, Gimli, SPARKLE, and Xoodyak on an 8-bit AVR, and a 32-bit ARM Cortex-M3 microcontroller. According to the study, the throughput of ASCON, SPARKLE and Xoodoo permutations is very similar on ARM; and on 8-bit AVR, SPARKLE outperforms ASCON and Xoodoo.

Fotovvat et al. [269] studied the performance of the candidates on Raspberry Pi 3, Raspberry Pi Zero W, and iMX233 using the power consumption, random access memory usage, and execution time metrics.

4.2 Hardware Benchmarking

In addition to hardware figures provided by the submitters, NIST considered results from the hardware benchmarking initiatives described in subsections 4.2.1 to 4.2.3 below. The platforms and metrics used in these efforts are summarized in Table 11.

4.2.1 FPGA Benchmarking by GMU CERG

The George Mason University Cryptographic Engineering Research Group [270] provided performance results using their tool suite: ATHENa [271], Minerva [272], and Xeda [273]. This effort evaluated the implementations of 27 second-round candidates and ranked them in terms of throughput and energy per bit. Optimization rules fixed an upper limit on implementation size and selected the clock frequency for each implementation that maximized its throughput. The same clock frequency was used for all implementations during power and energy analysis.

On the Xilinx Artix-7 platform, Subterranean 2.0, ASCON, Xoodyak, Gimli, KNOT, GIFT-COFB, DryGASCON, COMET, Spook, Elephant, TinyJAMBU, and Romulus were the top AEAD performers according to throughput. TinyJAMBU, Gimli, Subterranean 2.0, Romulus, and ESTATE had very compact implementations that were under 1000 Look Up Tables (LUTs). Gimli, Xoodyak, and ASCON performed hashing faster than SHA-2. Subterranean 2.0, Gimli, ACE, Xoodyak, and SATURNIN were the five candidates with the smallest implementations supporting hashing. SPARKLE had the only implementation that exceeded the resource limit.

Results were similar for the Intel Cyclone 10 LP FPGA. Subterranean 2.0, ASCON, Gimli, Xoodyak, KNOT, GIFT-COFB, Elephant, DryGASCON, TinyJAMBU, Spook, Romulus, SATURNIN, and PHOTON-Beetle had higher throughput than AES-GCM, where the AES-GCM implementation exceeded the 5000 LE budget. TinyJAMBU, Subterranean 2.0, ESTATE, SpoC, and Romulus were the five candidates with the smallest implementations. Gimli and Xoodyak also demonstrated faster hashing than SHA-2. Subterranean 2.0, ACE, Xoodyak, and Gimli were the candidates with the smallest implementations supporting hashing.

Results were also similar on the Lattice ECP5, where Subterranean 2.0, Xoodyak, Gimli, ASCON, KNOT, GIFT-COFB, Elephant, DryGASCON, and TinyJAMBU implementations processed plaintext faster than AES-GCM; and Gimli and Xoodyak exhibited higher hashing throughput than SHA-2. Subterranean 2.0, Gimli, ACE, and Xoodyak were the candidates with the smallest implementations supporting hashing.

4.2.2 ASIC Benchmarking by Khairallah et al.

Khairallah et al. [274, 275] synthesized ten of the candidates on 65nm and 28nm technologies. This work primarily considered two use cases. The first was performance efficiency, where the throughput/area ratio is the main concern. The second use case was lightweight protocols, where Bluetooth and Bluetooth Low-Energy were selected as representatives of such protocols. Implementations were synthesized according to four different optimization goals: balanced, low-area, high-speed, and low-frequency.

Implementations were compared according to area, energy, area$\times$energy, and throughput. TinyJAMBU and Romulus had the smallest implementations and strong performance when optimized for low area. Subterranean 2.0 was the most energy-efficient. When area is limited to 9000 GE, TinyJAMBU, Romulus, Subterranean 2.0, and Xoodyak display the best results. Pyjamask fared the worst of the ten candidates in all metrics.

4.2.3 ASIC Benchmarking by Aagaard and Zidarič

Aagaard and Zidarič [276] synthesized 22 of the candidates,[1] as well as two implementations of AES-GCM. Five different ASIC cell libraries and two synthesis tools were used. The cell libraries target 65nm, 90nm, and 130nm technologies. To illustrate design tradeoffs, the study provides the ratios of throughput to area, throughput to energy, and throughput to area×energy.

Results were presented as average scaled values taken across the different configurations. Throughput results reflect steady-state plaintext processing and do not include the cost of loading the key, state initialization, or AD processing. Hashing was not evaluated in this study.

TinyJAMBU had the smallest footprint followed by Romulus; however, they both had relatively poor performance. Subterranean 2.0 had the next lowest area and has outstanding performance. The candidates with the most energy-efficient implementations were Subterranean 2.0, Gimli, Xoodyak, and ASCON. In addition, TinyJAMBU, SPIX, and COMET were also very energy efficient at lower throughputs. Subterranean 2.0 and TinyJAMBU excel once again when considering area×energy. Xoodyak and ASCON had low area×energy with good throughput, followed by Romulus. COMET also demonstrates low area×energy at low throughput.

Table 11. Summary of the hardware benchmarking initiatives

Initiative	***Platforms***	***Metrics***
GMU CERG group [270]	Xilinx Artix-7 Intel Cyclone 10 LP Lattice Semiconductor ECP5	Resource utilization (LUT or LE, flip-flops) Maximum clock frequency (MHz) Throughput (Mbits/s) Energy per bit (nJ/bit)
Khairallah et al. [274]	TSMC 65nm FDSOI 28nm	Area (μm^2 and GE) Clock period (ns) Power (mW) Energy (mJ)
Aagaard and Zidarič [276]	ST Micro 65nm TSMC 65nm ST Micro 90nm TSMC 90nm ARM/IBM 130nm	Throughput (bits per cycle) Area (GE) Energy (nJ) Area×Energy (GE×nJ) Clock Speed (GHz)

[1] LOTUS-AEAD and LOCUS-AEAD are included in the same submission package.

5. Selecting the Finalists

To focus analysis on the candidates most likely to be standardized at the end of this process, the number of candidates under consideration had to be significantly reduced. Ten finalists were selected using the evaluation criteria described in Section 2. This section summarizes the selection process.

Security. The design rationale, security analysis, and proofs provided by the submission team, as well as the third-party analysis, played a significant role in the selection. Analysis that covered only small number of rounds, and required significant time and memory complexities increased confidence in the claimed security margins. In some cases, third-party results raised concerns about the validity of security claims; in particular distinguishers on the underlying permutations or weak key classes were of concern. There were also attack results that invalidated security claims, such as practical forgery attacks or key recovery attacks.

Performance. Software and hardware performance comparisons of the candidates were conducted by multiple public initiatives (see Section 4). Candidates that did not demonstrate significant performance benefits over current NIST standards were not favored for advancement. NIST also considered the stated goals of the candidate, performance across a variety of input sizes, and how candidates with similar features compared. Designs that demonstrated ample flexibility in selecting implementation trade-offs and could be applied to a wide variety of use cases with acceptable performance were favored.

Tweak Plans. NIST also considered the *tweak plans* of the candidates (in particular, their impact on the performance and the security of the candidate) as a part of the evaluation. In general, tweaks requiring significant structural changes were not looked upon favorably during the selection.

Diversity of Candidates. During the selection, NIST also considered the *diversity* of the finalists. When there were multiple promising candidates based on the same underlying primitive (e.g., permutation, block cipher, or tweakable block cipher), they were also compared to each other, and a subset of the candidates was selected from these candidates.

NIST Portfolio. NIST also considered how candidates would fit within its existing portfolio of cryptographic standards. Some of the candidates included minor modifications to AES in ways that would potentially cause issues during adoption and use. Modes that were instantiated with AES were not favored to advance to the next round, because there are limited performance benefits in constrained devices, especially when considering side-channel resistant implementations.

NIST selected ten finalists to move forward to the next round. For applications that only require AEAD functionality, the following six candidates will be considered: Elephant, GIFT-COFB, Grain-128AEAD, ISAP, Romulus, and TinyJAMBU. For applications that also require hashing functionality, the following four candidates will be considered: ASCON, PHOTON-Beetle, SPARKLE, and Xoodyak.

Eight of the finalists, namely ASCON, Elephant, GIFT-COFB, Grain-128AEAD, ISAP, PHOTON-Beetle, Romulus, and SPARKLE, rely on thoroughly-analyzed, and previously-published building blocks and components. ASCON, GIFT-COFB, ISAP, PHOTON-Beetle, Romulus, SPARKLE, TinyJAMBU, and Xoodyak demonstrated performance advantages over NIST standards in software benchmarks and are under consideration for software applications. For hardware applications, the finalists ASCON, Elephant, GIFT-COFB, PHOTON-Beetle, Romulus, TinyJAMBU, and Xoodyak demonstrated performance advantages over NIST standards. ISAP provides promising features for applications requiring side-channel resistance.

6. Next Steps

It is estimated that the third round of evaluation and review will take approximately 12 months. The selection will consider the security of the candidates and performance on software and hardware platforms, including the performance of protected implementations. During the final round, NIST is planning to host the fifth workshop to support the standardization process.

NIST encourages further analysis of the candidates that did not advance to the final round. The candidates that proposed new modes of operation instantiated with AES might still be beneficial for general-purpose applications. The modes of operations of these candidates may be considered later under NIST's mode development project [277].

References

[1] Sönmez Turan M, McKay KA, Çalık Ç, Chang D, Bassham I Lawrence E (2019) Status Report on the First Round of the NIST Lightweight Cryptography Standardization Process (National Institute of Standards and Technology), Report. https://doi.org/10.6028/NIST.IR.8268

[2] Aagaard M, AlTawy R, Gong G, Mandal K, Rohit R (2020) Updates on ACE, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/ace_update.pdf.

[3] Dobraunig C, Eichlseder M, Mendel F, Schläffer M (2020) Status Update on Ascon v1.2, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/ascon_update.pdf.

[4] Gueron S, Jha A, Nandi M (2020) Updates on COMET, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/comet_update.pdf.

[5] Riou S (2020) DryGASCON Status Update, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/DryGASCON_20200917-status-update.pdf.

[6] Beyne T, Chen YL, Dobraunig C, Mennink B (2020) Status Update on Elephant, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/Elephant_status-update-round-2.pdf.

[7] Chakraborti A, Datta N, Jha A, Lopez CM, Nandi M, Sasaki Y (2020) [lwc-forum] Official Comment: ESTATE, Email to lwc-forum. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/ESTATE-update-email.pdf.

[8] Andreeva E, Lallemand V, Purnal A, Reyhanitabar R, Roy A, Vizár D (2020) Status Update of ForkAE, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/ForkAEUpdate.pdf.

[9] Banik S, Chakraborti A, Iwata T, Minematsu K, Nandi M, Peyrin T, Sasaki Y, Sim SM, Todo Y (2020) GIFT-COFB: NIST LWC Second-round Candidate Status Update, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/GIFT-COFB_status_update.pdf.

[10] Bernstein DJ, Kölbl S, Lucks S, Massolino PMC, Nawaz FMK, Schneider T, Schwabe P, Standaert FX, Todo Y, Viguier B (2020) Gimli: NIST LWC Second-round Candidate Status Update, Update to the NIST Lightweight Cryp-

tography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/gimli_update.pdf.

[11] Hell M, Johansson T, Meier W, Sonnerup J, Yoshida H (2020) Grain-128AEAD - Status Document, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/Grain_128AEAD_status_document.pdf.

[12] Chakraborti A, Datta N, Jha A, Nandi M (2020) HyENA-v2: Tweak Proposal for HyENA, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/hyena_update.pdf.

[13] Dobraunig C, Eichlseder M, Mangard S, Mendel F, Mennink B, Primas R, Unterluggauer T (2020) NIST Update: ISAP v2.0, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/ISAP_update_isap.pdf.

[14] Zhang W, Ding T, Yang B, Bao Z, Xiang Z, Ji F, Zhao X, Zhou C (2020) Update on Security Analysis and Implementations of KNOT, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/KNOT_Update.pdf.

[15] Bhattacharjee A, List E, López CM, Nandi M (2020) Tweaks for Oribatida v1.3, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/oribatida_nist_lwc_2020_oribatida-v13_update.pdf.

[16] Bao Z, Chakraborti A, Datta N, Guo J, Nandi M, Peyrin T, Yasuda K (2020) PHOTON-Beetle Authenticated Encryption and Hash Family — Updated on Software Implementations, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/PHOTON-Beetle_software_update_18Sep2020.pdf.

[17] Iwata T, Khairallah M, Minematsu K, Peyrin T (2020) Romulus for Round 3, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/Romulus-for-Round-3.pdf.

[18] Canteaut A, Duval S, Leurent G, Naya-Plasencia M, Perrin L, Pornin T, Schrottenloher A (2020) An Update on Saturnin, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/Saturnin_update.pdf.

[19] Beierle C, Jean J, Kölbl S, Leander G, Moradi A, Peyrin T, Sasaki Y, Sasdrich P, Sim SM (2020) SKINNY-AEAD and SKINNY-Hash: NIST LWC

Second-round Candidate Status Update, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/SKINNY-AEAD_and_SKINNY-Hash_status_update.pdf.

[20] Beierle C, Biryukov A, dos Santos LC, Großschädl J, Perrin L, Udovenko A, Velichkov V, Wang Q (2020) An Update on the Sparkle Suite, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/SPARKLE_update.pdf.

[21] AlTawy R, Gong G, He M, Mandal K, Rohit R (2020) Updates on Spix, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/spix_update.pdf.

[22] AlTawy R, Gong G, He M, Mandal K, Nandi M, Rohit R (2020) Updates on SpoC, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/spoc_update.pdf.

[23] Bellizia D, Berti F, Bronchain O, Cassiers G, Duval S, Guo C, Leander G, Leurent G, Levi I, Momin C, Pereira O, Peters T, Standaert FX, Udvarhelyi B, Wiemer F (2020) Spook: Updates on the Round-2 Submission, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/Spook_updates.pdf.

[24] Daemen J, Massolino PMC, Mehrdad A, Rotella Y (2020) Updates on the Subterranean 2.0 Cipher Suite, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/subterranean_r2Update.pdf.

[25] Banik S, Bogdanov A, Peyrin T, Sasaki Y, Sim SM, Tischhauser E, Todo Y (2020) SUNDAE-GIFT: Status Update, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/SUNDAE-GIFT_Status-Update.pdf.

[26] Wu H, Huang T (2020) TinyJAMBU Update, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/TinyJambu_update_20200918.pdf.

[27] Aagaard M, AlTawy R, Gong G, Mandal K, Rohit R, Zidaric N (2020) Updates on WAGE, Update to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/wage_update.pdf.

[28] Daemen J, Hoffert S, Mella S, Peeters M, Assche GV, Keer RV (2020) Xoodyak, An Update, Update to the NIST Lightweight Cryptography Standardiza-

tion Process. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/Xoodyak-update.pdf.

[29] McKay KA, Bassham I Lawrence E, Sönmez Turan M, Mouha N (2017) Report on Lightweight Cryptography (National Institute of Standards and Technology), Report. https://doi.org/10.6028/NIST.IR.8114

[30] Bassham L, Çalık Ç, McKay K, Mouha N, Sönmez Turan M (2017) Profiles for the Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Publications/white-paper/2017/04/26/profiles-for-lightweight-cryptography-standardization-process/draft/documents/profiles-lwc-std-proc-draft.pdf.

[31] NIST (2018) Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf.

[32] Dworkin MJ (2007) Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC (National Institute of Standards and Technology), Report. https://doi.org/10.6028/NIST.SP.800-38D

[33] National Institute of Standards and Technology (2015) Secure Hash Standard (SHS) (U.S. Department of Commerce), Report. https://doi.org/10.6028/NIST.FIPS.180-4

[34] Aagaard M, AlTawy R, Gong G, Mandal K, Rohit R (2019) ACE: An Authenticated Encryption and Hash Algorithm, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[35] Dobraunig C, Eichlseder M, Mendel F, Schläffer M (2019) Ascon, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[36] Gueron S, Jha A, Nandi M (2019) COMET: COunter Mode Encryption with authentication Tag, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[37] Riou S (2019) DryGASCON, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[38] Beyne T, Chen YL, Dobraunig C, Mennink B (2019) Elephant v1.1, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[39] Chakraborti A, Datta N, Jha A, Lopez CM, Nandi M, Sasaki Y (2019) ESTATE, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[40] Andreeva E, Lallemand V, Purnal A, Reyhanitabar R, Roy A, Vizár D (2019) ForkAE v.1, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/

round-2-candidates.

[41] Banik S, Chakraborti A, Iwata T, Minematsu K, Nandi M, Peyrin T, Sasaki Y, Sim SM, Todo Y (2019) GIFT-COFB v1.0, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[42] Bernstein DJ, Kölbl S, Lucks S, Massolino PMC, Mendel F, Nawaz K, Schneider T, Schwabe P, Standaert FX, Todo Y, Viguier B (2019) Gimli, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[43] Hell M, Johansson T, Meier W, Sönnerup J, Yoshida H (2019) Grain-128AEAD - A Lightweight AEAD Stream Cipher, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[44] Chakraborti A, Datta N, Jha A, Nandi M (2019) HyENA, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[45] Dobraunig C, Eichlseder M, Mangard S, Mendel F, Mennink B, Primas R, Unterluggauer R (2019) ISAP v2.0, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[46] Zhang W, Ding T, Yang B, Bao Z, Xiang Z, Ji F, Zhao X (2019) KNOT, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[47] Chakraborti A, Datta N, Jha A, Lopez CM, Nandi M, Sasaki Y (2019) LOTUS-AEAD and LOCUS-AEAD, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[48] Chakraborty B, Nandi M (2019) mixFeed, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[49] Chakraborty B, Nandi M (2019) ORANGE, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[50] Bhattacharjee A, List E, López CM, Nandi M (2019) The Oribatida Family of Lightweight Authenticated Encryption Schemes, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[51] Bao Z, Chakraborti A, Datta N, Guo J, Nandi M, Peyrin T, Yasuda K (2019) PHOTON-Beetle Authenticated Encryption and Hash Family, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[52] Goudarzi D, Jean J, Kölbl S, Peyrin T, Rivain M, Sasaki Y, Sim SM (2019) Pyjamask

v1.0, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[53] Iwata T, Khairallah M, Minematsu K, Peyrin T (2019) Romulus v1.2, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[54] Naito Y, Matsui M, Sakai Y, Suzuki D, Sakiyama K, Sugawara T (2019) SAEAES, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[55] Canteaut A, Duval S, Leurent G, Naya-Plasencia M, Perrin L, Pornin T, Schrottenloher A (2019) Saturnin: A Suite of Lightweight Symmetric Algorithms for Post-Quantum Security, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[56] Beierle C, Jean J, Kölbl S, Leander G, Moradi A, Peyrin T, Sasaki Y, Sasdrich P, Sim S (2019) SKINNY-AEAD and SKINNY-Hash v1.1, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[57] Beierle C, Biryukov A, dos Santos LC, Großschädl J, Perrin L, Udovenko A, Velichkov V, Wang Q (2019) Schwaemm and Esch: Lightweight Authenticated Encryption and Hashing using the Sparkle Permutation Family, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[58] AlTawy R, Gong G, He M, Mandal K, Rohit R (2019) Spix: An Authenticated Cipher, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[59] AlTawy R, Gong G, He M, Jha A, Mandal K, Nandi M, Rohit R (2019) SpoC: An Authenticated Cipher, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[60] Bellizia D, Berti F, Bronchain O, Cassiers G, Duval S, Guo C, Leander G, Leurent G, Levi I, Momin C, Pereira O, Peters T, Standaert FX, Wiemer F (2019) Spook: Sponge-Based Leakage-Resistant Authenticated Encryption with a Masked Tweakable Block Cipher, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[61] Daemen J, Massolino PMC, Rotella Y (2019) The Subterranean 2.0 Cipher Suite, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[62] Banik S, Bogdanov A, Peyrin T, Sasaki Y, Sim SM, Tischhauser E, Todo Y (2019) SUNDAE-GIFT v1.0, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[63] Wu H, Huang T (2019) TinyJAMBU: A Family of Lightweight Authenticated Encryption Algorithms, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[64] Aagaard M, AlTawy R, Gong G, Mandal K, Rohit R, Zidaric N (2019) WAGE: An Authenticated Cipher, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[65] Daemen J, Hoffert S, Peeters M, Assche GV, Keer RV (2019) Xoodyak: A Lightweight Cryptographic Scheme, Submission to the NIST Lightweight Cryptography Standardization Process. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[66] Chakraborti A, Datta N, Jha A, Nandi M (2020) Structural Classification of Authenticated Encryption Schemes, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/structural-classification-lwc2020.pdf.

[67] Bovy E (2020) Comparison of the Second Round Candidates of the NIST Lightweight Cryptography Competition, Bachelor thesis Computing Science, Radboud University. https://www.cs.ru.nl/bachelors-theses/.

[68] Bertoni G, Daemen J, Peeters M, Assche GV (2008) On the Indifferentiability of the Sponge Construction. *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, ed Smart NP (Springer), *Lecture Notes in Computer Science*, Vol. 4965, pp 181–197. https://doi.org/10.1007/978-3-540-78967-3_11

[69] Matyas SM, Meyer CH, Oseas J (1985) Generating Strong One-way Functions with Cryptographic Algorithm, IBM Technical Disclosure Bulletin, 27:5658–5659, 1985.

[70] Merkle RC (1989) One Way Hash Functions and DES. *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, ed Brassard G (Springer), *Lecture Notes in Computer Science*, Vol. 435, pp 428–446. https://doi.org/10.1007/0-387-34805-0_40

[71] Damgård I (1989) A Design Principle for Hash Functions. *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, ed Brassard G (Springer), *Lecture Notes in Computer Science*, Vol. 435, pp 416–427. https://doi.org/10.1007/0-387-34805-0_39

[72] AlTawy R, Rohit R, He M, Mandal K, Yang G, Gong G (2018) SLISCP-light: Towards Hardware Optimized Sponge-specific Cryptographic Permutations. *ACM Trans Embed Comput Syst* 17(4):81:1–81:26. https://doi.org/10.1145/3233245

[73] Yang G, Zhu B, Suder V, Aagaard MD, Gong G (2015) The Simeck Family of

Lightweight Block Ciphers. *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, eds Güneysu T, Handschuh H (Springer), *Lecture Notes in Computer Science*, Vol. 9293, pp 307–329. https://doi.org/10.1007/978-3-662-48324-4_16

[74] Liu J, Liu G, Qu L (2020) A New Automatic Tool Searching for Impossible Differential of NIST Candidate ACE. *Mathematics* 8(9). https://doi.org/10.3390/math8091576

[75] Bagheri N (2015) Linear Cryptanalysis of Reduced-Round SIMECK Variants, Cryptology ePrint Archive, Report 2015/716. https://ia.cr/2015/716.

[76] Le DP, Lu R, Ghorbani AA (2020) Improved Fault Analysis on SIMECK Ciphers, Cryptology ePrint Archive, Report 2020/1263. https://ia.cr/2020/1263.

[77] Kölbl S, Roy A (2016) A Brief Comparison of Simon and Simeck. *Lightweight Cryptography for Security and Privacy - 5th International Workshop, LightSec 2016, Aksaray, Turkey, September 21-22, 2016, Revised Selected Papers*, ed Bogdanov A (Springer), *Lecture Notes in Computer Science*, Vol. 10098, pp 69–88. https://doi.org/10.1007/978-3-319-55714-4_6

[78] Lu J, Liu Y, Ashur T, Sun B, Li C (2020) Rotational-XOR Cryptanalysis of Simon-like Block Ciphers, Cryptology ePrint Archive, Report 2020/486. https://ia.cr/2020/486.

[79] Bertoni G, Daemen J, Peeters M, Assche GV (2012) Permutation-based Encryption, Authentication and Authenticated Encryption, DIAC – Directions in Authenticated Ciphers. https://keccak.team/files/KeccakDIAC2012.pdf.

[80] Bertoni G, Daemen J, Peeters M, Assche GV (2011) Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, eds Miri A, Vaudenay S (Springer), *Lecture Notes in Computer Science*, Vol. 7118, pp 320–337. https://doi.org/10.1007/978-3-642-28496-0_19

[81] Dobraunig C, Eichlseder M, Mendel F (2015) Heuristic Tool for Linear Cryptanalysis with Applications to CAESAR Candidates. *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, eds Iwata T, Cheon JH (Springer), *Lecture Notes in Computer Science*, Vol. 9453, pp 490–509. https://doi.org/10.1007/978-3-662-48800-3_20

[82] Todo Y (2015) Structural Evaluation by Generalized Integral Property. *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, eds Oswald E, Fischlin M (Springer), *Lecture Notes in Computer Science*, Vol. 9056, pp 287–314. https://doi.org/10.1007/978-3-662-46800-5_12

[83] Tezcan C (2016) Truncated, Impossible, and Improbable Differential Analysis of ASCON. *Proceedings of the 2nd International Conference on Information Systems Security and Privacy, ICISSP 2016, Rome, Italy, February 19-21, 2016*, eds Camp O, Furnell S, Mori P (SciTePress), pp 325–332. https://doi.org/10.5220/0005689903250332

[84] Dobraunig C, Eichlseder M, Mendel F, Schläffer M (2015) Cryptanalysis of Ascon. *Topics in Cryptology - CT-RSA 2015, The Cryptographer's Track at the RSA Conference 2015, San Francisco, CA, USA, April 20-24, 2015. Proceedings*, ed Nyberg K (Springer), *Lecture Notes in Computer Science*, Vol. 9048, pp 371–387. https://doi.org/10.1007/978-3-319-16715-2_20

[85] Li Z, Dong X, Wang X (2017) Conditional Cube Attack on Round-Reduced ASCON. *IACR Transactions on Symmetric Cryptology* 2017(1):175–202. https://doi.org/10.13154/tosc.v2017.i1.175-202

[86] Li Y, Zhang G, Wang W, Wang M (2017) Cryptanalysis of Round-reduced ASCON. *Sci China Inf Sci* 60(3):38102. https://doi.org/10.1007/s11432-016-0283-3

[87] Dwivedi AD, Kloucek M, Morawiecki P, Nikolic I, Pieprzyk J, Wójtowicz S (2017) SAT-based Cryptanalysis of Authenticated Ciphers from the CAESAR Competition. *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017) - Volume 4: SECRYPT, Madrid, Spain, July 24-26, 2017*, eds Samarati P, Obaidat MS, Cabello E (SciTePress), pp 237–246. https://doi.org/10.5220/0006387302370246

[88] Leander G, Tezcan C, Wiemer F (2018) Searching for Subspace Trails and Truncated Differentials. *IACR Trans Symmetric Cryptol* 2018(1):74–100. https://doi.org/10.13154/tosc.v2018.i1.74-100

[89] Zong R, Dong X, Wang X (2019) Collision Attacks on Round-Reduced Gimli-Hash/Ascon-Xof/Ascon-Hash, Cryptology ePrint Archive, Report 2019/1115. https://ia.cr/2019/1115.

[90] Dobraunig C, Eichlseder M, Mendel F, Schläffer M (2019) Preliminary Analysis of Ascon-Xof and Ascon-Hash. *Technical Report* https://ascon.iaik.tugraz.at/files/Preliminary_Analysis_of_Ascon-Xof_and_Ascon-Hash_v01.pdf.

[91] Tezcan C (2019) Distinguishers for Reduced Round Ascon, DryGASCON, and Shamash Permutations, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Presentations/distinguishers-for-reduced-round-ascon-drygascon-a/images-media/session5-tezcan-distinguishers-reduced-round.pdf.

[92] Dobraunig C, Eichlseder M, Mendel F, Schläffer M (2019) Ascon v1.2 – Analysis of Security and Efficiency, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/documents/papers/ascon-1-2-analysis-of-security-lwc2019.pdf.

[93] Ramezanpour K, Abdulgadir A, William Diehl JPK, Ampadu P (2020) Active and Passive Side-Channel Key Recovery Attacks on Ascon, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/

media/Events/lightweight-cryptography-workshop-2020/documents/papers/active-passive-recovery-attacks-ascon-lwc2020.pdf.

[94] National Institute of Standards and Technology (2001) Advanced Encryption Standard (AES) (U.S. Department of Commerce), Report. https://doi.org/10.6028/NIST.FIPS.197

[95] Koo B, Roh D, Kim H, Jung Y, Lee D, Kwon D (2017) CHAM: A Family of Lightweight Block Ciphers for Resource-Constrained Devices. *Information Security and Cryptology - ICISC 2017 - 20th International Conference, Seoul, South Korea, November 29 - December 1, 2017, Revised Selected Papers*, eds Kim H, Kim D (Springer), *Lecture Notes in Computer Science*, Vol. 10779, pp 3–25. https://doi.org/10.1007/978-3-319-78556-1_1

[96] Beaulieu R, Shors D, Smith J, Treatman-Clark S, Weeks B, Wingers L (2013) The SIMON and SPECK Families of Lightweight Block Ciphers, Cryptology ePrint Archive, Report 2013/404. http://ia.cr/2013/404.

[97] Gueron S, Jha A, Nandi M (2019) On the Security of COMET Authenticated Encryption Scheme, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/documents/papers/on-sthe-security-of-comet-lwc2019.pdf.

[98] Khairallah M (2020) Weak Keys in the Rekeying Paradigm: Application to COMET and mixFeed. *IACR Transactions on Symmetric Cryptology* 2019(4):272–289. https://doi.org/10.13154/tosc.v2019.i4.272-289

[99] Bernstein DJ, Gilbert H, Sönmez Turan M (2020) Observations on COMET, Cryptology ePrint Archive, Report 2020/1445. https://ia.cr/2020/1445.

[100] Gueron S, Jha A, Nandi M (2020) Revisiting the Security of COMET Authenticated Encryption Scheme, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/revisiting-security-of-comet-lwc2020.pdf.

[101] Bogdanov A, Knezevic M, Leander G, Toz D, Varıcı K, Verbauwhede I (2011) Spongent: A Lightweight Hash Function. *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, eds Preneel B, Takagi T (Springer), *Lecture Notes in Computer Science*, Vol. 6917, pp 312–325. https://doi.org/10.1007/978-3-642-23951-9_21

[102] National Institute of Standards and Technology (2015) SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions (U.S. Department of Commerce), Report. https://doi.org/10.6028/NIST.FIPS.202

[103] Zhou H, Zong R, Dong X, Jia K, Meier W (2020) Interpolation Attacks on Round-Reduced Elephant, Kravatte and Xoofff, Cryptology ePrint Archive, Report 2020/781. https://ia.cr/2020/781.

[104] Sun L, Wang W, Wang M (2020) MILP-aided Bit-based Division Property for Primitives with Non-bit-permutation Linear Layers. *IET Inf Secur* 14(1):12–20. https://doi.org/10.1049/iet-ifs.2018.5283

[105] Bogdanov A, Knezevic M, Leander G, Toz D, c ıKV, Verbauwhede I (2013) SPONGENT: The Design Space of Lightweight Cryptographic Hashing. *IEEE Trans Computers* 62(10):2041–2053. https://doi.org/10.1109/TC.2012.196

[106] Zhang G, Liu M (2016) A Distinguisher on PRESENT-Like Permutations with Application to SPONGENT, Cryptology ePrint Archive, Report 2016/236. http://ia.cr/2016/236.

[107] Bonnetain X, Jaques S (2020) Quantum Period Finding against Symmetric Primitives in Practice. *CoRR* abs/2011.07022. Available at https://arxiv.org/abs/2011.07022.

[108] Shi T, Wu W, Hu B, Guan J, Wang S (2021) Breaking LWC Candidates: sESTATE and Elephant in Quantum Setting. *Des Codes Cryptogr* https://doi.org/10.1007/s10623-021-00875-7

[109] Beyne T, Chen YL, Dobraunig C, Mennink B (2020) Updates on Elephant, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/update-on-elephant-lwc2020.pdf.

[110] Black J, Rogaway P (2005) CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions. *J Cryptol* 18(2):111–131. https://doi.org/10.1007/s00145-004-0016-3

[111] Banik S, Bogdanov A, Luykx A, Tischhauser E (2018) SUNDAE: Small Universal Deterministic Authenticated Encryption for the Internet of Things. *IACR Trans Symmetric Cryptol* 2018(3):1–35. https://doi.org/10.13154/tosc.v2018.i3.1-35

[112] Chakraborti A, Datta N, Jha A, Lopez CM, Nandi M, Sasaki Y (2019) ESTATE Authenticated Encryption Mode: Hardware Benchmarking and Security Analysis, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/documents/papers/estate-authenticated-encryption-mode-lwc2019.pdf.

[113] Chakraborti A, Datta N, Jha A, Mancillas-López C, Nandi M, Sasaki Y (2020) ESTATE: A Lightweight and Low Energy Authenticated Encryption Mode. *IACR Trans Symmetric Cryptol* 2020(S1):350–389. https://doi.org/10.13154/tosc.v2020.iS1.350-389

[114] Andreeva E, Reyhanitabar R, Varici K, Vizár D (2018) Forking a Blockcipher for Authenticated Encryption of Very Short Messages, Cryptology ePrint Archive, Report 2018/916. https://ia.cr/2018/916.

[115] Andreeva E, Lallemand V, Purnal A, Reyhanitabar R, Roy A, Vizar D (2019) Forkcipher: A New Primitive for Authenticated Encryption of Very Short Messages, Cryptology ePrint Archive, Report 2019/1004. https://ia.cr/2019/1004.

[116] Andreeva E, Lallemand V, Purnal A, Reyhanitabar R, Roy A, Vizár D (2019) Forkcipher: A New Primitive for Authenticated Encryption of Very Short Messages. *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part II*, eds Galbraith SD, Moriai S (Springer),

Lecture Notes in Computer Science, Vol. 11922, pp 153–182. https://doi.org/10.1007/978-3-030-34621-8_6

[117] Bariant A, David N, Leurent G (2020) Cryptanalysis of Forkciphers. *IACR Transactions on Symmetric Cryptology* 2020(1):233–265. https://doi.org/10.13154/tosc.v2020.i1.233-265

[118] Andreeva E, Bhati AS, Vizar D (2021) RUP Security of the SAEF Authenticated Encryption Mode, Cryptology ePrint Archive, Report 2021/103. https://ia.cr/2021/103.

[119] Andreeva E, Bhati AS, Vizar D (2020) Nonce-Misuse Security of the SAEF Authenticated Encryption Mode, Cryptology ePrint Archive, Report 2020/1524. https://ia.cr/2020/1524.

[120] Andreeva E, Deprez A, Pittevils J, Roy A, Bhati AS, Vizár D (2020) New Results and Insighs on ForkAE, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/new-results-ForkAE-lwc2020.pdf.

[121] Banik S, Pandey SK, Peyrin T, Sasaki Y, Sim SM, Todo Y (2017) GIFT: A Small Present, Cryptology ePrint Archive, Report 2017/622. https://ia.cr/2017/622.

[122] Banik S, Pandey SK, Peyrin T, Sasaki Y, Sim SM, Todo Y (2017) GIFT: A Small Present - Towards Reaching the Limit of Lightweight Encryption. *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, eds Fischer W, Homma N (Springer), *Lecture Notes in Computer Science*, Vol. 10529, pp 321–345. https://doi.org/10.1007/978-3-319-66787-4_16

[123] Chakraborti A, Iwata T, Minematsu K, Nandi M (2017) Blockcipher-Based Authenticated Encryption: How Small Can We Go? *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, eds Fischer W, Homma N (Springer), *Lecture Notes in Computer Science*, Vol. 10529, pp 277–298. https://doi.org/10.1007/978-3-319-66787-4_14

[124] Chakraborti A, Iwata T, Minematsu K, Nandi M (2020) Blockcipher-Based Authenticated Encryption: How Small Can We Go? *J Cryptol* 33(3):703–741. https://doi.org/10.1007/s00145-019-09325-z.

[125] Zhu B, Dong X, Yu H (2018) MILP-based Differential Attack on Round-reduced GIFT, Cryptology ePrint Archive, Report 2018/390. https://ia.cr/2018/390.

[126] Li L, Wu W, Zheng Y, Zhang L (2019) The Relationship between the Construction and Solution of the MILP Models and Applications, Cryptology ePrint Archive, Report 2019/049. https://ia.cr/2019/049.

[127] Liu Y, Sasaki Y (2019) Related-Key Boomerang Attacks on GIFT with Automated Trail Search Including BCT Effect, Cryptology ePrint Archive, Report 2019/669. https://ia.cr/2019/669.

[128] Ji F, Zhang W, Zhou C, Ding T (2020) Improved (Related-key) Differential Cryptanalysis on GIFT, Cryptology ePrint Archive, Report 2020/1242. https://ia.cr/2020/

1242.

[129] Bernstein DJ, Kölbl S, Lucks S, Massolino PMC, Mendel F, Nawaz K, Schneider T, Schwabe P, Standaert F, Todo Y, Viguier B (2017) Gimli : A Cross-Platform Permutation. *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, eds Fischer W, Homma N (Springer), *Lecture Notes in Computer Science*, Vol. 10529, pp 299–320. https://doi.org/10.1007/978-3-319-66787-4_15

[130] Hamburg M (2017) Cryptanalysis of 22 1/2 rounds of Gimli, Cryptology ePrint Archive, Report 2017/743. https://ia.cr/2017/743.

[131] Cai J, Wei Z, Zhang Y, Sun S, Hu L (2019) Zero-sum Distinguishers for Round-reduced GIMLI Permutation. *Proceedings of the 5th International Conference on Information Systems Security and Privacy, ICISSP 2019, Prague, Czech Republic, February 23-25, 2019*, eds Mori P, Furnell S, Camp O (SciTePress), pp 38–43. https://doi.org/10.5220/0007249000380043

[132] Liu F, Isobe T, Meier W (2019) Preimages and Collisions for Up to 5-Round Gimli-Hash Using Divide-and-Conquer Methods, Cryptology ePrint Archive, Report 2019/1080. https://ia.cr/2019/1080.

[133] Liu F, Isobe T, Meier W (2020) Exploiting Weak Diffusion of Gimli: Improved Distinguishers and Preimage Attacks, Cryptology ePrint Archive, Report 2020/561. https://ia.cr/2020/561.

[134] Liu F, Isobe T, Meier W (2020) Automatic Verification of Differential Characteristics: Application to Reduced Gimli (Full Version), Cryptology ePrint Archive, Report 2020/591. https://ia.cr/2020/591.

[135] Gutiérrez AF, Leurent G, Naya-Plasencia M, Perrin L, Schrottenloher A, Sibleyras F (2020) New results on Gimli: Full-Permutation Distinguishers and Improved Collisions, Cryptology ePrint Archive, Report 2020/744. https://ia.cr/2020/744.

[136] Cid C, Robshaw M, Babbage S, Borghoff J, Velichkov V (2012) The eSTREAM Portfolio in 2012, eSTREAM: the ECRYPT Stream Cipher Project. https://www.ecrypt.eu.org/stream/.

[137] ISO/IEC 29167-13:2015 (2015) Information technology — Automatic identification and data capture techniques — Part 13: Crypto suite Grain-128A security services for air interface communications.

[138] Berbain C, Gilbert H, Maximov A (2006) Cryptanalysis of Grain. *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers*, ed Robshaw MJB (Springer), *Lecture Notes in Computer Science*, Vol. 4047, pp 15–29. https://doi.org/10.1007/11799313_2

[139] Banik S, Maitra S, Sarkar S, Sönmez Turan M (2013) A Chosen IV Related Key Attack on Grain-128a. *Information Security and Privacy - 18th Australasian Conference, ACISP 2013, Brisbane, Australia, July 1-3, 2013. Proceedings*, eds Boyd C, Simpson L (Springer), *Lecture Notes in Computer Science*, Vol. 7959, pp 13–26. https://doi.org/10.1007/978-3-642-39059-3_2

[140] Castagnos G, Berzati A, Canovas C, Debraize B, Goubin L, Gouget A, Paillier P,

Salgado S (2009) Fault Analysis of Grain-128. *IEEE International Workshop on Hardware-Oriented Security and Trust, HOST 2009, San Francisco, CA, USA, July 27, 2009. Proceedings*, eds Tehranipoor M, Plusquellic J (IEEE Computer Society), pp 7–14. https://doi.org/10.1109/HST.2009.5225030

[141] Zhang H, Wang X (2009) Cryptanalysis of Stream Cipher Grain Family, Cryptology ePrint Archive, Report 2009/109. https://ia.cr/2009/109.

[142] Todo Y, Isobe T, Meier W, Aoki K, Zhang B (2018) Fast Correlation Attack Revisited - Cryptanalysis on Full Grain-128a, Grain-128, and Grain-v1. *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, eds Shacham H, Boldyreva A (Springer), *Lecture Notes in Computer Science*, Vol. 10992, pp 129–159. https://doi.org/10.1007/978-3-319-96881-0_5

[143] Wang Q, Hao Y, Todo Y, Li C, Isobe T, Meier W (2018) Improved Division Property Based Cube Attacks Exploiting Algebraic Properties of Superpoly. *Advances in Cryptology – CRYPTO 2018* (Springer), *Lecture Notes in Computer Science*, Vol. 10991, pp 275–305. https://doi.org/10.1007/978-3-319-96884-1_10

[144] Hao Y, Leander G, Meier W, Todo Y, Wang Q (2020) Modeling for Three-Subset Division Property Without Unknown Subset - Improved Cube Attacks Against Trivium and Grain-128AEAD. *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, eds Canteaut A, Ishai Y (Springer), *Lecture Notes in Computer Science*, Vol. 12105, pp 466–495. https://doi.org/10.1007/978-3-030-45721-1_17

[145] Chang D, Sönmez Turan M (2021) Recovering the Key from the Internal State of Grain-128AEAD, Cryptology ePrint Archive, Report 2021/439. https://ia.cr/2021/439.

[146] Mege A (2020) [lwc-forum] Official Comment: HYENA, Email to lwc-forum. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/official-comments/hyena-round2-official-comment.pdf.

[147] Chakraborti A, Datta N, Jha A, Nandi M (2020) [lwc-forum] Official Comment: HYENA, Email to lwc-forum. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-1/official-comments/HYENA-official-comment.pdf.

[148] Chakraborti A, Datta N, Jha A, Mitragotri S, Nandi M (2019) Security Analysis of HyENA Authenticated Encryption Mode, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/documents/papers/security-analysis-of-hyena-lwc2019.pdf.

[149] Chakraborti A, Datta N, Jha A, Mitragotri S, Nandi M (2020) From Combined to Hybrid: Making Feedback-based AE even Smaller. *IACR Trans Symmetric Cryptol* 2020(S1):417–445. https://doi.org/10.13154/tosc.v2020.iS1.417-445

[150] Dobraunig C, Mennink B (2019) Leakage Resilience of the ISAP Mode: a

Vulgarized Summary, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/documents/papers/leakage-resilience-isap-mode-lwc2019.pdf.

[151] Dobraunig C, Eichlseder M, Mangard S, Mendel F, Mennink B, Primas R, Unterluggauer T (2020) Isap v2.0. *IACR Trans Symmetric Cryptol* 2020(S1):390–416. https://doi.org/10.13154/tosc.v2020.iS1.390-416

[152] Dobraunig C, Mennin B (2019) Leakage Resilience of the ISAP Mode: A Vulgarized Summary, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/documents/papers/leakage-resilience-isap-mode-lwc2019.pdf.

[153] Guo J, Liu M, Song L (2016) Linear structures: Applications to cryptanalysis of round-reduced KECCAK. *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, eds Cheon JH, Takagi T, *Lecture Notes in Computer Science*, Vol. 10031, pp 249–274. https://doi.org/10.1007/978-3-662-53887-6_9

[154] Guo J, Liao G, Liu G, Liu M, Qiao K, Song L (2020) Practical collision attacks against round-reduced SHA-3. *J Cryptol* 33(1):228–270. https://doi.org/10.1007/s00145-019-09313-3

[155] Dobraunig C, Eichlseder M, Mangard S, Mendel F, Mennink B, Primas R, Unterluggauer T (2020) Updates on the Implementation Security of ISAP, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/updates-on-ISAP-lwc2020.pdf.

[156] Zhang W, Bao Z, Lin D, Rijmen V, Yang B, Verbauwhede I (2015) RECTANGLE: A Bit-slice Lightweight Block Cipher Suitable for Multiple Platforms, Sci. China Inf. Sci. 58, 1–15. https://doi.org/10.1007/s11432-015-5459-7.

[157] Ding T, Zhang W, Zhou C, Ji F (2020) An Automatic Search Tool for Iterative Trails and its Application to estimation of differentials and linear hulls, Cryptology ePrint Archive, Report 2020/1152. https://ia.cr/2020/1152.pdf.

[158] Zhang W, Ding T, Zhou C, Ji F (2020) Security Analysis of KNOT-AEAD and KNOT-Hash, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/security-analysis-of-KNOT-lwc2020.pdf.

[159] Rohit R (2019) [lwc-forum] Official Comment: KNOT, Email to lwc-forum. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-1/official-comments/KNOT-official-comment.pdf.

[160] KNOT team (2019) [lwc-forum] Official Comment: KNOT, Email to lwc-forum. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-1/official-comments/KNOT-official-comment.pdf.

[161] Rohit R (2019) [lwc-forum] Official Comment: KNOT, Email to lwc-forum. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/

round-1/official-comments/KNOT-official-comment.pdf.

[162] Sasaki Y (2018) Integer Linear Programming for Three-Subset Meet-in-the-Middle Attacks: Application to GIFT. *Advances in Information and Computer Security - 13th International Workshop on Security, IWSEC 2018, Sendai, Japan, September 3-5, 2018, Proceedings*, eds Inomata A, Yasuda K (Springer), *Lecture Notes in Computer Science*, Vol. 11049, pp 227–243. https://doi.org/10.1007/978-3-319-97916-8_15

[163] Chen H, Zong R, Dong X (2019) Improved Differential Attacks on GIFT-64. *ICICS* (Springer), *Lecture Notes in Computer Science*, Vol. 11999, pp 447–462.

[164] Cao M, Zhang W (2019) Related-Key Differential Cryptanalysis of the Reduced-Round Block Cipher GIFT. *IEEE Access* 7:175769–175778. https://doi.org/10.1109/ACCESS.2019.2957581

[165] Zhao B, Dong X, Meier W, Jia K, Wang G (2019) Generalized Related-Key Rectangle Attacks on Block Ciphers with Linear Key Schedule: Applications to SKINNY and GIFT, Cryptology ePrint Archive, Report 2019/714. https://ia.cr/2019/714.

[166] Chakraborti A, Datta N, Jha A, Lopez CM, Nandi M, Sasaki Y (2019) LOTUS and LOCUS AEAD: Hardware Benchmarking and Security Analysis, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/documents/papers/lotus-and-locus-aead-hardware-benchmarking-and-security-analysis-lwc2019.pdf.

[167] Chakraborti A, Datta N, Jha A, Mancillas-López C, Nandi M, Sasaki Y (2019) INT-RUP Secure Lightweight Parallel AE Modes. *IACR Trans Symmetric Cryptol* 2019(4):81–118. https://doi.org/10.13154/tosc.v2019.i4.81-118

[168] Khairallah M (2019) [lwc-forum] Official Comment: mixFeed, Email to lwc-forum. https://groups.google.com/a/list.nist.gov/g/lwc-forum/c/fk8XdiC4r10/m/cETppeTYAwAJ.

[169] Leurent G, Pernot C (2020) New Representations of the AES Key Schedule, Cryptology ePrint Archive, Report 2020/1253. https://ia.cr/2020/1253.

[170] Chakraborty B, Nandi M (2019) Security Proof of mixFeed, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/documents/papers/security-proof-of-mixfeed-lwc2019.pdf.

[171] Guo J, Peyrin T, Poschmann A (2011) The PHOTON Family of Lightweight Hash Functions. *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, ed Rogaway P (Springer), *Lecture Notes in Computer Science*, Vol. 6841, pp 222–239. https://doi.org/10.1007/978-3-642-22792-9_13

[172] Dobraunig C, Mendel F, Mennink B (2019) [lwc-forum] Official Comment: ORANGE, Email to lwc-forum. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-1/official-comments/ORANGE-official-comment.pdf.

[173] Dobraunig C, Mendel F, Mennink B (2020) Practical forgeries for ORANGE. *Infor-*

mation Processing Letters 159-160:105961. https://doi.org/https://doi.org/10.1016/j.ipl.2020.105961

[174] Chakraborty B, Nandi M (2019) Security Proof of ORANGE-Zest, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/documents/papers/security-proof-ORANGE-Zest-lwc2019.pdf.

[175] Sarkar S, Khairallah M, Rohit R (2019) [lwc-forum] Official Comment: ORANGE, Email to lwc-forum. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/official-comments/orange-round2-official-comment.pdf.

[176] Andreeva E, Bogdanov A, Luykx A, Mennink B, Mouha N, Yasuda K (2014) How to Securely Release Unverified Plaintext in Authenticated Encryption. *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, eds Sarkar P, Iwata T (Springer), *Lecture Notes in Computer Science*, Vol. 8873, pp 105–125. https://doi.org/10.1007/978-3-662-45611-8_6

[177] Rohit R, Sarkar S (2019) [lwc-forum] Official Comment: Oribatida, Email to lwc-forum. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/official-comments/Oribatida-round2-official-comment.pdf.

[178] Bhattacharjee A, List E, López CM, Nandi M (2019) Security Proofs for Oribatida, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/documents/papers/security-proofs-for-oribatida-lwc2019.pdf.

[179] Bhattacharjee A, López CM, List E, Nandi M (2021) The Oribatida v1.3 Family of Lightweight Authenticated Encryption Schemes. *Journal of Mathematical Cryptology* Volume 15 Issue 1:305–344. Available at https://doi.org/10.1515/jmc-2020-0018.

[180] ISO (2016) ISO/IEC 29192-5:2016, Information Technology - Security Techniques - Lightweight Cryptography - Part 5: Hash-functions. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=67173.

[181] Chakraborti A, Datta N, Nandi M, Yasuda K (2018) Beetle Family of Lightweight and Secure Authenticated Encryption Ciphers. *IACR Trans Cryptogr Hardw Embed Syst* 2018(2):218–241. https://doi.org/10.13154/tches.v2018.i2.218-241

[182] Chakraborty B, Jha A, Nandi M (2019) Security Proof of Beetle and SpoC, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/documents/papers/security-proof-of-beetle-spoc-proof-lwc2019.pdff.

[183] Chakraborty B, Jha A, Nandi M (2019) On the Security of Sponge-type Authenticated Encryption Modes, Cryptology ePrint Archive, Report 2019/1475. https://ia.cr/2019/1475.

[184] Chakraborty B, Jha A, Nandi M (2020) On the Security of Sponge-type Au-

thenticated Encryption Modes. *IACR Trans Symmetric Cryptol* 2020(2):93–119. https://doi.org/10.13154/tosc.v2020.i2.93-119

[185] Dobraunig C, Mennink B (2020) [lwc-forum] Official Comment: PHOTON-Beetle, Email to lwc-forum. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/official-comments/photon-beetle-round2-official-comment.pdf.

[186] Wang Q, Grassi L, Rechberger C (2018) Zero-Sum Partitions of PHOTON Permutations. *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, ed Smart NP (Springer), *Lecture Notes in Computer Science*, Vol. 10808, pp 279–299. https://doi.org/10.1007/978-3-319-76953-0_15

[187] Cui T, Sun L, Chen H, Wang M (2017) Statistical Integral Distinguisher with Multi-structure and Its Application on AES. *Information Security and Privacy - 22nd Australasian Conference, ACISP 2017, Auckland, New Zealand, July 3-5, 2017, Proceedings, Part I*, eds Pieprzyk J, Suriadi S (Springer), *Lecture Notes in Computer Science*, Vol. 10342, pp 402–420. https://doi.org/10.1007/978-3-319-60055-0_21

[188] Jean J, Naya-Plasencia M, Peyrin T (2012) Improved Rebound Attack on the Finalist Grøstl. *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, ed Canteaut A (Springer), *Lecture Notes in Computer Science*, Vol. 7549, pp 110–126. https://doi.org/10.1007/978-3-642-34047-5_7

[189] Jean J, Naya-Plasencia M, Peyrin T (2013) Multiple Limited-Birthday Distinguishers and Applications. *Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*, eds Lange T, Lauter KE, Lisonek P (Springer), *Lecture Notes in Computer Science*, Vol. 8282, pp 533–550. https://doi.org/10.1007/978-3-662-43414-7_27

[190] Goudarzi D, Jean J, Kölbl S, Peyrin T, Rivain M, Sasaki Y, Sim SM (2020) Pyjamask: Block Cipher and Authenticated Encryption with Highly Efficient Masked Implementation. *IACR Trans Symmetric Cryptol* 2020(S1):31–59. https://doi.org/10.13154/tosc.v2020.iS1.31-59

[191] Dobraunig C, Rotella Y, Schoone J (2020) Algebraic and Higher-Order Differential Cryptanalysis of Pyjamask-96. *IACR Trans Symmetric Cryptol* 2020(1):289–312. https://doi.org/10.13154/tosc.v2020.i1.289-312

[192] Beierle C, Jean J, Kölbl S, Leander G, Moradi A, Peyrin T, Sasaki Y, Sasdrich P, Sim SM (2016) The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS. *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, eds Robshaw M, Katz J (Springer), *Lecture Notes in Computer Science*, Vol. 9815, pp 123–153. https://doi.org/10.1007/978-3-662-53008-5_5

[193] Iwata T, Khairallah M, Minematsu K, Peyrin T (2019) Updates on Romulus, Remus and TGIF, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/

documents/papers/updates-on-romulus-remus-tgif-lwc2019.pdf.

[194] Iwata T, Khairallah M, Minematsu K, Peyrin T (2019) Duel of the titans: The romulus and remus families of lightweight AEAD algorithms. *IACR Cryptol ePrint Arch* 2019:992. Available at https://ia.cr/2019/992.

[195] Iwata T, Khairallah M, Minematsu K, Peyrin T (2020) Duel of the titans: The romulus and remus families of lightweight AEAD algorithms. *IACR Trans Symmetric Cryptol* 2020(1):43–120. https://doi.org/10.13154/tosc.v2020.i1.43-120

[196] Iwata T, Khairallah M, Minematsu K, Peyrin T (2020) New Results on Romulus, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/new-results-romulus-lwc2020.pdf.

[197] Guo C, Khairallah M, Peyrin T (2020) AET-LR: Rate-1 Leakage-Resilient AEAD based on the Romulus Family, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/AET-LR-lwc2020.pdf.

[198] Delaune S, Derbez P, Vavrille M (2021) Catching the Fastest Boomerangs - Application to SKINNY, Cryptology ePrint Archive, Report 2021/020. https://ia.cr/2021/020.

[199] Delaune S, Derbez P, Vavrille M (2020) Catching the fastest boomerangs application to SKINNY. *IACR Trans Symmetric Cryptol* 2020(4):104–129. https://doi.org/10.46586/tosc.v2020.i4.104-129

[200] Tolba M, Abdelkhalek A, Youssef AM (2017) Impossible differential cryptanalysis of reduced-round SKINNY. *Progress in Cryptology - AFRICACRYPT 2017 - 9th International Conference on Cryptology in Africa, Dakar, Senegal, May 24-26, 2017, Proceedings*, eds Joye M, Nitaj A, *Lecture Notes in Computer Science*, Vol. 10239, pp 117–134. https://doi.org/10.1007/978-3-319-57339-7_7

[201] Liu G, Ghosh M, Song L (2017) Security analysis of SKINNY under related-tweakey settings (long paper). *IACR Trans Symmetric Cryptol* 2017(3):37–72. https://doi.org/10.13154/tosc.v2017.i3.37-72

[202] Sadeghi S, Mohammadi T, Bagheri N (2018) Cryptanalysis of reduced round SKINNY block cipher. *IACR Trans Symmetric Cryptol* 2018(3):124–162. https://doi.org/10.13154/tosc.v2018.i3.124-162

[203] Hadipour H, Bagheri N, Song L (2020) Improved Rectangle Attacks on SKINNY and CRAFT, Cryptology ePrint Archive, Report 2020/1317. https://ia.cr/2020/1317.

[204] Shi D, Sun S, Derbez P, Todo Y, Sun B, Hu L (2018) Programming the demirci-selçuk meet-in-the-middle attack with constraints. *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II*, eds Peyrin T, Galbraith SD (Springer), *Lecture Notes in Computer Science*, Vol. 11273, pp 3–34. https://doi.org/10.1007/978-3-030-03329-3_1

[205] Chen Q, Shi D, Sun S, Hu L (2019) Automatic demirci-selçuk meet-in-the-middle attack on SKINNY with key-bridging. *Information and Communications Security - 21st International Conference, ICICS 2019, Beijing, China, December 15-17, 2019, Revised Selected Papers*, eds Zhou J, Luo X, Shen Q, Xu Z (Springer), *Lecture Notes in Computer Science*, Vol. 11999, pp 233–247. https://doi.org/10.1007/978-3-030-41579-2_14

[206] Zhao B, Dong X, Meier W, Jia K, Wang G (2020) Generalized Related-key Rectangle Attacks on Block Ciphers with Linear Key Schedule: Applications to SKINNY and GIFT. *Des Codes Cryptogr* 88(6):1103–1126. https://doi.org/10.1007/s10623-020-00730-1

[207] Naito Y, Matsui M, Sugawara T, Suzuki D (2018) SAEB: A Lightweight Blockcipher-Based AEAD Mode of Operation. *IACR Trans Cryptogr Hardw Embed Syst* 2018(2):192–217. https://doi.org/10.13154/tches.v2018.i2.192-217

[208] Krovetz T, Rogaway P (2011) The Software Performance of Authenticated-Encryption Modes. *Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers*, ed Joux A (Springer), *Lecture Notes in Computer Science*, Vol. 6733, pp 306–327. https://doi.org/10.1007/978-3-642-21702-9_18

[209] Tolba M, ElSheikh M, Youssef AM (2020) Impossible Differential Cryptanalysis of Reduced-Round Tweakable TWINE. *Progress in Cryptology - AFRICACRYPT 2020 - 12th International Conference on Cryptology in Africa, Cairo, Egypt, July 20-22, 2020, Proceedings*, eds Nitaj A, Youssef AM (Springer), *Lecture Notes in Computer Science*, Vol. 12174, pp 91–113. https://doi.org/10.1007/978-3-030-51938-4_5

[210] Beierle C, Jean J, Kölbl S, Leander G, Moradi A, Peyrin T, Sasaki Y, Sasdrich P, Sim SM (2020) SKINNY-AEAD and SKINNY-Hash. *IACR Trans Symmetric Cryptol* 2020(S1):88–131. https://doi.org/10.13154/tosc.v2020.iS1.88-131

[211] Beierle C, Biryukov A, dos Santos LC, Großschädl J, Perrin L, Udovenko A, Velichkov V, Wang Q (2020) Lightweight AEAD and Hashing using the Sparkle Permutation Family. *IACR Trans Symmetric Cryptol* 2020(S1):208–261. https://doi.org/10.13154/tosc.v2020.iS1.208-261

[212] Beierle C, Biryukov A, dos Santos LC, Großschädl J, Perrin L, Udovenko A, Velichkov V, Wang Q (2020) Alzette: A 64-Bit ARX-box - (Feat. CRAX and TRAX. *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, eds Micciancio D, Ristenpart T (Springer), *Lecture Notes in Computer Science*, Vol. 12172, pp 419–448. https://doi.org/10.1007/978-3-030-56877-1_15

[213] Liu Y, Sun S, Li C (2021) Rotational Cryptanalysis From a Differential-linear Perspective: Practical Distinguishers for Round-reduced FRIET, Xoodoo, and Alzette, Cryptology ePrint Archive, Report 2021/189. https://ia.cr/2021/189.

[214] Kraleva L, Posteuca R, Rijmen V (2020) Cryptanalysis of the Permutation Based Algorithm SpoC, Cryptology ePrint Archive, Report 2020/1072. https://ia.cr/2020/1072.

[215] Kraleva L, Posteuca R, Rijmen V (2020) Cryptanalysis of the permutation based algorithm SpoC, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/cryptanalysis-of-SpoC-lwc2020.pdf.

[216] Kraleva L, Posteuca R, Rijmen V (2020) Cryptanalysis of the permutation based algorithm spoc. *Progress in Cryptology - INDOCRYPT 2020 - 21st International Conference on Cryptology in India, Bangalore, India, December 13-16, 2020, Proceedings*, eds Bhargavan K, Oswald E, Prabhakaran M (Springer), *Lecture Notes in Computer Science*, Vol. 12578, pp 273–293. https://doi.org/10.1007/978-3-030-65277-7_12

[217] Hosoyamada A, Naya-Plasencia M, Sasaki Y (2020) Improved Attacks on sLiSCP Permutation and Tight Bound of Limited Birthday Distinguishers. *IACR Transactions on Symmetric Cryptology* 2020(4):147–172. https://doi.org/10.46586/tosc.v2020.i4.147-172

[218] Derbez P, Huynh P, Lallemand V, Naya-Plasencia M, Perrin L, Schrottenloher A (2020) Cryptanalysis Results on Spook - Bringing Full-Round Shadow-512 to the Light. *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, eds Micciancio D, Ristenpart T (Springer), *Lecture Notes in Computer Science*, Vol. 12172, pp 359–388. https://doi.org/10.1007/978-3-030-56877-1_13

[219] Bellizia D, Berti F, Bronchain O, Cassiers G, Duval S, Guo C, Leander G, Leurent G, Levi I, Momin C, Pereira O, Peters T, Standaert F, Udvarhelyi B, Wiemer F (2020) Spook: Sponge-based leakage-resistant authenticated encryption with a masked tweakable block cipher. *IACR Trans Symmetric Cryptol* 2020(S1):295–349. https://doi.org/10.13154/tosc.v2020.iS1.295-349

[220] Liu F, Isobe T, Meier W (2020) Cube-Based Cryptanalysis of Subterranean-SAE. *IACR Transactions on Symmetric Cryptology* 2019(4):192–222. https://doi.org/10.13154/tosc.v2019.i4.192-222

[221] Huang S, Wang X, Xu G, Wang M, Zhao J (2017) Conditional Cube Attack on Reduced-Round Keccak Sponge Function. *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, eds Coron J, Nielsen JB, *Lecture Notes in Computer Science*, Vol. 10211, pp 259–288. https://doi.org/10.1007/978-3-319-56614-6_9

[222] Song L, Tu Y, Shi D, Hu L (2020) Security Analysis of Subterranean 2.0, Cryptology ePrint Archive, Report 2020/1133. https://ia.cr/2020/1133.

[223] Nandi M (2009) Fast and Secure CBC-Type MAC Algorithms. *Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers*, ed Dunkelman O (Springer), *Lecture Notes in Computer Science*, Vol. 5665, pp 375–393. https://doi.org/10.1007/978-3-642-03317-9_23

[224] Mege A (2019) [lwc-forum] Official Comment: SUNDAE-GIFT, Email to lwc-forum. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-1/official-comments/SUNDAE-GIFT-official-comment.pdf.

[225] Peyrin T (2019) [lwc-forum] Official Comment: SUNDAE-GIFT, Email to lwc-forum. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-1/official-comments/SUNDAE-GIFT-official-comment.pdf.

[226] Chang D, Datta N, Dutta A, Mennink B, Nandi M, Sanadhya S, Sibleyras F (2019) Release of Unverified Plaintext: Tight Unified Model and Application to ANYDAE. *IACR Trans Symmetric Cryptol* 2019(4):119–146. https://doi.org/10.13154/tosc.v2019.i4.119-146

[227] Wu H, Huang T (2014) JAMBU Lightweight Authenticated Encryption Mode and AES-JAMBU, CAESAR competition proposal.

[228] Saha D, Sasaki Y, Shi D, Sibleyras F, Sun S, Zhang Y (2020) On the Security Margin of TinyJAMBU with Refined Differential and Linear Cryptanalysis, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/security-margin-tinyJAMBU-lwc2020.pdf.

[229] Saha D, Sasaki Y, Shi D, Sibleyras F, Sun S, Zhang Y (2020) On the Security Margin of TinyJAMBU with Refined Differential and Linear Cryptanalysis, Cryptology ePrint Archive, Report 2020/1045. https://ia.cr/2020/1045.

[230] Saha D, Sasaki Y, Shi D, Sibleyras F, Sun S, Zhang Y (2020) On the Security Margin of TinyJAMBU with Refined Differential and Linear Cryptanalysis. *IACR Trans Symmetric Cryptol* 2020(3):152–174. https://doi.org/10.13154/tosc.v2020.i3.152-174

[231] AlTawy R, Gong G, Mandal K, Rohit R (2020) WAGE: An Authenticated Encryption with a Twist. *IACR Trans Symmetric Cryptol* 2020(S1):132–159. https://doi.org/10.13154/tosc.v2020.iS1.132-159

[232] Nawaz Y, Gong G (2005) The WG Stream Cipher, eSTREAM, ECRYPT Stream Cipher Project, Report 2005/033. http://www.ecrypt.eu.org/stream.

[233] Nawaz Y, Gong G (2008) WG: A family of stream ciphers with designed randomness properties. *Inf Sci* 178(7):1903–1916. https://doi.org/10.1016/j.ins.2007.12.002

[234] Song L, Guo J (2018) Cube-Attack-Like Cryptanalysis of Round-Reduced KECCAK Using MILP. *IACR Trans Symmetric Cryptol* 2018(3):182–214. https://doi.org/10.13154/tosc.v2018.i3.182-214

[235] Zhou H, Li Z, Dong X, Jia K, Meier W (2020) Practical Key-Recovery Attacks On Round-Reduced Ketje Jr, Xoodoo-AE And Xoodyak. *Comput J* 63(8):1231–1246. https://doi.org/10.1093/comjnl/bxz152

[236] Liu F, Isobe T, Meier W, Yang Z (2020) Algebraic Attacks on Round-Reduced Keccak/Xoodoo, Cryptology ePrint Archive, Report 2020/346. https://ia.cr/2020/346.

[237] Belaïd S, Dagand P, Mercadier D, Rivain M, Wintersdorff R (2020) Tornado: Automatic generation of probing-secure masked bitsliced implementations. *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on*

the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III, eds Canteaut A, Ishai Y (Springer), *Lecture Notes in Computer Science*, Vol. 12107, pp 311–341. https://doi.org/10.1007/978-3-030-45727-3_11

[238] Abdulgadir A, Diehl W, Kaps JP (2019) An open-source platform for evaluating side-channel countermeasures in hardware implementations of lightweight authenticated ciphers, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/Presentations/2019/an-open-source-platform-for-evaluating-side-channe.

[239] Coleman F, Rezvani B, Sachin S, Diehl W (2020) Side Channel Resistance at a Cost: A Comparison of ARX-Based Authenticated Encryption. *2020 30th International Conference on Field-Programmable Logic and Applications (FPL)*, pp 193–199. https://doi.org/10.1109/FPL50879.2020.00040

[240] Guo C, Pereira O, Peters T, Standaert F (2019) Authenticated Encryption with Nonce Misuse and Physical Leakage: Definitions, Separation Results and First Construction - (Extended Abstract). *Progress in Cryptology - LATINCRYPT 2019 - 6th International Conference on Cryptology and Information Security in Latin America, Santiago de Chile, Chile, October 2-4, 2019, Proceedings*, eds Schwabe P, Thériault N (Springer), *Lecture Notes in Computer Science*, Vol. 11774, pp 150–172. https://doi.org/10.1007/978-3-030-30530-7_8

[241] Bellizia D, Bronchain O, Cassiers G, Grosso V, Guo C, Momin C, Pereira O, Peters T, Standaert FX (2020) Mode-Level vs. Implementation-Level Physical Security in Symmetric Cryptography: A Practical Guide Through the Leakage-Resistance Jungle, Cryptology ePrint Archive, Report 2020/211. https://ia.cr/2020/211.

[242] Mandal K, Gong G (2020) Can LWC and PEC be Friends?: Evaluating Lightweight Ciphers in Privacy-enhancing Cryptography, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/can-lwc-pec-be-friends-lwc2020.pdf.

[243] Adomnicai A, Peyrin T (2020) Fixslicing - Application to Some NIST LWC Round 2 Candidates, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/fixslicing-lwc2020.pdf.

[244] Kiaei P, Krishna AS, Schaumont P (2020) Parallel Synchronous Code Generation for Second Round Light Weight Candidates, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/parallel-synchronous-code-generation-lwc2020.pdf.

[245] Chakraborti A, Datta N, Nandi M (2016) INT-RUP Analysis of Block-cipher Based Authenticated Encryption Schemes. *Topics in Cryptology - CT-RSA 2016 - The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings*, ed Sako K (Springer), *Lecture Notes in Computer Science*, Vol. 9610, pp 39–54. https://doi.org/10.1007/

978-3-319-29485-8_3

[246] Mege A (2019) Tag Length Impact on Confidentiality Security, Email to lwc-forum. https://groups.google.com/a/list.nist.gov/g/lwc-forum/c/2a0H-HQHgqU.

[247] Ashur T, Dunkelman O, Luykx A (2017) Boosting Authenticated Encryption Robustness with Minimal Modifications. *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, eds Katz J, Shacham H (Springer), *Lecture Notes in Computer Science*, Vol. 10403, pp 3–33. https://doi.org/10.1007/978-3-319-63697-9_1

[248] Shor PW (1997) Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J Comput* 26(5):1484–1509. https://doi.org/10.1137/S0097539795293172

[249] Alagic G, Alperin-Sheriff J, Apon D, Cooper D, Dang Q, Miller C, Moody D, Peralta R, Perlner R, Robinson A, Smith-Tone D, Liu YK (2019) NISTIR 8240 Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process. https://doi.org/10.6028/NIST.IR.8240.

[250] Grover LK (1996) A Fast Quantum Mechanical Algorithm for Database Search. *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, ed Miller GL (ACM), pp 212–219. https://doi.org/10.1145/237814.237866

[251] Kuwakado H, Morii M (2012) Security on the quantum-type Even-Mansour cipher. *Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2012, Honolulu, HI, USA, October 28-31, 2012* (IEEE), pp 312–316.

[252] Kuwakado H, Morii M (2010) Quantum Distinguisher between the 3-round Feistel Cipher and the Random Permutation. *IEEE International Symposium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings* (IEEE), pp 2682–2685. https://doi.org/10.1109/ISIT.2010.5513654

[253] Kaplan M, Leurent G, Leverrier A, Naya-Plasencia M (2016) Breaking Symmetric Cryptosystems Using Quantum Period Finding. *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, eds Robshaw M, Katz J (Springer), *Lecture Notes in Computer Science*, Vol. 9815, pp 207–237. https://doi.org/10.1007/978-3-662-53008-5_8

[254] Bonnetain X, Hosoyamada A, Naya-Plasencia M, Sasaki Y, Schrottenloher A (2020) Quantum Attacks without Superposition Queries: the Offline Simon's Algorithm. *CoRR* abs/2002.12439. Available at https://arxiv.org/abs/2002.12439.

[255] NIST (2020) Benchmarking of Lightweight Cryptographic Algorithms on Microcontrollers, GitHub Repository. https://github.com/usnistgov/Lightweight-Cryptography-Benchmarking.

[256] Renner S, Pozzobon E, Mottok J LWC Benchmark, GitHub repository. https://lab.las3.de/gitlab/lwc/compare.

[257] Weatherley R Lightweight Cryptography Primitives, GitHub repository. https://

github.com/rweather/lightweight-crypto.

[258] Renner S, Pozzobon E, Mottok J NIST LWC Software Performance Benchmarks on Microcontrollers, Project Webpage. https://lwc.las3.de/ (Accessed February 27, 2021).

[259] Renner S, Pozzobon E, Mottok J (2020) Current and Future Efforts in Benchmarking NIST LWC Ciphers, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/current-and-future-efforts-in-benchmarking-lwc2020.pdf.

[260] Renner S, Pozzobon E, Mottok J (2020) A Hardware in the Loop Benchmark Suite to Evaluate NIST LWC Ciphers on Microcontrollers. *Information and Communications Security*, eds Meng W, Gollmann D, Jensen CD, Zhou J (Springer International Publishing, Cham), pp 495–509. https://doi.org/10.1007/978-3-030-61078-4_28.

[261] Bernstein DJ (2008) ChaCha, a variant of Salsa20, State of the Art of Stream Ciphers (SASC) 2008. Available at https://www.ecrypt.eu.org/stvl/sasc2008/.

[262] Bernstein DJ (2005) The Poly1305-AES Message-Authentication Code. *FSE*, eds Gilbert H, Handschuh H (Springer), *Lecture Notes in Computer Science*, Vol. 3557, pp 32–49. https://doi.org/10.1007/11502760_3.

[263] Nir Y, Langley A (2018) ChaCha20 and Poly1305 for IETF Protocols, Internet Research Task Force (IRTF), Request for Comments (RFC) 8439. https://doi.org/10.17487/RFC8439.

[264] Aumasson J, Neves S, Wilcox-O'Hearn Z, Winnerlein C (2013) BLAKE2: simpler, smaller, fast as MD5. *Applied Cryptography and Network Security - 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings*, eds Jr MJJ, Locasto ME, Mohassel P, Safavi-Naini R (Springer), *Lecture Notes in Computer Science*, Vol. 7954, pp 119–135. https://doi.org/10.1007/978-3-642-38980-1_8

[265] Campos F, Jellema L, Lemmen M, Müller L, Sprenkels D, Viguier B (2020) Assembly or Optimized C for Lightweight Cryptography on RISC-V?, Cryptology ePrint Archive, Report 2020/836. https://ia.cr/2020/836.

[266] Nisanci G, Atay R, Pehlivanoglu MK, Kavun EB, n TY (2019) Will the Future Lightweight Standard be RISC-V Friendly?, NIST Lightweight Cryptography Workshop 2019. https://csrc.nist.gov/CSRC/media/Presentations/will-the-future-lightweight-standard-be-risc-v-fri/images-media/session4-yalcin-will-future-lw-standard-be-risc-v-friendly.pdf.

[267] Bernstein D, Lange T eBACS: ECRYPT Benchmarking of Cryptographic Systems. https://bench.cr.yp.to/.

[268] dos Santos LC, Großschädl J (2020) An Evaluation of the Multi-Platform Efficiency of Lightweight Cryptographic Permutations, NIST Lightweight Cryptography Workshop 2020. https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/performance-evaluation-cryptographic-permutations-lwc2020.pdf.

[269] Fotovvat A, Rahman GME, Vedaei SS, Wahid KA (2021) Comparative performance

analysis of lightweight cryptography algorithms for iot sensor nodes. *IEEE Internet of Things Journal* 8(10):8279–8290. https://doi.org/10.1109/JIOT.2020.3044526

[270] Mohajerani K, Haeussler R, Nagpal R, Farahmand F, Abdulgadir A, Kaps JP, Gaj K (2021) FPGA Benchmarking of Round 2 Candidates in the NIST Lightweight Cryptography Standardization Process: Methodology, Metrics, Tools, and Results, Cryptology ePrint Archive, Report 2020/1207 (updated February 24, 2021). https://ia.cr/2020/1207/20210224:202629.

[271] ATHENa, Project Webpage. https://cryptography.gmu.edu/athena/index.php?id=LWC.

[272] Farahmand F, Ferozpuri A, Diehl W, Gaj K (2017) Minerva: Automated hardware optimization tool. *International Conference on ReConFigurable Computing and FPGAs, ReConFig 2017, Cancun, Mexico, December 4-6, 2017* (IEEE), pp 1–8. https://doi.org/10.1109/RECONFIG.2017.8279804

[273] Mohajerani K, Nagpal R (2020) Xeda: Cross-EDA Abstraction and Automation. https://github.com/XedaHQ/xeda.

[274] Khairallah M, Peyrin T, Chattopadhyay A (2020) Preliminary Hardware Benchmarking of a Group of Round 2 NIST Lightweight AEAD Candidates, Cryptology ePrint Archive, Report 2020/1459. https://ia.cr/2020/1459.

[275] Lightweight Cryptography ASIC Benchmarking, GitHub repository. https://github.com/mustafam001/lwc-aead-rtl.

[276] Aagaard MD, Zidaric N (2021) ASIC Benchmarking of Round 2 Candidates in the NIST Lightweight Cryptography Standardization Process, Cryptology ePrint Archive, Report 2021/049. https://ia.cr/2021/049.

[277] NIST Modes development. https://csrc.nist.gov/projects/block-cipher-techniques/bcm/modes-development.

A. NIST Software Benchmarking Results

This section presents NIST's software benchmarking results on microcontrollers (see Table 10). The benchmarking results on the NIST standards AES-GCM and SHA-256 are also included for comparison purposes. For additional results, see the GitHub repository of the project [255].

Table 12 presents the code sizes of the smallest implementations of the primary AEAD variants on each tested microcontroller. Candidates with a smaller code size than AES-GCM are highlighted. The results on code size for the primary hash variants are summarized in Table 13, where the candidates having smaller code size than SHA-256 are highlighted.

Table 14 presents the timing results for the fastest implementations of the primary AEAD variants for processing 16-byte message and 16-byte AD. Entries with a '-' sign indicate that the timing information could not be measured for the variant on that microcontroller. Table 15 shows the timing results for the fastest implementations of the primary hash variants for processing 16-byte message.

Figures 1, 2, 3, 4, 5, and 6 provide the code size and timing information for the smallest implementations of the primary AEAD variants on each microcontroller, respectively. The timing information is provided for processing a 16-byte message and 16-byte AD. Note that the code sizes belong to the smallest implementations where timing result could be collected.

Figures 7, 8, 9, 10, 11, and 12 show the timing comparison of the primary AEAD variants against AES-GCM for each microcontroller, respectively. In the figures, the relative timings for each candidate are shown by a matrix of values, where the rows correspond to the message lengths,and the columns correspond to the AD lengths. The lengths for different MCUs vary depending on the longest input that could be measured on that microcontroller. If an entry has the value *x*, it means the execution time of the candidate is *x* times the execution time of AES-GCM for that particular input length. Values less than 1 indicate a candidate performing better than AES-GCM, and these cells are highlighted.

Table 12. Code sizes (in bytes) for the smallest implementations of the primary AEAD variants on microcontrollers

	ATmega328P	ATmega4809	SAM D21	nRF52840	PIC32MX	ESP8266
AES-GCM	2810	3072	1648	1784	3012	2332
ACE	4356	3794	1948	1952	3716	2600
ASCON	3662	3590	**1392**	**1392**	**2368**	**2092**
COMET	7360	7082	7168	6944	10976	7484
DryGASCON	5810	5878	2392	2200	3956	2640
Elephant	6740	6574	3728	2600	**2944**	3108
ESTATE	6570	6436	2228	2192	3800	2656
ForkAE	13858	13434	4704	4416	7792	5712
GIFT-COFB	2948	**2882**	**1460**	**1504**	**2760**	**1836**
Gimli	3028	3092	**1136**	**1040**	**1876**	**1164**
Grain-128AEAD	9600	9418	2512	2088	4192	2904
HyENA	6676	6562	2400	2384	4428	3228
ISAP	3742	3642	**1420**	**1728**	**2708**	**1928**
KNOT	**1132**	**1124**	**1020**	**1112**	**1936**	**1288**
LOTUS-AEAD	9306	9570	3584	3464	6236	4344
mixFeed	4222	4015	2076	2248	3440	2504
ORANGE	5222	4714	3176	2712	5040	3444
Oribatida	4716	4674	2428	1880	3492	2376
PHOTON-Beetle	**1596**	**1588**	4344	3124	7852	5812
Pyjamask	4034	3858	1684	**1768**	**2988**	**2248**
Romulus	4814	4644	3800	3256	5956	4232
SAEAES	13506	13584	7472	7320	9684	7572
Saturnin	3358	3400	1780	1920	3220	**2252**
SKINNY-AEAD	10840	10934	7572	6816	6956	5112
SPARKLE	3944	3834	1656	**1688**	**2780**	**1896**
SPIX	4722	4308	2112	**1704**	3272	2424
SpoC	4040	3578	2140	2088	3464	2372
Spook	6546	6354	2724	2296	4412	2852
Subterranean 2.0	5962	5984	3344	3232	4904	3856
SUNDAE-GIFT	4798	4976	1956	**1760**	5064	**2300**
TinyJambu	3106	**2810**	**872**	**888**	**1584**	**1076**
WAGE	6052	5948	3768	3360	5804	4432
Xoodyak	2906	3072	**1260**	3252	**2020**	**1468**

Table 13. Code sizes (in bytes) for the smallest implementations of the primary hash variants on microcontrollers

	ATmega328P	ATmega4809	SAM D21	nRF52840	PIC32MX	ESP8266
SHA-256	2284	2322	944	904	1576	1084
ACE	2790	2482	1164	1216	2216	1484
ASCON	**2164**	**2148**	1064	**888**	**1404**	1124
DryGASCON	4648	4596	1320	1168	2136	1504
Gimli	**1220**	**1228**	**416**	**352**	**628**	**412**
KNOT	**2128**	**1970**	**636**	**512**	**784**	**572**
ORANGE	2882	2894	1556	1456	2580	1760
PHOTON-Beetle	**1006**	**1000**	3612	2420	6128	4532
Saturnin	2582	2664	1280	1352	2100	1572
SKINNY-HASH	2800	2694	2300	1896	2936	2136
SPARKLE	**1836**	**1938**	**848**	**752**	**1316**	**960**
Subterranean 2.0	2614	2592	2364	2352	3136	2624
Xoodyak	**1996**	**2028**	**868**	2676	**1248**	**1004**

Table 14. Timings (in µs) for the fastest implementations of primary AEAD variants for authenticated encryption of 16-byte message and 16-byte AD on microcontrollers

	ATmega328P	ATmega4809	SAM D21	nRF52840	PIC32MX	ESP8266
AES-GCM	16144	14560	1137	548	385	326
ACE	37988	37032	4708	1364	1458	1640
ASCON	**7524**	**6976**	**812**	**217**	**197**	**316**
COMET	-	-	1511	1134	549	435
DryGASCON	121952	71876	1347	**522**	418	597
Elephant	37744	37744	42349	16837	14180	28170
ESTATE	**6952**	**7548**	3259	1204	1020	1249
ForkAE	-	21064	9310	3546	3521	3750
GIFT-COFB	**5196**	**5108**	**670**	**249**	**217**	**310**
Gimli	**7852**	**7572**	**1047**	**388**	**254**	**312**
Grain-128AEAD	19812	18836	2372	688	561	755
HyENA	-	207656	52290	28798	17324	18380
ISAP	41852	37420	6405	2556	5220	5324
KNOT	**7340**	**7000**	1207	**366**	**385**	433
LOTUS-AEAD	17044	16476	2977	958	835	1386
mixFeed	**6736**	**6704**	2659	1784	821	1092
ORANGE	**8052**	**7560**	2213	1165	826	2308
Oribatida	22220	20784	3631	1055	1034	1223
PHOTON-Beetle	**12472**	**11976**	3545	1912	1355	3758
Pyjamask	**10536**	**10448**	1323	**316**	**334**	451
Romulus	**11664**	**10824**	1888	704	616	656
SAEAES	-	**6676**	**1017**	**476**	416	**258**
Saturnin	**12696**	**11472**	1632	597	510	709
SKINNY-AEAD	**13108**	**12200**	4321	796	1270	1707
SPARKLE	**3660**	**3380**	**434**	**135**	**128**	**246**
SPIX	**15300**	14920	2249	611	715	782
SpoC	**13540**	**13476**	2945	1035	972	1008
Spook	**7532**	**7156**	**935**	**294**	**270**	**317**
Subterranean 2.0	**7732**	**7168**	3817	1826	1455	2099
SUNDAE-GIFT	**8940**	**8880**	1321	616	412	596
TinyJambu	**8080**	**7964**	**774**	**283**	**320**	**324**
WAGE	23428	21112	21932	8089	6894	7674
Xoodyak	**6604**	**5908**	**541**	**155**	**137**	**204**

Table 15. Timings (in µs) for the fastest implementations of primary hash variants for processing 16-byte message on microcontrollers

	ATmega328P	ATmega4809	SAM D21	nRF52840	PIC32MX	ESP8266
SHA-256	10672	10608	211	72	56	56
ACE	**9160**	**8940**	1132	324	348	389
ASCON	**3836**	**3552**	367	95	86	123
DryGASCON	11928	10980	**177**	**50**	**53**	78
Gimli	**1972**	**1920**	270	103	61	75
KNOT	**9144**	**8100**	761	217	236	290
ORANGE	**3816**	**3584**	1043	564	399	1114
PHOTON-Beetle	**2464**	**2364**	696	377	268	742
Saturnin	**1848**	**1676**	251	88	72	101
SKINNY-HASH	45944	34596	7329	3418	2331	3803
SPARKLE	**1104**	**1036**	**141**	**39**	**40**	68
Subterranean 2.0	**4972**	**4620**	2425	1120	934	1358
Xoodyak	**1440**	**1288**	**122**	**42**	**36**	**44**

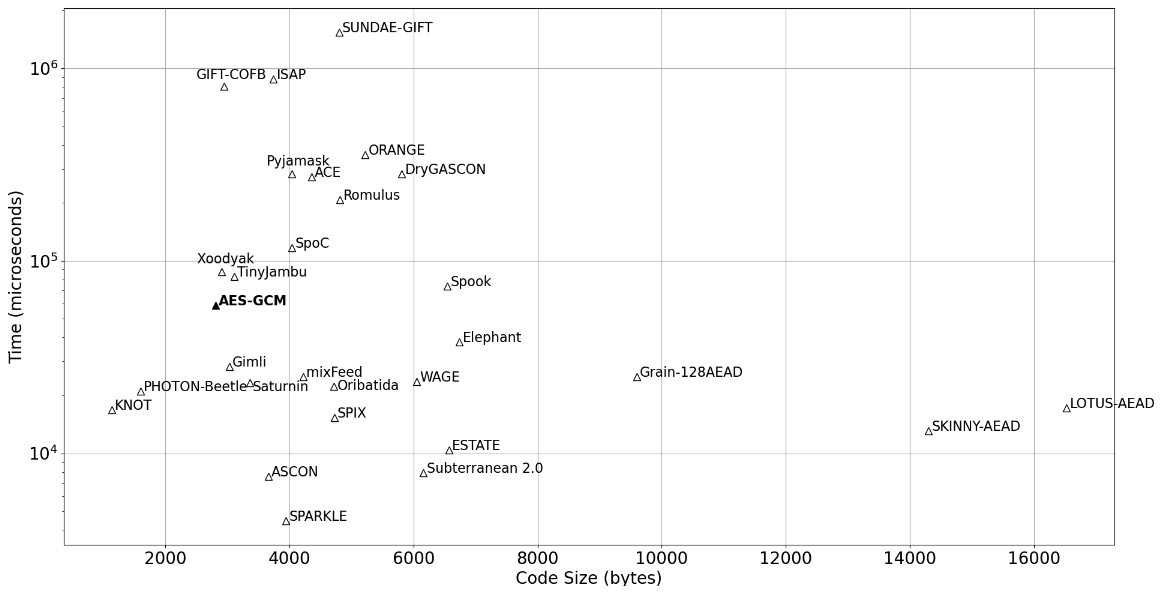


Figure 1. Code size vs. speed results of the smallest primary AEAD variants for authenticated encryption of 16-byte message and 16-byte AD on ATmega328P

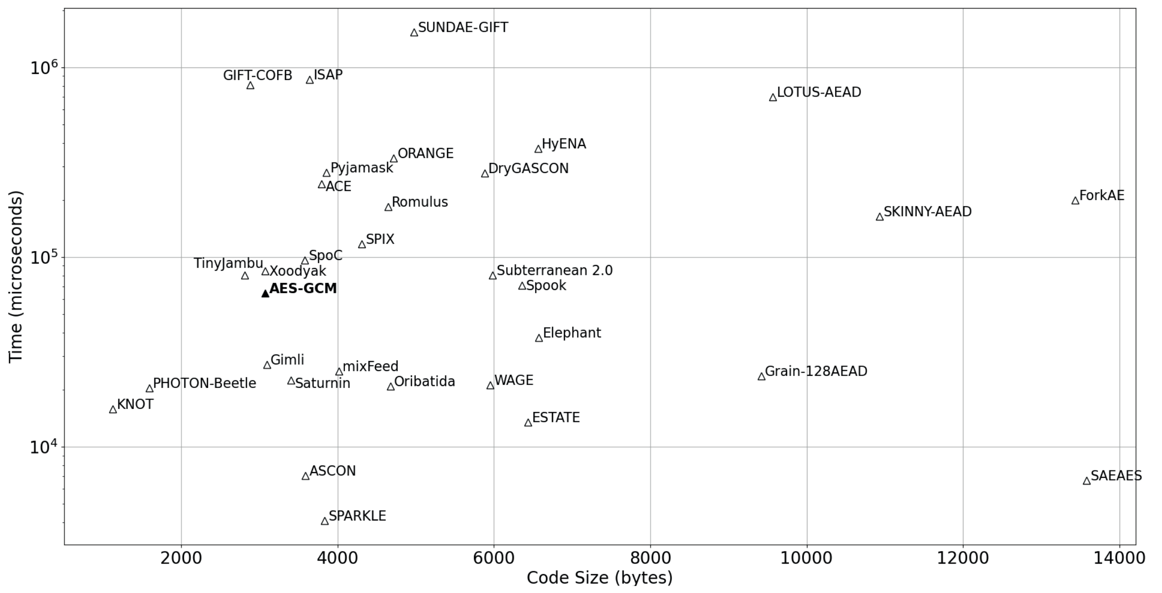


Figure 2. Code size vs. speed results of the smallest primary AEAD variants for authenticated encryption of 16-byte message and 16-byte AD on ATmega4809

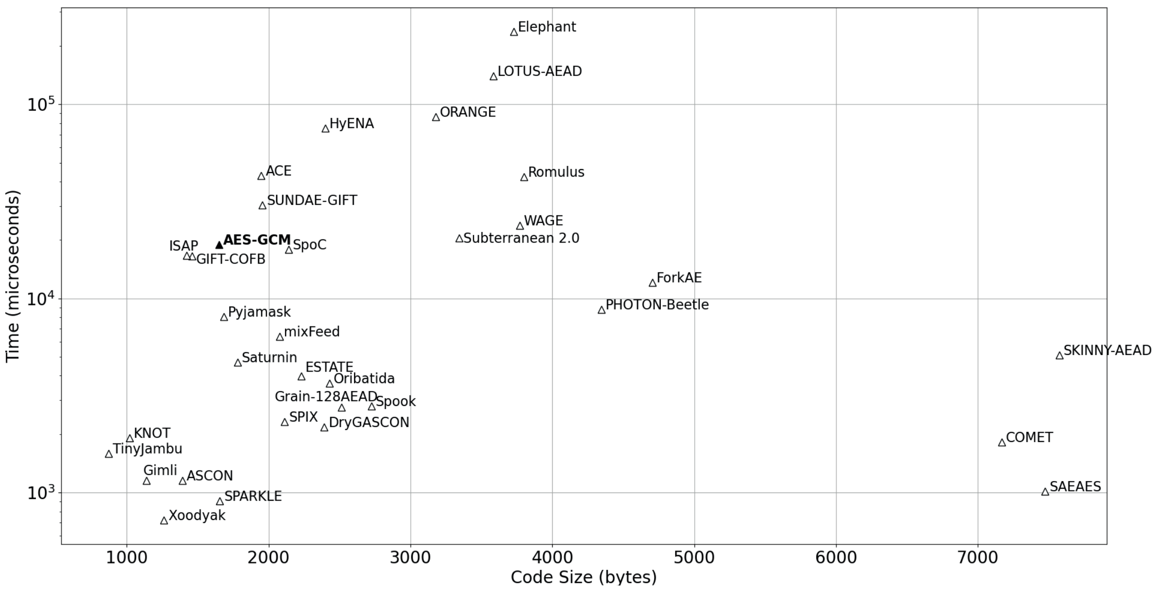


Figure 3. Code size vs. speed results of the smallest primary AEAD variants for authenticated encryption of 16-byte message and 16-byte AD on SAM D21

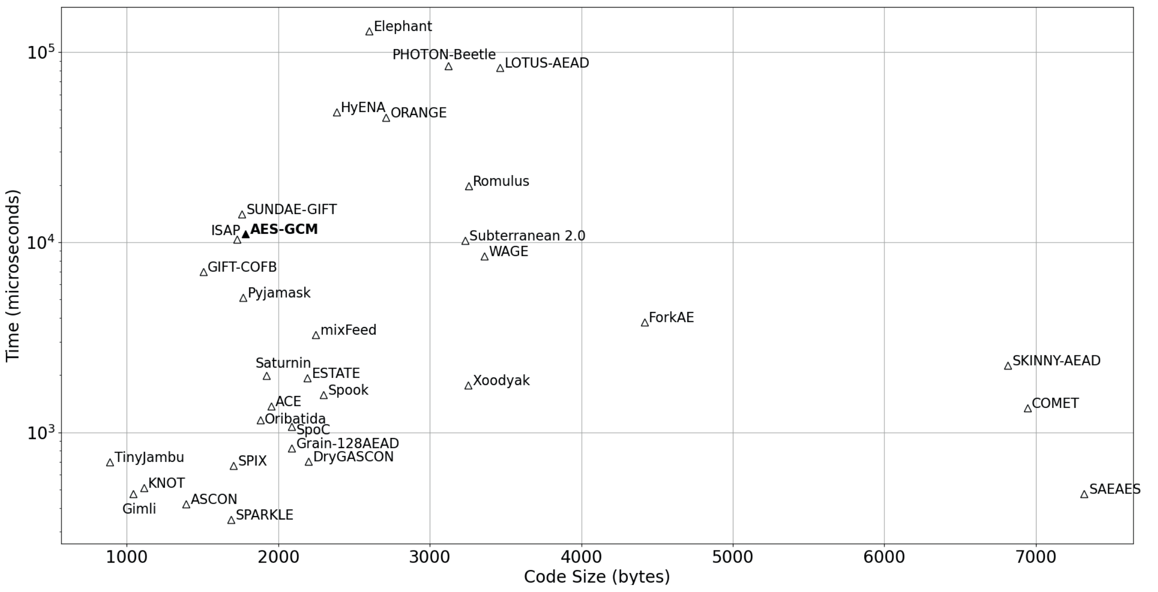


Figure 4. Code size vs. speed results of the smallest primary AEAD variants for authenticated encryption of 16-byte message and 16-byte AD on nRF52840

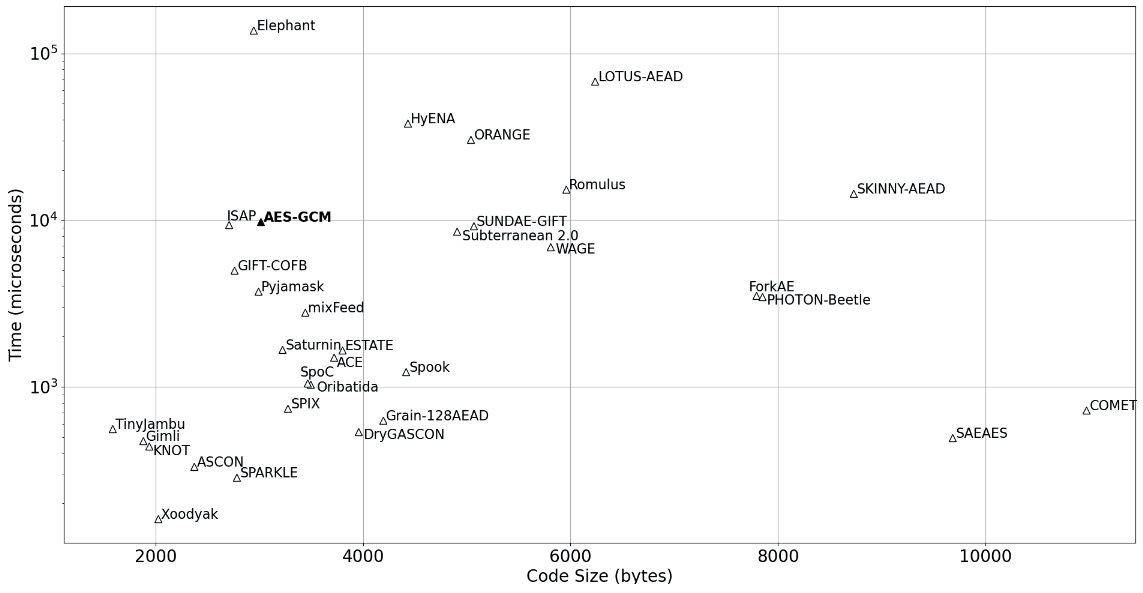


Figure 5. Code size vs. speed results of the smallest primary AEAD variants for authenticated encryption of 16-byte message and 16-byte AD on PIC32MX

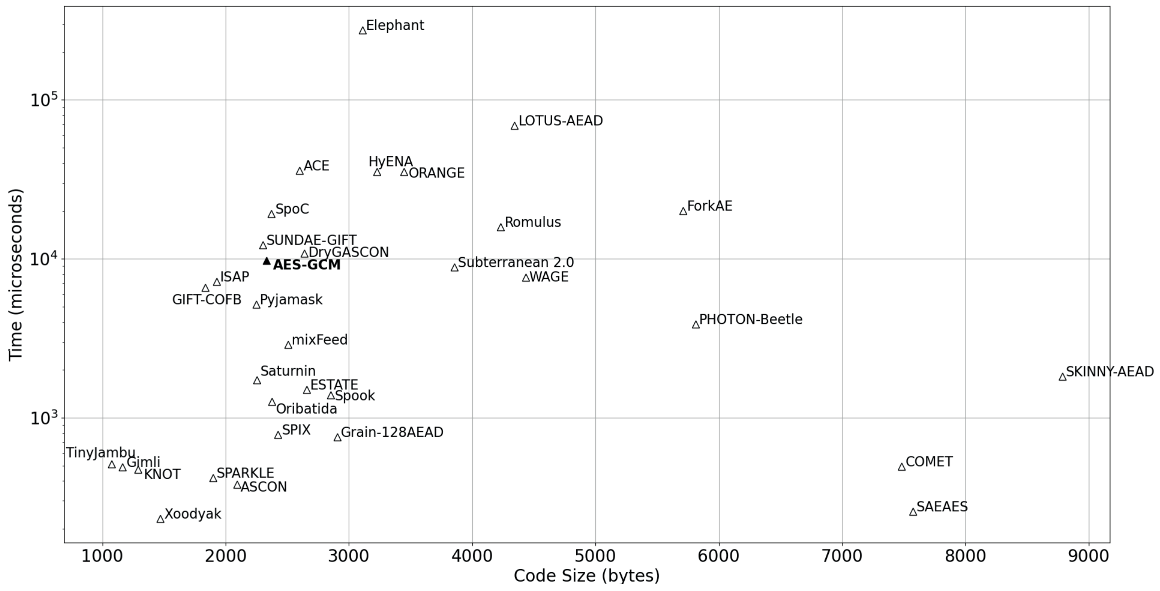
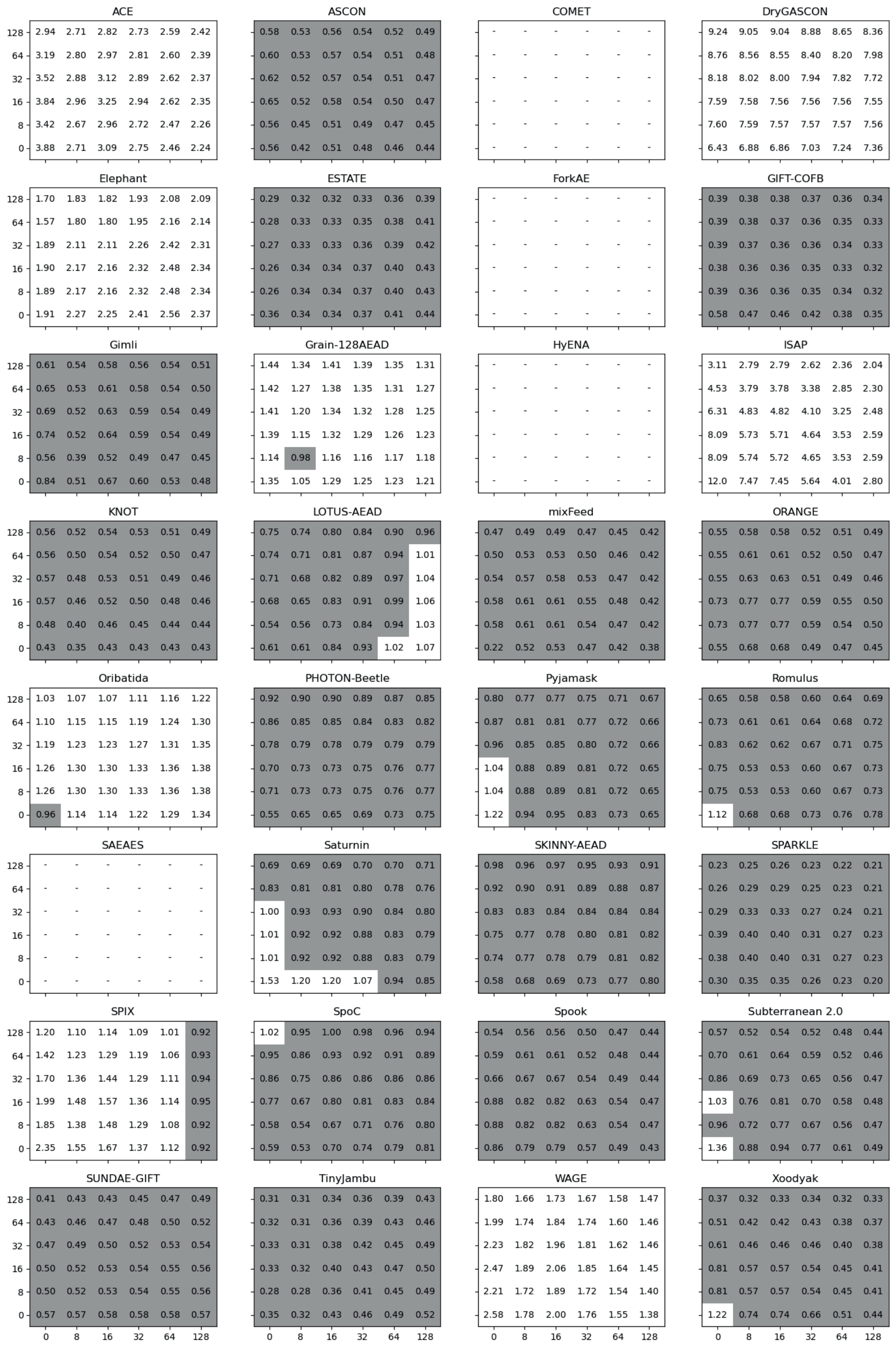


Figure 6. Code size vs. speed results of the smallest primary AEAD variants for authenticated encryption of 16-byte message and 16-byte AD on ESP8266



ACE

	0	8	16	32	64	128
128	2.94	2.71	2.82	2.73	2.59	2.42
64	3.19	2.80	2.97	2.81	2.60	2.39
32	3.52	2.88	3.12	2.89	2.62	2.37
16	3.84	2.96	3.25	2.94	2.62	2.35
8	3.42	2.67	2.96	2.72	2.47	2.26
0	3.88	2.71	3.09	2.75	2.46	2.24

ASCON

	0	8	16	32	64	128
128	0.58	0.53	0.56	0.54	0.52	0.49
64	0.60	0.53	0.57	0.54	0.51	0.48
32	0.62	0.52	0.57	0.54	0.51	0.47
16	0.65	0.52	0.58	0.54	0.50	0.47
8	0.56	0.45	0.51	0.49	0.47	0.45
0	0.56	0.42	0.51	0.48	0.46	0.44

COMET

	0	8	16	32	64	128
128	-	-	-	-	-	-
64	-	-	-	-	-	-
32	-	-	-	-	-	-
16	-	-	-	-	-	-
8	-	-	-	-	-	-
0	-	-	-	-	-	-

DryGASCON

	0	8	16	32	64	128
128	9.24	9.05	9.04	8.88	8.65	8.36
64	8.76	8.56	8.55	8.40	8.20	7.98
32	8.18	8.02	8.00	7.94	7.82	7.72
16	7.59	7.58	7.56	7.56	7.56	7.55
8	7.60	7.59	7.57	7.57	7.57	7.56
0	6.43	6.88	6.86	7.03	7.24	7.36

Elephant

	0	8	16	32	64	128
128	1.70	1.83	1.82	1.93	2.08	2.09
64	1.57	1.80	1.80	1.95	2.16	2.14
32	1.89	2.11	2.11	2.26	2.42	2.31
16	1.90	2.17	2.16	2.32	2.48	2.34
8	1.89	2.17	2.16	2.32	2.48	2.34
0	1.91	2.27	2.25	2.41	2.56	2.37

ESTATE

	0	8	16	32	64	128
128	0.29	0.32	0.32	0.33	0.36	0.39
64	0.28	0.33	0.33	0.35	0.38	0.41
32	0.27	0.33	0.33	0.36	0.39	0.42
16	0.26	0.34	0.34	0.37	0.40	0.43
8	0.26	0.34	0.34	0.37	0.40	0.43
0	0.36	0.34	0.34	0.37	0.41	0.44

ForkAE

	0	8	16	32	64	128
128	-	-	-	-	-	-
64	-	-	-	-	-	-
32	-	-	-	-	-	-
16	-	-	-	-	-	-
8	-	-	-	-	-	-
0	-	-	-	-	-	-

GIFT-COFB

	0	8	16	32	64	128
128	0.39	0.38	0.38	0.37	0.36	0.34
64	0.39	0.38	0.37	0.36	0.35	0.33
32	0.39	0.37	0.36	0.36	0.34	0.33
16	0.38	0.36	0.36	0.35	0.33	0.32
8	0.39	0.36	0.36	0.35	0.34	0.32
0	0.58	0.47	0.46	0.42	0.38	0.35

Gimli

	0	8	16	32	64	128
128	0.61	0.54	0.58	0.56	0.54	0.51
64	0.65	0.53	0.61	0.58	0.54	0.50
32	0.69	0.52	0.63	0.59	0.54	0.49
16	0.74	0.52	0.64	0.59	0.54	0.49
8	0.56	0.39	0.52	0.49	0.47	0.45
0	0.84	0.51	0.67	0.60	0.53	0.48

Grain-128AEAD

	0	8	16	32	64	128
128	1.44	1.34	1.41	1.39	1.35	1.31
64	1.42	1.27	1.38	1.35	1.31	1.27
32	1.41	1.20	1.34	1.32	1.28	1.25
16	1.39	1.15	1.32	1.29	1.26	1.23
8	1.14	0.98	1.16	1.16	1.17	1.18
0	1.35	1.05	1.29	1.25	1.23	1.21

HyENA

	0	8	16	32	64	128
128	-	-	-	-	-	-
64	-	-	-	-	-	-
32	-	-	-	-	-	-
16	-	-	-	-	-	-
8	-	-	-	-	-	-
0	-	-	-	-	-	-

ISAP

	0	8	16	32	64	128
128	3.11	2.79	2.79	2.62	2.36	2.04
64	4.53	3.79	3.78	3.38	2.85	2.30
32	6.31	4.83	4.82	4.10	3.25	2.48
16	8.09	5.73	5.71	4.64	3.53	2.59
8	8.09	5.74	5.72	4.65	3.53	2.59
0	12.0	7.47	7.45	5.64	4.01	2.80

KNOT

	0	8	16	32	64	128
128	0.56	0.52	0.54	0.53	0.51	0.49
64	0.56	0.50	0.54	0.52	0.50	0.47
32	0.57	0.48	0.53	0.51	0.49	0.46
16	0.57	0.46	0.52	0.50	0.48	0.46
8	0.48	0.40	0.46	0.45	0.44	0.44
0	0.43	0.35	0.43	0.43	0.43	0.43

LOTUS-AEAD

	0	8	16	32	64	128
128	0.75	0.74	0.80	0.84	0.90	0.96
64	0.74	0.71	0.81	0.87	0.94	1.01
32	0.71	0.68	0.82	0.89	0.97	1.04
16	0.68	0.65	0.83	0.91	0.99	1.06
8	0.54	0.56	0.73	0.84	0.94	1.03
0	0.61	0.61	0.84	0.93	1.02	1.07

mixFeed

	0	8	16	32	64	128
128	0.47	0.49	0.49	0.47	0.45	0.42
64	0.50	0.53	0.53	0.50	0.46	0.42
32	0.54	0.57	0.58	0.53	0.47	0.42
16	0.58	0.61	0.61	0.55	0.48	0.42
8	0.58	0.61	0.61	0.54	0.47	0.42
0	0.22	0.52	0.53	0.47	0.42	0.38

ORANGE

	0	8	16	32	64	128
128	0.55	0.58	0.58	0.52	0.51	0.49
64	0.55	0.61	0.61	0.52	0.50	0.47
32	0.55	0.63	0.63	0.51	0.49	0.46
16	0.73	0.77	0.77	0.59	0.55	0.50
8	0.73	0.77	0.77	0.59	0.54	0.50
0	0.55	0.68	0.68	0.49	0.47	0.45

Oribatida

	0	8	16	32	64	128
128	1.03	1.07	1.07	1.11	1.16	1.22
64	1.10	1.15	1.15	1.19	1.24	1.30
32	1.19	1.23	1.23	1.27	1.31	1.35
16	1.26	1.30	1.30	1.33	1.36	1.38
8	1.26	1.30	1.30	1.33	1.36	1.38
0	0.96	1.14	1.14	1.22	1.29	1.34

PHOTON-Beetle

	0	8	16	32	64	128
128	0.92	0.90	0.90	0.89	0.87	0.85
64	0.86	0.85	0.85	0.84	0.83	0.82
32	0.78	0.79	0.78	0.79	0.79	0.79
16	0.70	0.73	0.73	0.75	0.76	0.77
8	0.71	0.73	0.73	0.75	0.76	0.77
0	0.55	0.65	0.65	0.69	0.73	0.75

Pyjamask

	0	8	16	32	64	128
128	0.80	0.77	0.77	0.75	0.71	0.67
64	0.87	0.81	0.81	0.77	0.72	0.66
32	0.96	0.85	0.85	0.80	0.72	0.66
16	1.04	0.88	0.89	0.81	0.72	0.65
8	1.04	0.88	0.89	0.81	0.72	0.65
0	1.22	0.94	0.95	0.83	0.73	0.65

Romulus

	0	8	16	32	64	128
128	0.65	0.58	0.58	0.60	0.64	0.69
64	0.73	0.61	0.61	0.64	0.68	0.72
32	0.83	0.62	0.62	0.67	0.71	0.75
16	0.75	0.53	0.53	0.60	0.67	0.73
8	0.75	0.53	0.53	0.60	0.67	0.73
0	1.12	0.68	0.68	0.73	0.76	0.78

SAEAES

	0	8	16	32	64	128
128	-	-	-	-	-	-
64	-	-	-	-	-	-
32	-	-	-	-	-	-
16	-	-	-	-	-	-
8	-	-	-	-	-	-
0	-	-	-	-	-	-

Saturnin

	0	8	16	32	64	128
128	0.69	0.69	0.69	0.70	0.70	0.71
64	0.83	0.81	0.81	0.80	0.78	0.76
32	1.00	0.93	0.93	0.90	0.84	0.80
16	1.01	0.92	0.92	0.88	0.83	0.79
8	1.01	0.92	0.92	0.88	0.83	0.79
0	1.53	1.20	1.20	1.07	0.94	0.85

SKINNY-AEAD

	0	8	16	32	64	128
128	0.98	0.96	0.97	0.95	0.93	0.91
64	0.92	0.90	0.91	0.89	0.88	0.87
32	0.83	0.83	0.84	0.84	0.84	0.84
16	0.75	0.77	0.78	0.80	0.81	0.82
8	0.74	0.77	0.78	0.79	0.81	0.82
0	0.58	0.68	0.69	0.73	0.77	0.80

SPARKLE

	0	8	16	32	64	128
128	0.23	0.25	0.26	0.23	0.22	0.21
64	0.26	0.29	0.29	0.25	0.23	0.21
32	0.29	0.33	0.33	0.27	0.24	0.21
16	0.39	0.40	0.40	0.31	0.27	0.23
8	0.38	0.40	0.40	0.31	0.27	0.23
0	0.30	0.35	0.35	0.26	0.23	0.20

SPIX

	0	8	16	32	64	128
128	1.20	1.10	1.14	1.09	1.01	0.92
64	1.42	1.23	1.29	1.19	1.06	0.93
32	1.70	1.36	1.44	1.29	1.11	0.94
16	1.99	1.48	1.57	1.36	1.14	0.95
8	1.85	1.38	1.48	1.29	1.08	0.92
0	2.35	1.55	1.67	1.37	1.12	0.92

SpoC

	0	8	16	32	64	128
128	1.02	0.95	1.00	0.98	0.96	0.94
64	0.95	0.86	0.93	0.92	0.91	0.89
32	0.86	0.75	0.86	0.86	0.86	0.86
16	0.77	0.67	0.80	0.81	0.83	0.84
8	0.58	0.54	0.67	0.71	0.76	0.80
0	0.59	0.53	0.70	0.74	0.79	0.81

Spook

	0	8	16	32	64	128
128	0.54	0.56	0.56	0.50	0.47	0.44
64	0.59	0.61	0.61	0.52	0.48	0.44
32	0.66	0.67	0.67	0.54	0.49	0.44
16	0.88	0.82	0.82	0.63	0.54	0.47
8	0.88	0.82	0.82	0.63	0.54	0.47
0	0.86	0.79	0.79	0.57	0.49	0.43

Subterranean 2.0

	0	8	16	32	64	128
128	0.57	0.52	0.54	0.52	0.48	0.44
64	0.70	0.61	0.64	0.59	0.52	0.46
32	0.86	0.69	0.73	0.65	0.56	0.47
16	1.03	0.76	0.81	0.70	0.58	0.48
8	0.96	0.72	0.77	0.67	0.56	0.47
0	1.36	0.88	0.94	0.77	0.61	0.49

SUNDAE-GIFT

	0	8	16	32	64	128
128	0.41	0.43	0.43	0.45	0.47	0.49
64	0.43	0.46	0.47	0.48	0.50	0.52
32	0.47	0.49	0.50	0.52	0.53	0.54
16	0.50	0.52	0.53	0.54	0.55	0.56
8	0.50	0.52	0.53	0.54	0.55	0.56
0	0.57	0.57	0.58	0.58	0.58	0.57

TinyJambu

	0	8	16	32	64	128
128	0.31	0.31	0.34	0.36	0.39	0.43
64	0.32	0.31	0.36	0.39	0.43	0.46
32	0.33	0.31	0.38	0.42	0.45	0.49
16	0.33	0.32	0.40	0.43	0.47	0.50
8	0.28	0.28	0.36	0.41	0.45	0.49
0	0.35	0.32	0.43	0.46	0.49	0.52

WAGE

	0	8	16	32	64	128
128	1.80	1.66	1.73	1.67	1.58	1.47
64	1.99	1.74	1.84	1.74	1.60	1.46
32	2.23	1.82	1.96	1.81	1.62	1.46
16	2.47	1.89	2.06	1.85	1.64	1.45
8	2.21	1.72	1.89	1.72	1.54	1.40
0	2.58	1.78	2.00	1.76	1.55	1.38

Xoodyak

	0	8	16	32	64	128
128	0.37	0.32	0.33	0.34	0.32	0.33
64	0.51	0.42	0.42	0.43	0.38	0.37
32	0.61	0.46	0.46	0.46	0.40	0.38
16	0.81	0.57	0.57	0.54	0.45	0.41
8	0.81	0.57	0.57	0.54	0.45	0.41
0	1.22	0.74	0.74	0.66	0.51	0.44

Figure 7. Relative speeds of the candidates compared to AES-GCM on ATmega328P

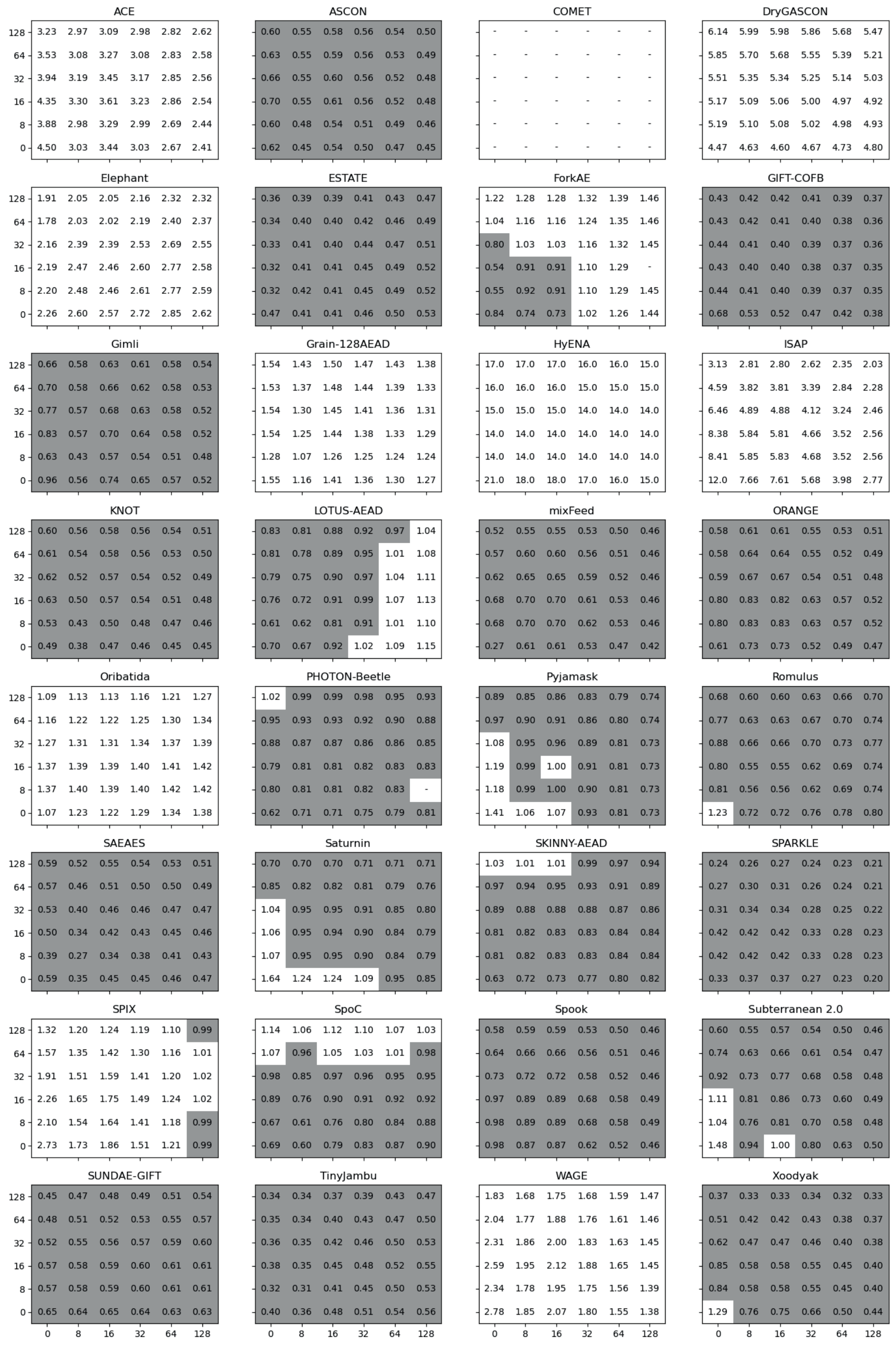
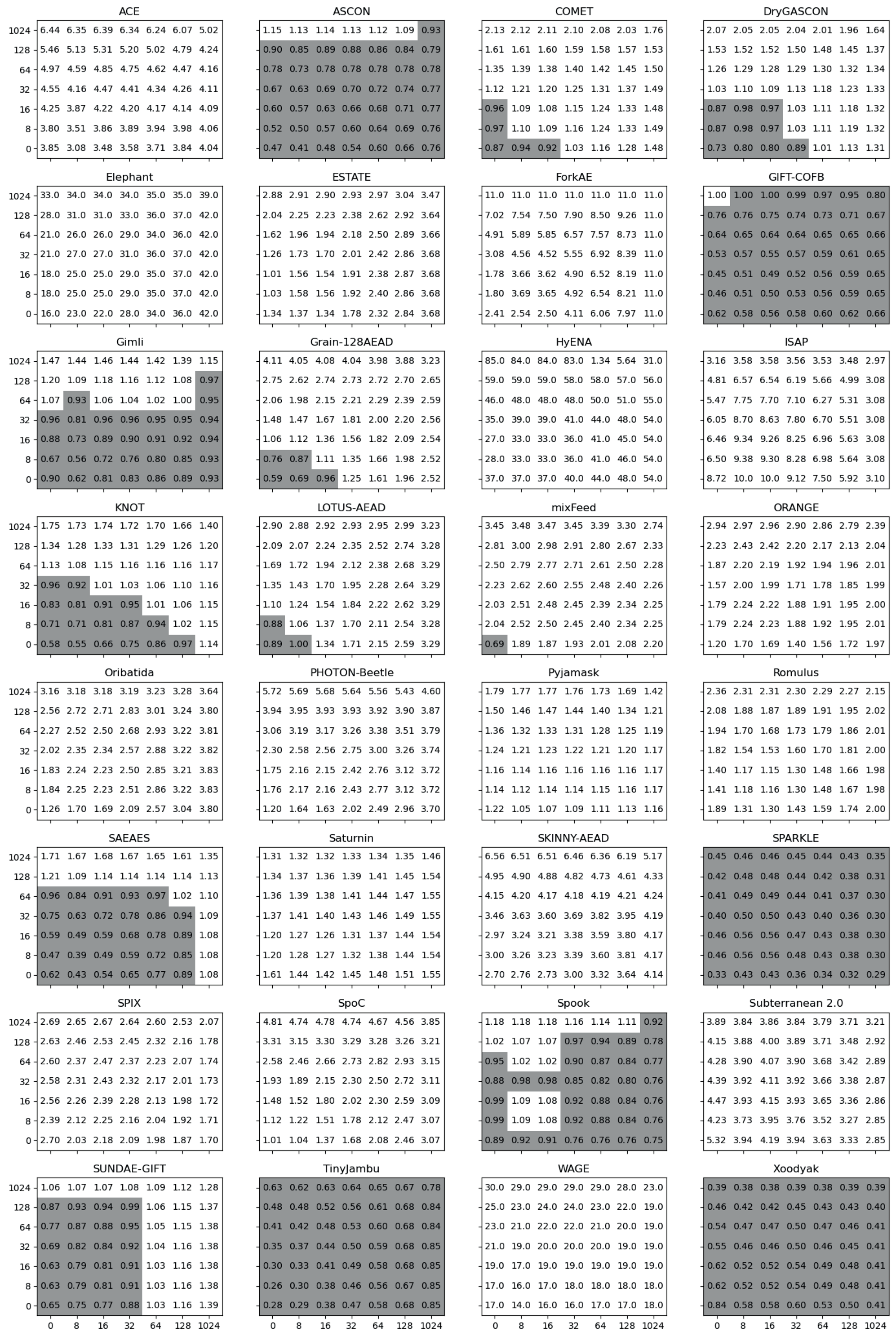


Figure 8. Relative speeds of the candidates compared to AES-GCM on ATmega4809



ACE

	0	8	16	32	64	128	1024
1024	6.44	6.35	6.39	6.34	6.24	6.07	5.02
128	5.46	5.13	5.31	5.20	5.02	4.79	4.24
64	4.97	4.59	4.85	4.75	4.62	4.47	4.16
32	4.55	4.16	4.47	4.41	4.34	4.26	4.11
16	4.25	3.87	4.22	4.20	4.17	4.14	4.09
8	3.80	3.51	3.86	3.89	3.94	3.98	4.06
0	3.85	3.08	3.48	3.58	3.71	3.84	4.04

ASCON

	0	8	16	32	64	128	1024
1024	1.15	1.13	1.14	1.13	1.12	1.09	0.93
128	0.90	0.85	0.89	0.88	0.86	0.84	0.79
64	0.78	0.73	0.78	0.78	0.78	0.78	0.78
32	0.67	0.63	0.69	0.70	0.72	0.74	0.77
16	0.60	0.57	0.63	0.66	0.68	0.71	0.77
8	0.52	0.50	0.57	0.60	0.64	0.69	0.76
0	0.47	0.41	0.48	0.54	0.60	0.66	0.76

COMET

	0	8	16	32	64	128	1024
1024	2.13	2.12	2.11	2.10	2.08	2.03	1.76
128	1.61	1.61	1.60	1.59	1.58	1.57	1.53
64	1.35	1.39	1.38	1.40	1.42	1.45	1.50
32	1.12	1.21	1.20	1.25	1.31	1.37	1.49
16	0.96	1.09	1.08	1.15	1.24	1.33	1.48
8	0.97	1.10	1.09	1.16	1.24	1.33	1.49
0	0.87	0.94	0.92	1.03	1.16	1.28	1.48

DryGASCON

	0	8	16	32	64	128	1024
1024	2.07	2.05	2.05	2.04	2.01	1.96	1.64
128	1.53	1.52	1.52	1.50	1.48	1.45	1.37
64	1.26	1.29	1.28	1.29	1.30	1.32	1.34
32	1.03	1.10	1.09	1.13	1.18	1.23	1.33
16	0.87	0.98	0.97	1.03	1.11	1.18	1.32
8	0.87	0.98	0.97	1.03	1.11	1.19	1.32
0	0.73	0.80	0.80	0.89	1.01	1.13	1.31

Elephant

	0	8	16	32	64	128	1024
1024	33.0	34.0	34.0	34.0	35.0	35.0	39.0
128	28.0	31.0	31.0	33.0	36.0	37.0	42.0
64	21.0	26.0	26.0	29.0	34.0	36.0	42.0
32	21.0	27.0	27.0	31.0	36.0	37.0	42.0
16	18.0	25.0	25.0	29.0	35.0	37.0	42.0
8	18.0	25.0	25.0	29.0	35.0	37.0	42.0
0	16.0	23.0	22.0	28.0	34.0	36.0	42.0

ESTATE

	0	8	16	32	64	128	1024
1024	2.88	2.91	2.90	2.93	2.97	3.04	3.47
128	2.04	2.25	2.23	2.38	2.62	2.92	3.64
64	1.62	1.96	1.94	2.18	2.50	2.89	3.66
32	1.26	1.73	1.70	2.01	2.42	2.86	3.68
16	1.01	1.56	1.54	1.91	2.38	2.87	3.68
8	1.03	1.58	1.56	1.92	2.40	2.86	3.68
0	1.34	1.37	1.34	1.78	2.32	2.84	3.68

ForkAE

	0	8	16	32	64	128	1024
1024	11.0	11.0	11.0	11.0	11.0	11.0	11.0
128	7.02	7.54	7.50	7.90	8.50	9.26	11.0
64	4.91	5.89	5.85	6.57	7.57	8.73	11.0
32	3.08	4.56	4.52	5.55	6.92	8.39	11.0
16	1.78	3.66	3.62	4.90	6.52	8.19	11.0
8	1.80	3.69	3.65	4.92	6.54	8.21	11.0
0	2.41	2.54	2.50	4.11	6.06	7.97	11.0

GIFT-COFB

	0	8	16	32	64	128	1024
1024	1.00	1.00	1.00	0.99	0.97	0.95	0.80
128	0.76	0.76	0.75	0.74	0.73	0.71	0.67
64	0.64	0.65	0.64	0.64	0.65	0.65	0.66
32	0.53	0.57	0.55	0.57	0.59	0.61	0.65
16	0.45	0.51	0.49	0.52	0.56	0.59	0.65
8	0.46	0.51	0.50	0.53	0.56	0.59	0.65
0	0.62	0.58	0.56	0.58	0.60	0.62	0.66

Gimli

	0	8	16	32	64	128	1024
1024	1.47	1.44	1.46	1.44	1.42	1.39	1.15
128	1.20	1.09	1.18	1.16	1.12	1.08	0.97
64	1.07	0.93	1.06	1.04	1.02	1.00	0.95
32	0.96	0.81	0.96	0.96	0.95	0.95	0.94
16	0.88	0.73	0.89	0.90	0.91	0.92	0.94
8	0.67	0.56	0.72	0.76	0.80	0.85	0.93
0	0.90	0.62	0.81	0.83	0.86	0.89	0.93

Grain-128AEAD

	0	8	16	32	64	128	1024
1024	4.11	4.05	4.08	4.04	3.98	3.88	3.23
128	2.75	2.62	2.74	2.73	2.72	2.70	2.65
64	2.06	1.98	2.15	2.21	2.29	2.39	2.59
32	1.48	1.47	1.67	1.81	2.00	2.20	2.56
16	1.06	1.12	1.36	1.56	1.82	2.09	2.54
8	0.76	0.87	1.11	1.35	1.66	1.98	2.52
0	0.59	0.69	0.96	1.25	1.61	1.96	2.52

HyENA

	0	8	16	32	64	128	1024
1024	85.0	84.0	84.0	83.0	1.34	5.64	31.0
128	59.0	59.0	59.0	58.0	58.0	57.0	56.0
64	46.0	48.0	48.0	48.0	50.0	51.0	55.0
32	35.0	39.0	39.0	41.0	44.0	48.0	54.0
16	27.0	33.0	33.0	36.0	41.0	45.0	54.0
8	28.0	33.0	33.0	36.0	41.0	46.0	54.0
0	37.0	37.0	37.0	40.0	44.0	48.0	54.0

ISAP

	0	8	16	32	64	128	1024
1024	3.16	3.58	3.58	3.56	3.53	3.48	2.97
128	4.81	6.57	6.54	6.19	5.66	4.99	3.08
64	5.47	7.75	7.70	7.10	6.27	5.31	3.08
32	6.05	8.70	8.63	7.80	6.70	5.51	3.08
16	6.46	9.34	9.26	8.25	6.96	5.63	3.08
8	6.50	9.38	9.30	8.28	6.98	5.64	3.08
0	8.72	10.0	10.0	9.12	7.50	5.92	3.10

KNOT

	0	8	16	32	64	128	1024
1024	1.75	1.73	1.74	1.72	1.70	1.66	1.40
128	1.34	1.28	1.33	1.31	1.29	1.26	1.20
64	1.13	1.08	1.15	1.16	1.16	1.16	1.17
32	0.96	0.92	1.01	1.03	1.06	1.10	1.16
16	0.83	0.81	0.91	0.95	1.01	1.06	1.15
8	0.71	0.71	0.81	0.87	0.94	1.02	1.15
0	0.58	0.55	0.66	0.75	0.86	0.97	1.14

LOTUS-AEAD

	0	8	16	32	64	128	1024
1024	2.90	2.88	2.92	2.93	2.95	2.99	3.23
128	2.09	2.07	2.24	2.35	2.52	2.74	3.28
64	1.69	1.72	1.94	2.12	2.38	2.68	3.29
32	1.35	1.43	1.70	1.95	2.28	2.64	3.29
16	1.10	1.24	1.54	1.84	2.22	2.62	3.29
8	0.88	1.06	1.37	1.70	2.11	2.54	3.28
0	0.89	1.00	1.34	1.71	2.15	2.59	3.29

mixFeed

	0	8	16	32	64	128	1024
1024	3.45	3.48	3.47	3.45	3.39	3.30	2.74
128	2.81	3.00	2.98	2.91	2.80	2.67	2.33
64	2.50	2.79	2.77	2.71	2.61	2.50	2.28
32	2.23	2.62	2.60	2.55	2.48	2.40	2.26
16	2.03	2.51	2.48	2.45	2.39	2.34	2.25
8	2.04	2.52	2.50	2.45	2.40	2.34	2.25
0	0.69	1.89	1.87	1.93	2.01	2.08	2.20

ORANGE

	0	8	16	32	64	128	1024
1024	2.94	2.97	2.96	2.90	2.86	2.79	2.39
128	2.23	2.43	2.42	2.20	2.17	2.13	2.04
64	1.87	2.20	2.19	1.92	1.94	1.96	2.01
32	1.57	2.00	1.99	1.71	1.78	1.85	1.99
16	1.79	2.24	2.22	1.88	1.91	1.95	2.00
8	1.79	2.24	2.23	1.88	1.92	1.95	2.01
0	1.20	1.70	1.69	1.40	1.56	1.72	1.97

Oribatida

	0	8	16	32	64	128	1024
1024	3.16	3.18	3.18	3.19	3.23	3.28	3.64
128	2.56	2.72	2.71	2.83	3.01	3.24	3.80
64	2.27	2.52	2.50	2.68	2.93	3.22	3.81
32	2.02	2.35	2.34	2.57	2.88	3.22	3.82
16	1.83	2.24	2.23	2.50	2.85	3.21	3.83
8	1.84	2.25	2.23	2.51	2.86	3.22	3.83
0	1.26	1.70	1.69	2.09	2.57	3.04	3.80

PHOTON-Beetle

	0	8	16	32	64	128	1024
1024	5.72	5.69	5.68	5.64	5.56	5.43	4.60
128	3.94	3.95	3.93	3.93	3.92	3.90	3.87
64	3.06	3.19	3.17	3.26	3.38	3.51	3.79
32	2.30	2.58	2.56	2.75	3.00	3.26	3.74
16	1.75	2.16	2.15	2.42	2.76	3.12	3.72
8	1.76	2.17	2.16	2.43	2.77	3.12	3.72
0	1.20	1.64	1.63	2.02	2.49	2.96	3.70

Pyjamask

	0	8	16	32	64	128	1024
1024	1.79	1.77	1.77	1.76	1.73	1.69	1.42
128	1.50	1.46	1.47	1.44	1.40	1.34	1.21
64	1.36	1.32	1.33	1.31	1.28	1.25	1.19
32	1.24	1.21	1.23	1.22	1.21	1.20	1.17
16	1.16	1.14	1.16	1.16	1.16	1.16	1.17
8	1.14	1.12	1.14	1.14	1.15	1.16	1.17
0	1.22	1.05	1.07	1.09	1.11	1.13	1.16

Romulus

	0	8	16	32	64	128	1024
1024	2.36	2.31	2.31	2.30	2.29	2.27	2.15
128	2.08	1.88	1.87	1.89	1.91	1.95	2.02
64	1.94	1.70	1.68	1.73	1.79	1.86	2.01
32	1.82	1.54	1.53	1.60	1.70	1.81	2.00
16	1.40	1.17	1.15	1.30	1.48	1.66	1.98
8	1.41	1.18	1.16	1.30	1.48	1.67	1.98
0	1.89	1.31	1.30	1.43	1.59	1.74	2.00

SAEAES

	0	8	16	32	64	128	1024
1024	1.71	1.67	1.68	1.67	1.65	1.61	1.35
128	1.21	1.09	1.14	1.14	1.14	1.14	1.13
64	0.96	0.84	0.91	0.93	0.97	1.02	1.10
32	0.75	0.63	0.72	0.78	0.86	0.94	1.09
16	0.59	0.49	0.59	0.68	0.78	0.89	1.08
8	0.47	0.39	0.49	0.59	0.72	0.85	1.08
0	0.62	0.43	0.54	0.65	0.77	0.89	1.08

Saturnin

	0	8	16	32	64	128	1024
1024	1.31	1.32	1.32	1.33	1.34	1.35	1.46
128	1.34	1.37	1.36	1.39	1.41	1.45	1.54
64	1.36	1.39	1.38	1.41	1.44	1.47	1.55
32	1.37	1.41	1.40	1.43	1.46	1.49	1.55
16	1.20	1.27	1.26	1.31	1.37	1.44	1.54
8	1.20	1.28	1.27	1.32	1.38	1.44	1.54
0	1.61	1.44	1.42	1.45	1.48	1.51	1.55

SKINNY-AEAD

	0	8	16	32	64	128	1024
1024	6.56	6.51	6.51	6.46	6.36	6.19	5.17
128	4.95	4.90	4.88	4.82	4.73	4.61	4.33
64	4.15	4.20	4.17	4.18	4.19	4.21	4.24
32	3.46	3.63	3.60	3.69	3.82	3.95	4.19
16	2.97	3.24	3.21	3.38	3.59	3.80	4.17
8	3.00	3.26	3.23	3.39	3.60	3.81	4.17
0	2.70	2.76	2.73	3.00	3.32	3.64	4.14

SPARKLE

	0	8	16	32	64	128	1024
1024	0.45	0.46	0.46	0.45	0.44	0.43	0.35
128	0.42	0.48	0.48	0.44	0.42	0.38	0.31
64	0.41	0.49	0.49	0.44	0.41	0.37	0.30
32	0.40	0.50	0.50	0.43	0.40	0.36	0.30
16	0.46	0.56	0.56	0.47	0.43	0.38	0.30
8	0.46	0.56	0.56	0.48	0.43	0.38	0.30
0	0.33	0.43	0.43	0.36	0.34	0.32	0.29

SPIX

	0	8	16	32	64	128	1024
1024	2.69	2.65	2.67	2.64	2.60	2.53	2.07
128	2.63	2.46	2.53	2.45	2.32	2.16	1.78
64	2.60	2.37	2.47	2.37	2.23	2.07	1.74
32	2.58	2.31	2.43	2.32	2.17	2.01	1.73
16	2.56	2.26	2.39	2.28	2.13	1.98	1.72
8	2.39	2.12	2.25	2.16	2.04	1.92	1.71
0	2.70	2.03	2.18	2.09	1.98	1.87	1.70

SpoC

	0	8	16	32	64	128	1024
1024	4.81	4.74	4.78	4.74	4.67	4.56	3.85
128	3.31	3.15	3.30	3.29	3.28	3.26	3.21
64	2.58	2.46	2.66	2.73	2.82	2.93	3.15
32	1.93	1.89	2.15	2.30	2.50	2.72	3.11
16	1.48	1.52	1.80	2.02	2.30	2.59	3.09
8	1.12	1.22	1.51	1.78	2.12	2.47	3.07
0	1.01	1.04	1.37	1.68	2.08	2.46	3.07

Spook

	0	8	16	32	64	128	1024
1024	1.18	1.18	1.18	1.16	1.14	1.11	0.92
128	1.02	1.07	1.07	0.97	0.94	0.89	0.78
64	0.95	1.02	1.02	0.90	0.87	0.84	0.77
32	0.88	0.98	0.98	0.85	0.82	0.80	0.76
16	0.99	1.09	1.08	0.92	0.88	0.84	0.76
8	0.99	1.09	1.08	0.92	0.88	0.84	0.76
0	0.89	0.92	0.91	0.76	0.76	0.76	0.75

Subterranean 2.0

	0	8	16	32	64	128	1024
1024	3.89	3.84	3.86	3.84	3.79	3.71	3.21
128	4.15	3.88	4.00	3.89	3.71	3.48	2.92
64	4.28	3.90	4.07	3.90	3.68	3.42	2.89
32	4.39	3.92	4.11	3.92	3.66	3.38	2.87
16	4.47	3.93	4.15	3.93	3.65	3.36	2.86
8	4.23	3.73	3.95	3.76	3.52	3.27	2.85
0	5.32	3.94	4.19	3.94	3.63	3.33	2.85

SUNDAE-GIFT

	0	8	16	32	64	128	1024
1024	1.06	1.07	1.07	1.08	1.09	1.12	1.28
128	0.87	0.93	0.94	0.99	1.06	1.15	1.37
64	0.77	0.87	0.88	0.95	1.05	1.15	1.38
32	0.69	0.82	0.84	0.92	1.04	1.16	1.38
16	0.63	0.79	0.81	0.91	1.03	1.16	1.38
8	0.63	0.79	0.81	0.91	1.03	1.16	1.38
0	0.65	0.75	0.77	0.88	1.03	1.16	1.39

TinyJambu

	0	8	16	32	64	128	1024
1024	0.63	0.62	0.63	0.64	0.65	0.67	0.78
128	0.48	0.48	0.52	0.56	0.61	0.68	0.84
64	0.41	0.42	0.48	0.53	0.60	0.68	0.84
32	0.35	0.37	0.44	0.50	0.59	0.68	0.85
16	0.30	0.33	0.41	0.49	0.58	0.68	0.85
8	0.26	0.30	0.38	0.46	0.56	0.67	0.85
0	0.28	0.29	0.38	0.47	0.58	0.68	0.85

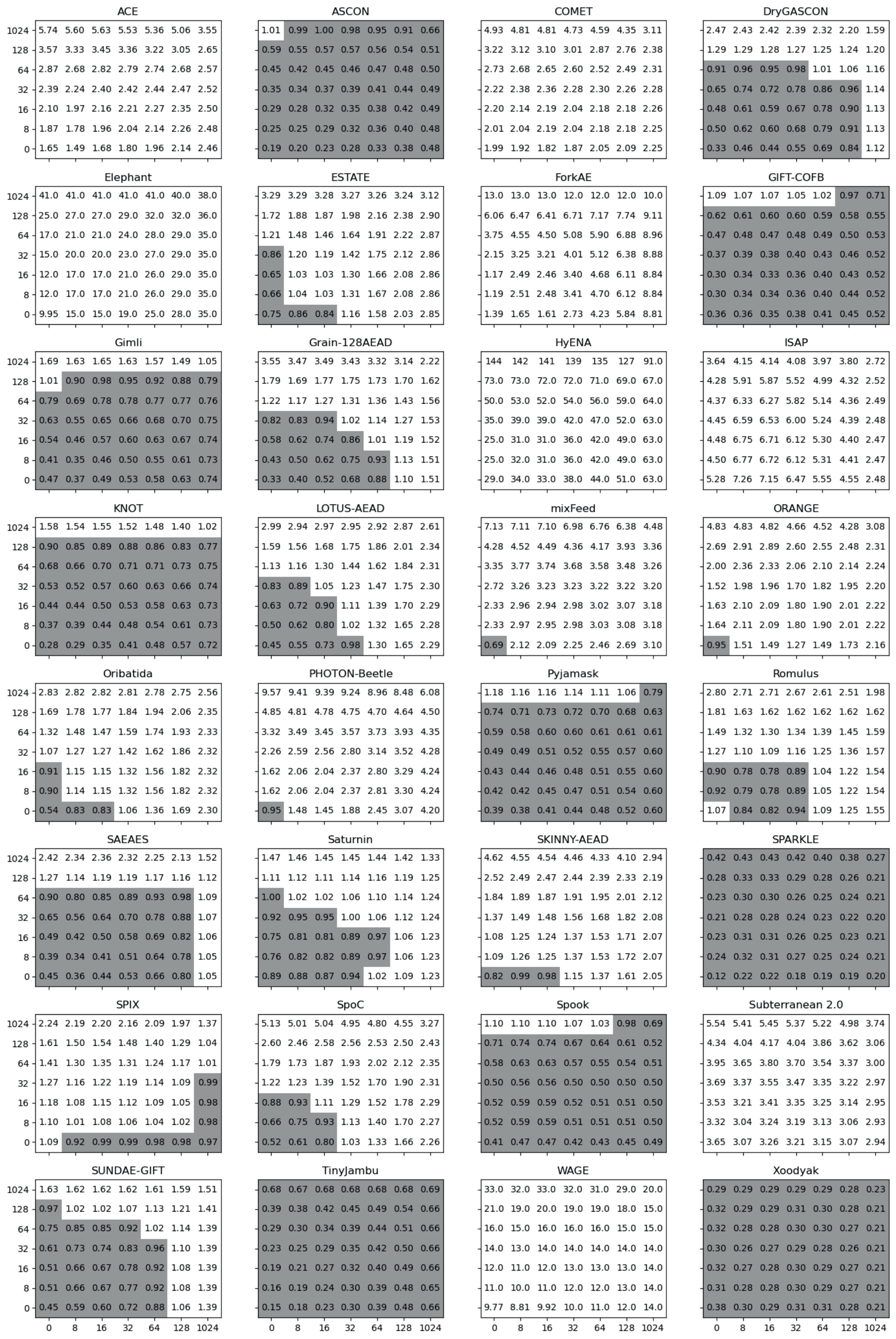
WAGE

	0	8	16	32	64	128	1024
1024	30.0	29.0	29.0	29.0	29.0	28.0	23.0
128	25.0	23.0	24.0	24.0	23.0	22.0	19.0
64	23.0	21.0	22.0	22.0	21.0	20.0	19.0
32	21.0	19.0	20.0	20.0	20.0	19.0	19.0
16	19.0	17.0	19.0	19.0	19.0	19.0	19.0
8	17.0	16.0	17.0	18.0	18.0	18.0	18.0
0	17.0	14.0	16.0	16.0	17.0	17.0	18.0

Xoodyak

	0	8	16	32	64	128	1024
1024	0.39	0.38	0.38	0.39	0.38	0.39	0.39
128	0.46	0.42	0.42	0.45	0.43	0.43	0.40
64	0.54	0.47	0.47	0.50	0.47	0.46	0.41
32	0.55	0.46	0.46	0.50	0.46	0.45	0.41
16	0.62	0.52	0.52	0.54	0.49	0.48	0.41
8	0.62	0.52	0.52	0.54	0.49	0.48	0.41
0	0.84	0.58	0.58	0.60	0.53	0.50	0.41

Figure 9. Relative speeds of the candidates compared to AES-GCM on SAM D21



ACE

	0	8	16	32	64	128	1024
1024	5.74	5.60	5.63	5.53	5.36	5.06	3.55
128	3.57	3.33	3.45	3.36	3.22	3.05	2.65
64	2.87	2.68	2.82	2.79	2.74	2.68	2.57
32	2.39	2.24	2.40	2.42	2.44	2.47	2.52
16	2.10	1.97	2.16	2.21	2.27	2.35	2.50
8	1.87	1.78	1.96	2.04	2.14	2.26	2.48
0	1.65	1.49	1.68	1.80	1.96	2.14	2.46

ASCON

	0	8	16	32	64	128	1024
1024	1.01	0.99	1.00	0.98	0.95	0.91	0.66
128	0.59	0.55	0.57	0.57	0.56	0.54	0.51
64	0.45	0.42	0.45	0.46	0.47	0.48	0.50
32	0.35	0.34	0.37	0.39	0.41	0.44	0.49
16	0.29	0.28	0.32	0.35	0.38	0.42	0.49
8	0.25	0.25	0.29	0.32	0.36	0.40	0.48
0	0.19	0.20	0.23	0.28	0.33	0.38	0.48

COMET

	0	8	16	32	64	128	1024
1024	4.93	4.81	4.81	4.73	4.59	4.35	3.11
128	3.22	3.12	3.10	3.01	2.87	2.76	2.38
64	2.73	2.68	2.65	2.60	2.52	2.49	2.31
32	2.22	2.38	2.36	2.28	2.30	2.26	2.28
16	2.20	2.14	2.19	2.04	2.18	2.18	2.26
8	2.01	2.04	2.19	2.04	2.18	2.18	2.25
0	1.99	1.92	1.82	1.87	2.05	2.09	2.25

DryGASCON

	0	8	16	32	64	128	1024
1024	2.47	2.43	2.42	2.39	2.32	2.20	1.59
128	1.29	1.29	1.28	1.27	1.25	1.24	1.20
64	0.91	0.96	0.95	0.98	1.01	1.06	1.16
32	0.65	0.74	0.72	0.78	0.86	0.96	1.14
16	0.48	0.61	0.59	0.67	0.78	0.90	1.13
8	0.50	0.62	0.60	0.68	0.79	0.91	1.13
0	0.33	0.46	0.44	0.55	0.69	0.84	1.12

Elephant

	0	8	16	32	64	128	1024
1024	41.0	41.0	41.0	41.0	41.0	40.0	38.0
128	25.0	27.0	27.0	29.0	32.0	32.0	36.0
64	17.0	21.0	21.0	24.0	28.0	29.0	35.0
32	15.0	20.0	20.0	23.0	27.0	29.0	35.0
16	12.0	17.0	17.0	21.0	26.0	29.0	35.0
8	12.0	17.0	17.0	21.0	26.0	29.0	35.0
0	9.95	15.0	15.0	19.0	25.0	28.0	35.0

ESTATE

	0	8	16	32	64	128	1024
1024	3.29	3.29	3.28	3.27	3.26	3.24	3.12
128	1.72	1.88	1.87	1.98	2.16	2.38	2.90
64	1.21	1.48	1.46	1.64	1.91	2.22	2.87
32	0.86	1.20	1.19	1.42	1.75	2.12	2.86
16	0.65	1.03	1.03	1.30	1.66	2.08	2.86
8	0.66	1.04	1.03	1.31	1.67	2.08	2.86
0	0.75	0.86	0.84	1.16	1.58	2.03	2.85

ForkAE

	0	8	16	32	64	128	1024
1024	13.0	13.0	13.0	12.0	12.0	12.0	10.0
128	6.06	6.47	6.41	6.71	7.17	7.74	9.11
64	3.75	4.55	4.50	5.08	5.90	6.88	8.96
32	2.15	3.25	3.21	4.01	5.12	6.38	8.88
16	1.17	2.49	2.46	3.40	4.68	6.11	8.84
8	1.19	2.51	2.48	3.41	4.70	6.12	8.84
0	1.39	1.65	1.61	2.73	4.23	5.84	8.81

GIFT-COFB

	0	8	16	32	64	128	1024
1024	1.09	1.07	1.07	1.05	1.02	0.97	0.71
128	0.62	0.61	0.60	0.60	0.59	0.58	0.55
64	0.47	0.48	0.47	0.48	0.49	0.50	0.53
32	0.37	0.39	0.38	0.40	0.43	0.46	0.52
16	0.30	0.34	0.33	0.36	0.40	0.43	0.52
8	0.30	0.34	0.34	0.36	0.40	0.44	0.52
0	0.36	0.36	0.35	0.38	0.41	0.45	0.52

Gimli

	0	8	16	32	64	128	1024
1024	1.69	1.63	1.65	1.63	1.57	1.49	1.05
128	1.01	0.90	0.98	0.95	0.92	0.88	0.79
64	0.79	0.69	0.78	0.78	0.77	0.77	0.76
32	0.63	0.55	0.65	0.66	0.68	0.70	0.75
16	0.54	0.46	0.57	0.60	0.63	0.67	0.74
8	0.41	0.35	0.46	0.50	0.55	0.61	0.73
0	0.47	0.37	0.49	0.53	0.58	0.63	0.74

Grain-128AEAD

	0	8	16	32	64	128	1024
1024	3.55	3.47	3.49	3.43	3.32	3.14	2.22
128	1.79	1.69	1.77	1.75	1.73	1.70	1.62
64	1.22	1.17	1.27	1.31	1.36	1.43	1.56
32	0.82	0.83	0.94	1.02	1.14	1.27	1.53
16	0.58	0.62	0.74	0.86	1.01	1.19	1.52
8	0.43	0.50	0.62	0.75	0.93	1.13	1.51
0	0.33	0.40	0.52	0.68	0.88	1.10	1.51

HyENA

	0	8	16	32	64	128	1024
1024	144	142	141	139	135	127	91.0
128	73.0	73.0	72.0	72.0	71.0	69.0	67.0
64	50.0	53.0	52.0	54.0	56.0	59.0	64.0
32	35.0	39.0	39.0	42.0	47.0	52.0	63.0
16	25.0	31.0	31.0	36.0	42.0	49.0	63.0
8	25.0	32.0	31.0	36.0	42.0	49.0	63.0
0	29.0	34.0	33.0	38.0	44.0	51.0	63.0

ISAP

	0	8	16	32	64	128	1024
1024	3.64	4.15	4.14	4.08	3.97	3.80	2.72
128	4.28	5.91	5.87	5.52	4.99	4.32	2.52
64	4.37	6.33	6.27	5.82	5.14	4.36	2.49
32	4.45	6.59	6.53	6.00	5.24	4.39	2.48
16	4.48	6.75	6.71	6.12	5.30	4.40	2.47
8	4.50	6.77	6.72	6.12	5.31	4.41	2.47
0	5.28	7.26	7.15	6.47	5.55	4.55	2.48

KNOT

	0	8	16	32	64	128	1024
1024	1.58	1.54	1.55	1.52	1.48	1.40	1.02
128	0.90	0.85	0.89	0.88	0.86	0.83	0.77
64	0.68	0.66	0.70	0.71	0.71	0.73	0.75
32	0.53	0.52	0.57	0.60	0.63	0.66	0.74
16	0.44	0.44	0.50	0.53	0.58	0.63	0.73
8	0.37	0.39	0.44	0.48	0.54	0.61	0.73
0	0.28	0.29	0.35	0.41	0.48	0.57	0.72

LOTUS-AEAD

	0	8	16	32	64	128	1024
1024	2.99	2.94	2.97	2.95	2.92	2.87	2.61
128	1.59	1.56	1.68	1.75	1.86	2.01	2.34
64	1.13	1.16	1.30	1.44	1.62	1.84	2.31
32	0.83	0.89	1.05	1.23	1.47	1.75	2.30
16	0.63	0.72	0.90	1.11	1.39	1.70	2.29
8	0.50	0.62	0.80	1.02	1.32	1.65	2.28
0	0.45	0.55	0.73	0.98	1.30	1.65	2.29

mixFeed

	0	8	16	32	64	128	1024
1024	7.13	7.11	7.10	6.98	6.76	6.38	4.48
128	4.28	4.52	4.49	4.36	4.17	3.93	3.36
64	3.35	3.77	3.74	3.68	3.58	3.48	3.26
32	2.72	3.26	3.23	3.23	3.22	3.22	3.20
16	2.33	2.96	2.94	2.98	3.02	3.07	3.18
8	2.33	2.97	2.95	2.98	3.03	3.08	3.18
0	0.69	2.12	2.09	2.25	2.46	2.69	3.10

ORANGE

	0	8	16	32	64	128	1024
1024	4.83	4.83	4.82	4.66	4.52	4.28	3.08
128	2.69	2.91	2.89	2.60	2.55	2.48	2.31
64	2.00	2.36	2.33	2.06	2.10	2.14	2.24
32	1.52	1.98	1.96	1.70	1.82	1.95	2.20
16	1.63	2.10	2.09	1.80	1.90	2.01	2.22
8	1.64	2.11	2.09	1.80	1.90	2.01	2.22
0	0.95	1.51	1.49	1.27	1.49	1.73	2.16

Oribatida

	0	8	16	32	64	128	1024
1024	2.83	2.82	2.82	2.81	2.78	2.75	2.56
128	1.69	1.78	1.77	1.84	1.94	2.06	2.35
64	1.32	1.48	1.47	1.59	1.74	1.93	2.33
32	1.07	1.27	1.27	1.42	1.62	1.86	2.32
16	0.91	1.15	1.15	1.32	1.56	1.82	2.32
8	0.90	1.14	1.15	1.32	1.56	1.82	2.32
0	0.54	0.83	0.83	1.06	1.36	1.69	2.30

PHOTON-Beetle

	0	8	16	32	64	128	1024
1024	9.57	9.41	9.39	9.24	8.96	8.48	6.08
128	4.85	4.81	4.78	4.75	4.70	4.64	4.50
64	3.32	3.49	3.45	3.57	3.73	3.93	4.35
32	2.26	2.59	2.56	2.80	3.14	3.52	4.28
16	1.62	2.06	2.04	2.37	2.80	3.29	4.24
8	1.62	2.06	2.04	2.37	2.81	3.30	4.24
0	0.95	1.48	1.45	1.88	2.45	3.07	4.20

Pyjamask

	0	8	16	32	64	128	1024
1024	1.18	1.16	1.16	1.14	1.11	1.06	0.79
128	0.74	0.71	0.73	0.72	0.70	0.68	0.63
64	0.59	0.58	0.60	0.60	0.61	0.61	0.61
32	0.49	0.49	0.51	0.52	0.55	0.57	0.60
16	0.43	0.44	0.46	0.48	0.51	0.55	0.60
8	0.42	0.42	0.45	0.47	0.51	0.54	0.60
0	0.39	0.38	0.41	0.44	0.48	0.52	0.60

Romulus

	0	8	16	32	64	128	1024
1024	2.80	2.71	2.71	2.67	2.61	2.51	1.98
128	1.81	1.63	1.62	1.62	1.62	1.62	1.62
64	1.49	1.32	1.30	1.34	1.39	1.45	1.59
32	1.27	1.10	1.09	1.16	1.25	1.36	1.57
16	0.90	0.78	0.78	0.89	1.04	1.22	1.54
8	0.92	0.79	0.78	0.89	1.05	1.22	1.54
0	1.07	0.84	0.82	0.94	1.09	1.25	1.55

SAEAES

	0	8	16	32	64	128	1024
1024	2.42	2.34	2.36	2.32	2.25	2.13	1.52
128	1.27	1.14	1.19	1.19	1.17	1.16	1.12
64	0.90	0.80	0.85	0.89	0.93	0.98	1.09
32	0.65	0.56	0.64	0.70	0.78	0.88	1.07
16	0.49	0.42	0.50	0.58	0.69	0.82	1.06
8	0.39	0.34	0.41	0.51	0.64	0.78	1.05
0	0.45	0.36	0.44	0.53	0.66	0.80	1.05

Saturnin

	0	8	16	32	64	128	1024
1024	1.47	1.46	1.45	1.45	1.44	1.42	1.33
128	1.11	1.12	1.11	1.14	1.16	1.19	1.25
64	1.00	1.02	1.02	1.06	1.10	1.14	1.24
32	0.92	0.95	0.95	1.00	1.06	1.12	1.24
16	0.75	0.81	0.81	0.89	0.97	1.06	1.23
8	0.76	0.82	0.82	0.89	0.97	1.06	1.23
0	0.89	0.88	0.87	0.94	1.02	1.09	1.23

SKINNY-AEAD

	0	8	16	32	64	128	1024
1024	4.62	4.55	4.54	4.46	4.33	4.10	2.94
128	2.52	2.49	2.47	2.44	2.39	2.33	2.19
64	1.84	1.89	1.87	1.91	1.95	2.01	2.12
32	1.37	1.49	1.48	1.56	1.68	1.82	2.08
16	1.08	1.25	1.24	1.37	1.53	1.71	2.07
8	1.09	1.26	1.25	1.37	1.53	1.72	2.07
0	0.82	0.99	0.98	1.15	1.37	1.61	2.05

SPARKLE

	0	8	16	32	64	128	1024
1024	0.42	0.43	0.43	0.42	0.40	0.38	0.27
128	0.28	0.33	0.33	0.29	0.28	0.26	0.21
64	0.23	0.30	0.30	0.26	0.25	0.24	0.21
32	0.21	0.28	0.28	0.24	0.23	0.22	0.20
16	0.23	0.31	0.31	0.26	0.25	0.23	0.21
8	0.24	0.32	0.31	0.27	0.25	0.24	0.21
0	0.12	0.22	0.22	0.18	0.19	0.19	0.20

SPIX

	0	8	16	32	64	128	1024
1024	2.24	2.19	2.20	2.16	2.09	1.97	1.37
128	1.61	1.50	1.54	1.48	1.40	1.29	1.04
64	1.41	1.30	1.35	1.31	1.24	1.17	1.01
32	1.27	1.16	1.22	1.19	1.14	1.09	0.99
16	1.18	1.08	1.15	1.12	1.09	1.05	0.98
8	1.10	1.01	1.08	1.06	1.04	1.02	0.98
0	1.09	0.92	0.99	0.99	0.98	0.98	0.97

SpoC

	0	8	16	32	64	128	1024
1024	5.13	5.01	5.04	4.95	4.80	4.55	3.27
128	2.60	2.46	2.58	2.56	2.53	2.50	2.43
64	1.79	1.73	1.87	1.93	2.02	2.12	2.35
32	1.22	1.23	1.39	1.52	1.70	1.90	2.31
16	0.88	0.93	1.11	1.29	1.52	1.78	2.29
8	0.66	0.75	0.93	1.13	1.40	1.70	2.27
0	0.52	0.61	0.80	1.03	1.33	1.66	2.26

Spook

	0	8	16	32	64	128	1024
1024	1.10	1.10	1.10	1.07	1.03	0.98	0.69
128	0.71	0.74	0.74	0.67	0.64	0.61	0.52
64	0.58	0.63	0.63	0.57	0.55	0.54	0.51
32	0.50	0.56	0.56	0.50	0.50	0.50	0.50
16	0.52	0.59	0.59	0.52	0.51	0.51	0.50
8	0.52	0.59	0.59	0.51	0.51	0.51	0.50
0	0.41	0.47	0.47	0.42	0.43	0.45	0.49

Subterranean 2.0

	0	8	16	32	64	128	1024
1024	5.54	5.41	5.45	5.37	5.22	4.98	3.74
128	4.34	4.04	4.17	4.04	3.86	3.62	3.06
64	3.95	3.65	3.80	3.70	3.54	3.37	3.00
32	3.69	3.37	3.55	3.47	3.35	3.22	2.97
16	3.53	3.21	3.41	3.35	3.25	3.14	2.95
8	3.32	3.04	3.24	3.19	3.13	3.06	2.93
0	3.65	3.07	3.26	3.21	3.15	3.07	2.94

SUNDAE-GIFT

	0	8	16	32	64	128	1024
1024	1.63	1.62	1.62	1.62	1.61	1.59	1.51
128	0.97	1.02	1.02	1.07	1.13	1.21	1.41
64	0.75	0.85	0.85	0.92	1.02	1.14	1.39
32	0.61	0.73	0.74	0.83	0.96	1.10	1.39
16	0.51	0.66	0.67	0.78	0.92	1.08	1.39
8	0.51	0.66	0.67	0.77	0.92	1.08	1.39
0	0.45	0.59	0.60	0.72	0.88	1.06	1.39

TinyJambu

	0	8	16	32	64	128	1024
1024	0.68	0.67	0.68	0.68	0.68	0.68	0.69
128	0.39	0.38	0.42	0.45	0.49	0.54	0.66
64	0.29	0.30	0.34	0.39	0.44	0.51	0.66
32	0.23	0.25	0.29	0.35	0.42	0.50	0.66
16	0.19	0.21	0.27	0.32	0.40	0.49	0.66
8	0.16	0.19	0.24	0.30	0.39	0.48	0.65
0	0.15	0.18	0.23	0.30	0.39	0.48	0.66

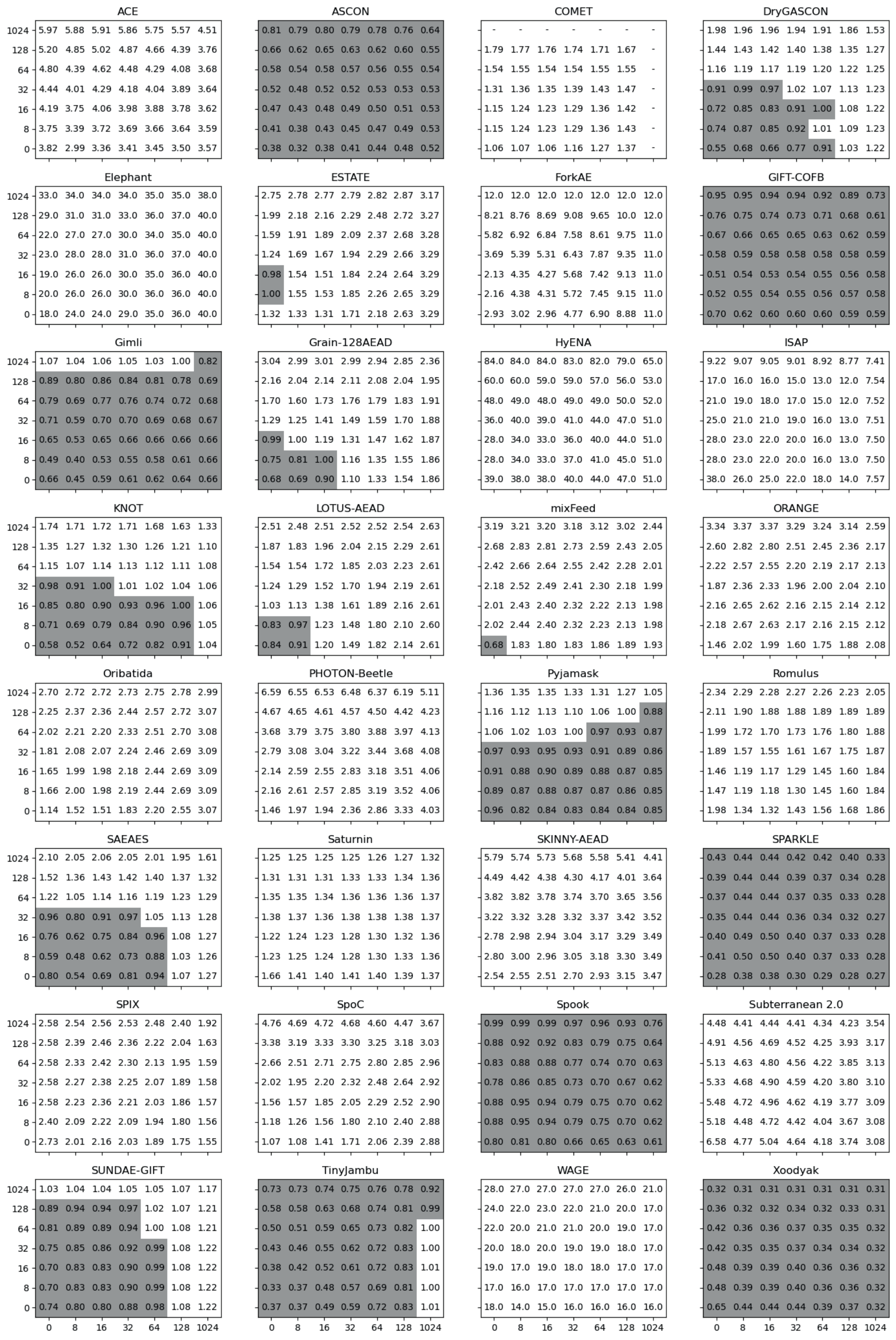
WAGE

	0	8	16	32	64	128	1024
1024	33.0	32.0	33.0	32.0	31.0	29.0	20.0
128	21.0	19.0	20.0	19.0	19.0	18.0	15.0
64	16.0	15.0	16.0	16.0	16.0	15.0	15.0
32	14.0	13.0	14.0	14.0	14.0	14.0	14.0
16	12.0	11.0	12.0	13.0	13.0	13.0	14.0
8	11.0	10.0	11.0	12.0	12.0	13.0	14.0
0	9.77	8.81	9.92	10.0	11.0	12.0	14.0

Xoodyak

	0	8	16	32	64	128	1024
1024	0.29	0.29	0.29	0.29	0.29	0.28	0.23
128	0.32	0.29	0.29	0.31	0.30	0.28	0.21
64	0.32	0.28	0.28	0.30	0.30	0.27	0.21
32	0.30	0.26	0.27	0.29	0.28	0.26	0.21
16	0.32	0.27	0.28	0.30	0.29	0.27	0.21
8	0.31	0.28	0.28	0.30	0.29	0.27	0.21
0	0.38	0.30	0.29	0.31	0.31	0.28	0.21

Figure 10. Relative speeds of the candidates compared to AES-GCM on nRF52840



ACE

	0	8	16	32	64	128	1024
1024	5.97	5.88	5.91	5.86	5.75	5.57	4.51
128	5.20	4.85	5.02	4.87	4.66	4.39	3.76
64	4.80	4.39	4.62	4.48	4.29	4.08	3.68
32	4.44	4.01	4.29	4.18	4.04	3.89	3.64
16	4.19	3.75	4.06	3.98	3.88	3.78	3.62
8	3.75	3.39	3.72	3.69	3.66	3.64	3.59
0	3.82	2.99	3.36	3.41	3.45	3.50	3.57

ASCON

	0	8	16	32	64	128	1024
1024	0.81	0.79	0.80	0.79	0.78	0.76	0.64
128	0.66	0.62	0.65	0.63	0.62	0.60	0.55
64	0.58	0.54	0.58	0.57	0.56	0.55	0.54
32	0.52	0.48	0.52	0.52	0.53	0.53	0.53
16	0.47	0.43	0.48	0.49	0.50	0.51	0.53
8	0.41	0.38	0.43	0.45	0.47	0.49	0.53
0	0.38	0.32	0.38	0.41	0.44	0.48	0.52

COMET

	0	8	16	32	64	128	1024
1024	-	-	-	-	-	-	-
128	1.79	1.77	1.76	1.74	1.71	1.67	-
64	1.54	1.55	1.54	1.54	1.55	1.55	-
32	1.31	1.36	1.35	1.39	1.43	1.47	-
16	1.15	1.24	1.23	1.29	1.36	1.42	-
8	1.15	1.24	1.23	1.29	1.36	1.43	-
0	1.06	1.07	1.06	1.16	1.27	1.37	-

DryGASCON

	0	8	16	32	64	128	1024
1024	1.98	1.96	1.96	1.94	1.91	1.86	1.53
128	1.44	1.43	1.42	1.40	1.38	1.35	1.27
64	1.16	1.19	1.17	1.19	1.20	1.22	1.25
32	0.91	0.99	0.97	1.02	1.07	1.13	1.23
16	0.72	0.85	0.83	0.91	1.00	1.08	1.22
8	0.74	0.87	0.85	0.92	1.01	1.09	1.23
0	0.55	0.68	0.66	0.77	0.91	1.03	1.22

Elephant

	0	8	16	32	64	128	1024
1024	33.0	34.0	34.0	34.0	35.0	35.0	38.0
128	29.0	31.0	31.0	33.0	36.0	37.0	40.0
64	22.0	27.0	27.0	30.0	34.0	35.0	40.0
32	23.0	28.0	28.0	31.0	36.0	37.0	40.0
16	19.0	26.0	26.0	30.0	35.0	36.0	40.0
8	20.0	26.0	26.0	30.0	36.0	36.0	40.0
0	18.0	24.0	24.0	29.0	35.0	36.0	40.0

ESTATE

	0	8	16	32	64	128	1024
1024	2.75	2.78	2.77	2.79	2.82	2.87	3.17
128	1.99	2.18	2.16	2.29	2.48	2.72	3.27
64	1.59	1.91	1.89	2.09	2.37	2.68	3.28
32	1.24	1.69	1.67	1.94	2.29	2.66	3.29
16	0.98	1.54	1.51	1.84	2.24	2.64	3.29
8	1.00	1.55	1.53	1.85	2.26	2.65	3.29
0	1.32	1.33	1.31	1.71	2.18	2.63	3.29

ForkAE

	0	8	16	32	64	128	1024
1024	12.0	12.0	12.0	12.0	12.0	12.0	12.0
128	8.21	8.76	8.69	9.08	9.65	10.0	12.0
64	5.82	6.92	6.84	7.58	8.61	9.75	11.0
32	3.69	5.39	5.31	6.43	7.87	9.35	11.0
16	2.13	4.35	4.27	5.68	7.42	9.13	11.0
8	2.16	4.38	4.31	5.72	7.45	9.15	11.0
0	2.93	3.02	2.96	4.77	6.90	8.88	11.0

GIFT-COFB

	0	8	16	32	64	128	1024
1024	0.95	0.95	0.94	0.94	0.92	0.89	0.73
128	0.76	0.75	0.74	0.73	0.71	0.68	0.61
64	0.67	0.66	0.65	0.65	0.63	0.62	0.59
32	0.58	0.59	0.58	0.58	0.58	0.58	0.59
16	0.51	0.54	0.53	0.54	0.55	0.56	0.58
8	0.52	0.55	0.54	0.55	0.56	0.57	0.58
0	0.70	0.62	0.60	0.60	0.60	0.59	0.59

Gimli

	0	8	16	32	64	128	1024
1024	1.07	1.04	1.06	1.05	1.03	1.00	0.82
128	0.89	0.80	0.86	0.84	0.81	0.78	0.69
64	0.79	0.69	0.77	0.76	0.74	0.72	0.68
32	0.71	0.59	0.70	0.70	0.69	0.68	0.67
16	0.65	0.53	0.65	0.66	0.66	0.66	0.66
8	0.49	0.40	0.53	0.55	0.58	0.61	0.66
0	0.66	0.45	0.59	0.61	0.62	0.64	0.66

Grain-128AEAD

	0	8	16	32	64	128	1024
1024	3.04	2.99	3.01	2.99	2.94	2.85	2.36
128	2.16	2.04	2.14	2.11	2.08	2.04	1.95
64	1.70	1.60	1.73	1.76	1.79	1.83	1.91
32	1.29	1.25	1.41	1.49	1.59	1.70	1.88
16	0.99	1.00	1.19	1.31	1.47	1.62	1.87
8	0.75	0.81	1.00	1.16	1.35	1.55	1.86
0	0.68	0.69	0.90	1.10	1.33	1.54	1.86

HyENA

	0	8	16	32	64	128	1024
1024	84.0	84.0	84.0	83.0	82.0	79.0	65.0
128	60.0	60.0	59.0	59.0	57.0	56.0	53.0
64	48.0	49.0	48.0	49.0	49.0	50.0	52.0
32	36.0	40.0	39.0	41.0	44.0	47.0	51.0
16	28.0	34.0	33.0	36.0	40.0	44.0	51.0
8	28.0	34.0	33.0	37.0	41.0	45.0	51.0
0	39.0	38.0	38.0	40.0	44.0	47.0	51.0

ISAP

	0	8	16	32	64	128	1024
1024	9.22	9.07	9.05	9.01	8.92	8.77	7.41
128	17.0	16.0	16.0	15.0	13.0	12.0	7.54
64	21.0	19.0	18.0	17.0	15.0	12.0	7.52
32	25.0	21.0	21.0	19.0	16.0	13.0	7.51
16	28.0	23.0	22.0	20.0	16.0	13.0	7.50
8	28.0	23.0	22.0	20.0	16.0	13.0	7.50
0	38.0	26.0	25.0	22.0	18.0	14.0	7.57

KNOT

	0	8	16	32	64	128	1024
1024	1.74	1.71	1.72	1.71	1.68	1.63	1.33
128	1.35	1.27	1.32	1.30	1.26	1.21	1.10
64	1.15	1.07	1.14	1.13	1.12	1.11	1.08
32	0.98	0.91	1.00	1.01	1.02	1.04	1.06
16	0.85	0.80	0.90	0.93	0.96	1.00	1.06
8	0.71	0.69	0.79	0.84	0.90	0.96	1.05
0	0.58	0.52	0.64	0.72	0.82	0.91	1.04

LOTUS-AEAD

	0	8	16	32	64	128	1024
1024	2.51	2.48	2.51	2.52	2.52	2.54	2.63
128	1.87	1.83	1.96	2.04	2.15	2.29	2.61
64	1.54	1.54	1.72	1.85	2.03	2.23	2.61
32	1.24	1.29	1.52	1.70	1.94	2.19	2.61
16	1.03	1.13	1.38	1.61	1.89	2.16	2.61
8	0.83	0.97	1.23	1.48	1.80	2.10	2.60
0	0.84	0.91	1.20	1.49	1.82	2.14	2.61

mixFeed

	0	8	16	32	64	128	1024
1024	3.19	3.21	3.20	3.18	3.12	3.02	2.44
128	2.68	2.83	2.81	2.73	2.59	2.43	2.05
64	2.42	2.66	2.64	2.55	2.42	2.28	2.01
32	2.18	2.52	2.49	2.41	2.30	2.18	1.99
16	2.01	2.43	2.40	2.32	2.22	2.13	1.98
8	2.02	2.44	2.40	2.32	2.23	2.13	1.98
0	0.68	1.83	1.80	1.83	1.86	1.89	1.93

ORANGE

	0	8	16	32	64	128	1024
1024	3.34	3.37	3.37	3.29	3.24	3.14	2.59
128	2.60	2.82	2.80	2.51	2.45	2.36	2.17
64	2.22	2.57	2.55	2.20	2.19	2.17	2.13
32	1.87	2.36	2.33	1.96	2.00	2.04	2.10
16	2.16	2.65	2.62	2.16	2.15	2.14	2.12
8	2.18	2.67	2.63	2.17	2.16	2.15	2.12
0	1.46	2.02	1.99	1.60	1.75	1.88	2.08

Oribatida

	0	8	16	32	64	128	1024
1024	2.70	2.72	2.72	2.73	2.75	2.78	2.99
128	2.25	2.37	2.36	2.44	2.57	2.72	3.07
64	2.02	2.21	2.20	2.33	2.51	2.70	3.08
32	1.81	2.08	2.07	2.24	2.46	2.69	3.09
16	1.65	1.99	1.98	2.18	2.44	2.69	3.09
8	1.66	2.00	1.98	2.19	2.44	2.69	3.09
0	1.14	1.52	1.51	1.83	2.20	2.55	3.07

PHOTON-Beetle

	0	8	16	32	64	128	1024
1024	6.59	6.55	6.53	6.48	6.37	6.19	5.11
128	4.67	4.65	4.61	4.57	4.50	4.42	4.23
64	3.68	3.79	3.75	3.80	3.88	3.97	4.13
32	2.79	3.08	3.04	3.22	3.44	3.68	4.08
16	2.14	2.59	2.55	2.83	3.18	3.51	4.06
8	2.16	2.61	2.57	2.85	3.19	3.52	4.06
0	1.46	1.97	1.94	2.36	2.86	3.33	4.03

Pyjamask

	0	8	16	32	64	128	1024
1024	1.36	1.35	1.35	1.33	1.31	1.27	1.05
128	1.16	1.12	1.13	1.10	1.06	1.00	0.88
64	1.06	1.02	1.03	1.00	0.97	0.93	0.87
32	0.97	0.93	0.95	0.93	0.91	0.89	0.86
16	0.91	0.88	0.90	0.89	0.88	0.87	0.85
8	0.89	0.87	0.88	0.87	0.87	0.86	0.85
0	0.96	0.82	0.84	0.83	0.84	0.84	0.85

Romulus

	0	8	16	32	64	128	1024
1024	2.34	2.29	2.28	2.27	2.26	2.23	2.05
128	2.11	1.90	1.88	1.88	1.89	1.89	1.89
64	1.99	1.72	1.70	1.73	1.76	1.80	1.88
32	1.89	1.57	1.55	1.61	1.67	1.75	1.87
16	1.46	1.19	1.17	1.29	1.45	1.60	1.84
8	1.47	1.19	1.18	1.30	1.45	1.60	1.84
0	1.98	1.34	1.32	1.43	1.56	1.68	1.86

SAEAES

	0	8	16	32	64	128	1024
1024	2.10	2.05	2.06	2.05	2.01	1.95	1.61
128	1.52	1.36	1.43	1.42	1.40	1.37	1.32
64	1.22	1.05	1.14	1.16	1.19	1.23	1.29
32	0.96	0.80	0.91	0.97	1.05	1.13	1.28
16	0.76	0.62	0.75	0.84	0.96	1.08	1.27
8	0.59	0.48	0.62	0.73	0.88	1.03	1.26
0	0.80	0.54	0.69	0.81	0.94	1.07	1.27

Saturnin

	0	8	16	32	64	128	1024
1024	1.25	1.25	1.25	1.25	1.26	1.27	1.32
128	1.31	1.31	1.31	1.33	1.33	1.34	1.36
64	1.35	1.35	1.34	1.36	1.36	1.36	1.37
32	1.38	1.37	1.36	1.38	1.38	1.38	1.37
16	1.22	1.24	1.23	1.28	1.30	1.32	1.36
8	1.23	1.25	1.24	1.28	1.30	1.33	1.36
0	1.66	1.41	1.40	1.41	1.40	1.39	1.37

SKINNY-AEAD

	0	8	16	32	64	128	1024
1024	5.79	5.74	5.73	5.68	5.58	5.41	4.41
128	4.49	4.42	4.38	4.30	4.17	4.01	3.64
64	3.82	3.82	3.78	3.74	3.70	3.65	3.56
32	3.22	3.32	3.28	3.32	3.37	3.42	3.52
16	2.78	2.98	2.94	3.04	3.17	3.29	3.49
8	2.80	3.00	2.96	3.05	3.18	3.30	3.49
0	2.54	2.55	2.51	2.70	2.93	3.15	3.47

SPARKLE

	0	8	16	32	64	128	1024
1024	0.43	0.44	0.44	0.42	0.42	0.40	0.33
128	0.39	0.44	0.44	0.39	0.37	0.34	0.28
64	0.37	0.44	0.44	0.37	0.35	0.33	0.28
32	0.35	0.44	0.44	0.36	0.34	0.32	0.27
16	0.40	0.49	0.50	0.40	0.37	0.33	0.28
8	0.41	0.50	0.50	0.40	0.37	0.33	0.28
0	0.28	0.38	0.38	0.30	0.29	0.28	0.27

SPIX

	0	8	16	32	64	128	1024
1024	2.58	2.54	2.56	2.53	2.48	2.40	1.92
128	2.58	2.39	2.46	2.36	2.22	2.04	1.63
64	2.58	2.33	2.42	2.30	2.13	1.95	1.59
32	2.58	2.27	2.38	2.25	2.07	1.89	1.58
16	2.58	2.23	2.36	2.21	2.03	1.86	1.57
8	2.40	2.09	2.22	2.09	1.94	1.80	1.56
0	2.73	2.01	2.16	2.03	1.89	1.75	1.55

SpoC

	0	8	16	32	64	128	1024
1024	4.76	4.69	4.72	4.68	4.60	4.47	3.67
128	3.38	3.19	3.33	3.30	3.25	3.18	3.03
64	2.66	2.51	2.71	2.75	2.80	2.85	2.96
32	2.02	1.95	2.20	2.32	2.48	2.64	2.92
16	1.56	1.57	1.85	2.05	2.29	2.52	2.90
8	1.18	1.26	1.56	1.80	2.10	2.40	2.88
0	1.07	1.08	1.41	1.71	2.06	2.39	2.88

Spook

	0	8	16	32	64	128	1024
1024	0.99	0.99	0.99	0.97	0.96	0.93	0.76
128	0.88	0.92	0.92	0.83	0.79	0.75	0.64
64	0.83	0.88	0.88	0.77	0.74	0.70	0.63
32	0.78	0.86	0.85	0.73	0.70	0.67	0.62
16	0.88	0.95	0.94	0.79	0.75	0.70	0.62
8	0.88	0.95	0.94	0.79	0.75	0.70	0.62
0	0.80	0.81	0.80	0.66	0.65	0.63	0.61

Subterranean 2.0

	0	8	16	32	64	128	1024
1024	4.48	4.41	4.44	4.41	4.34	4.23	3.54
128	4.91	4.56	4.69	4.52	4.25	3.93	3.17
64	5.13	4.63	4.80	4.56	4.22	3.85	3.13
32	5.33	4.68	4.90	4.59	4.20	3.80	3.10
16	5.48	4.72	4.96	4.62	4.19	3.77	3.09
8	5.18	4.48	4.72	4.42	4.04	3.67	3.08
0	6.58	4.77	5.04	4.64	4.18	3.74	3.08

SUNDAE-GIFT

	0	8	16	32	64	128	1024
1024	1.03	1.04	1.04	1.05	1.05	1.07	1.17
128	0.89	0.94	0.94	0.97	1.02	1.07	1.21
64	0.81	0.89	0.89	0.94	1.00	1.08	1.21
32	0.75	0.85	0.86	0.92	0.99	1.08	1.22
16	0.70	0.83	0.83	0.90	0.99	1.08	1.22
8	0.70	0.83	0.83	0.90	0.99	1.08	1.22
0	0.74	0.80	0.80	0.88	0.98	1.08	1.22

TinyJambu

	0	8	16	32	64	128	1024
1024	0.73	0.73	0.74	0.75	0.76	0.78	0.92
128	0.58	0.58	0.63	0.68	0.74	0.81	0.99
64	0.50	0.51	0.59	0.65	0.73	0.82	1.00
32	0.43	0.46	0.55	0.62	0.72	0.83	1.00
16	0.38	0.42	0.52	0.61	0.72	0.83	1.01
8	0.33	0.37	0.48	0.57	0.69	0.81	1.00
0	0.37	0.37	0.49	0.59	0.72	0.83	1.01

WAGE

	0	8	16	32	64	128	1024
1024	28.0	27.0	27.0	27.0	27.0	26.0	21.0
128	24.0	22.0	23.0	22.0	21.0	20.0	17.0
64	22.0	20.0	21.0	21.0	20.0	19.0	17.0
32	20.0	18.0	20.0	19.0	19.0	18.0	17.0
16	19.0	17.0	19.0	18.0	18.0	17.0	17.0
8	17.0	16.0	17.0	17.0	17.0	17.0	17.0
0	18.0	14.0	15.0	16.0	16.0	16.0	16.0

Xoodyak

	0	8	16	32	64	128	1024
1024	0.32	0.31	0.31	0.31	0.31	0.31	0.31
128	0.36	0.32	0.32	0.34	0.32	0.33	0.31
64	0.42	0.36	0.36	0.37	0.35	0.35	0.32
32	0.42	0.35	0.35	0.37	0.34	0.34	0.32
16	0.48	0.39	0.39	0.40	0.36	0.36	0.32
8	0.48	0.39	0.39	0.40	0.36	0.36	0.32
0	0.65	0.44	0.44	0.44	0.39	0.37	0.32

Figure 11. Relative speeds of the candidates compared to AES-GCM on PIC32MX

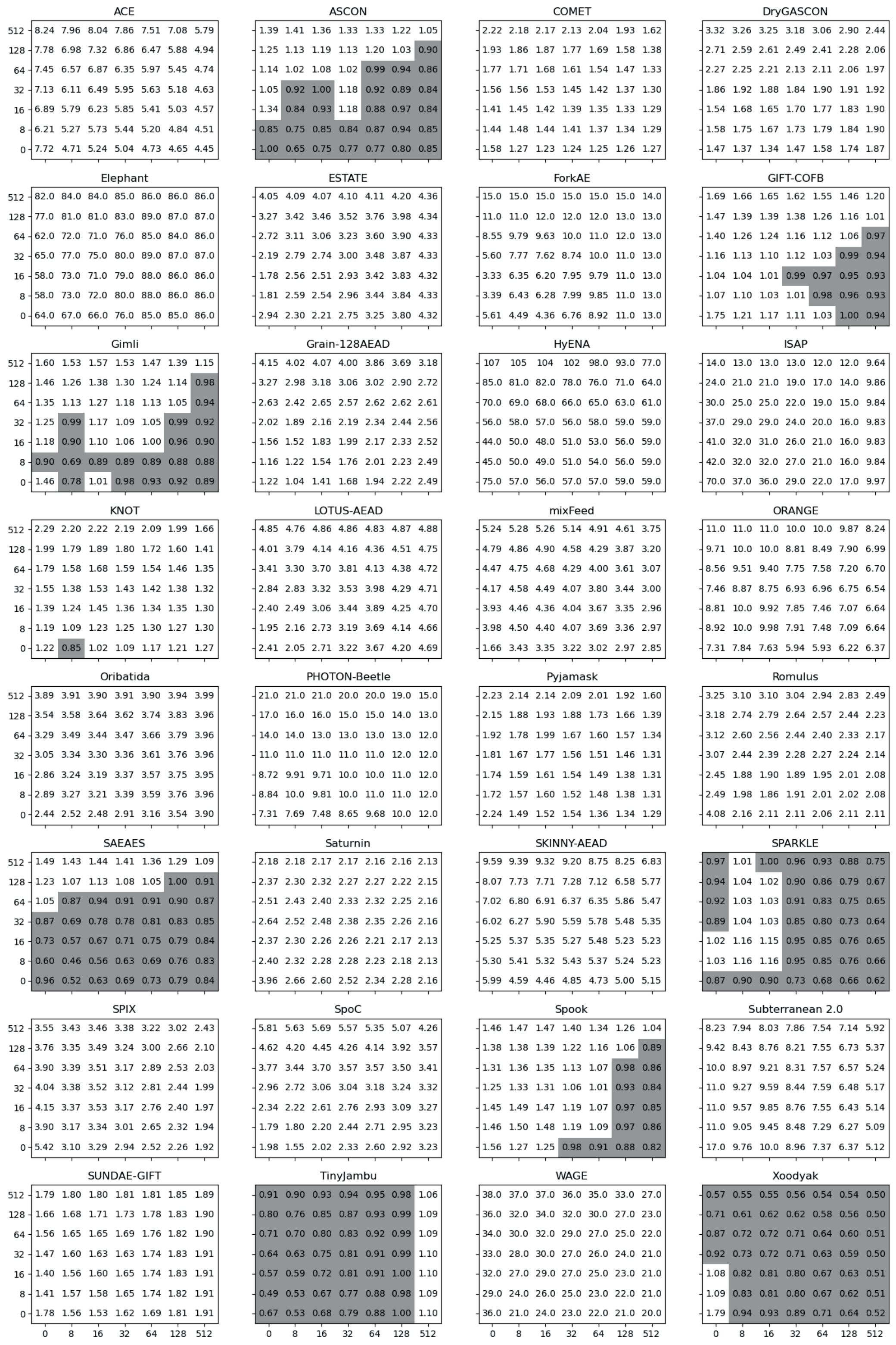


Figure 12. Relative speeds of the candidates compared to AES-GCM on ESP8266