

Final Steps of the NIST Lightweight Cryptography Standardization

Meltem Sönmez Turan

This Talk will cover

NIST

an overview of the NIST lightweight cryptography standardization

an update on standardization of Ascon family



- Part of US Department of Commerce
- Founded in 1901, known as the National Bureau of Standards (NBS) prior to 1988

MISSION

to promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve our quality of life.



3,400+
FEDERAL
EMPLOYEES



3,500+
ASSOCIATES



5
NOBEL PRIZES

Laboratory Programs → Information Technology Lab → Computer Security Division

Developing Crypto Standards

- International “competitions” e.g., AES, SHA-3, PQC, Lightweight Crypto
- Adoption of existing standards e.g., RSA, HMAC
- Open call for proposals: e.g., block cipher modes of operations

CSD Publications

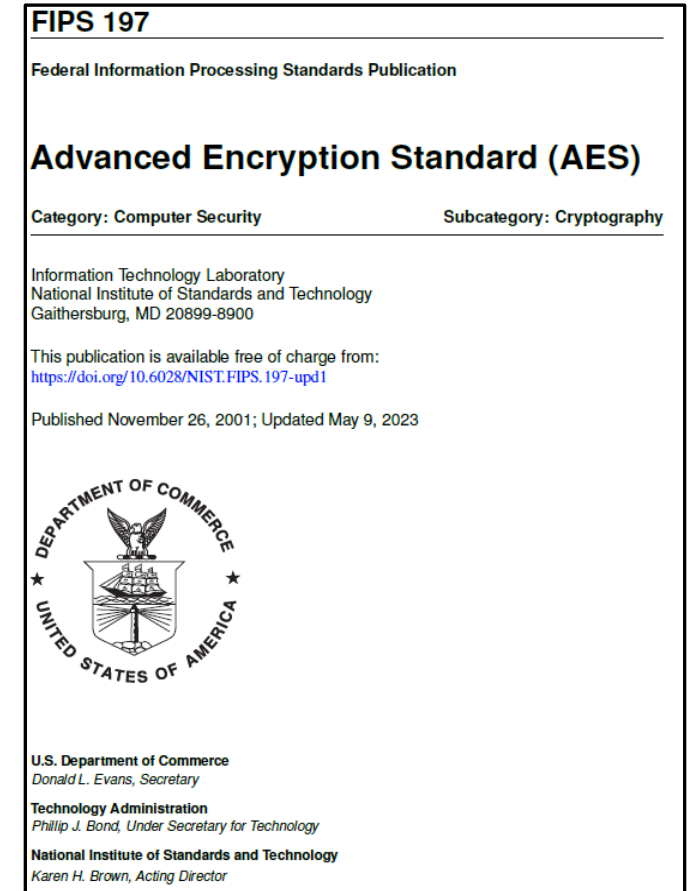
- Federal Information Processing Standards (FIPS): Specify approved crypto standards.
- NIST Special Publications (SPs): Guidelines, technical specifications, recommendations etc.
- NIST Internal or Interagency Reports (IR): Reports of research findings.

Principles

Transparency, openness, balance, integrity, technical merit, global acceptability, usability, continuous improvement, innovation etc.

Advanced Encryption Standard (AES)

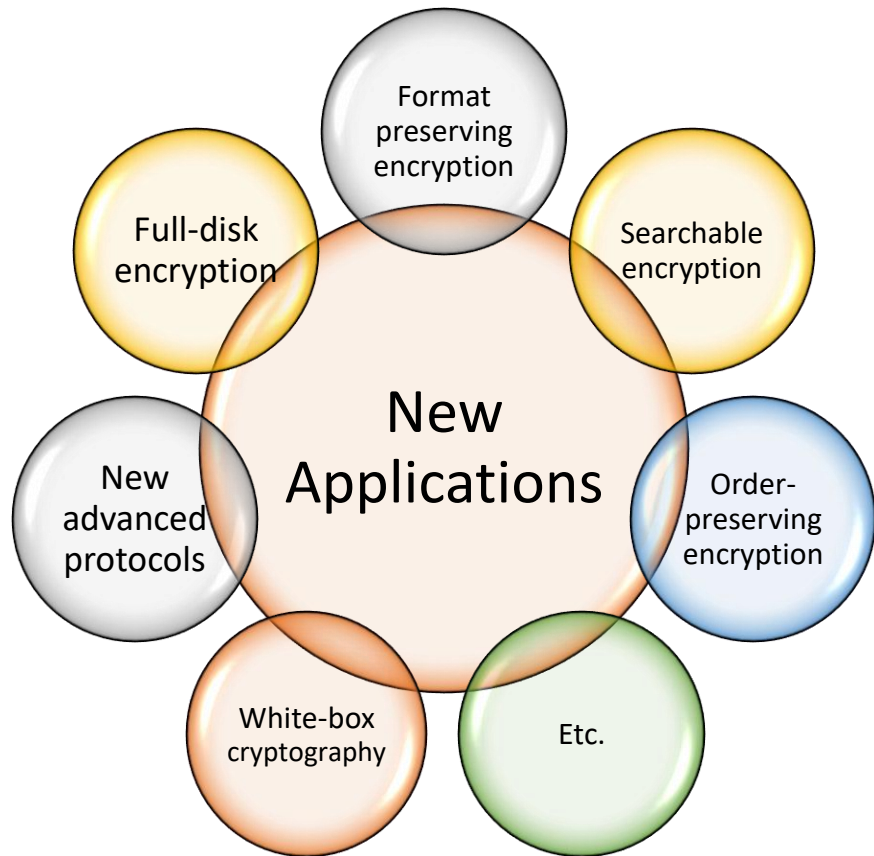
- FIPS 197 Advanced Encryption Standard – Published in 2001.
- Reviewed¹ after 20 years and updated in May 2023.
- Widely adopted, with significant impact on economy²
- Instantiated with a *mode of operation* from SP 800-38 series (e.g., CBC, OFB, CBC, GCM, ...)



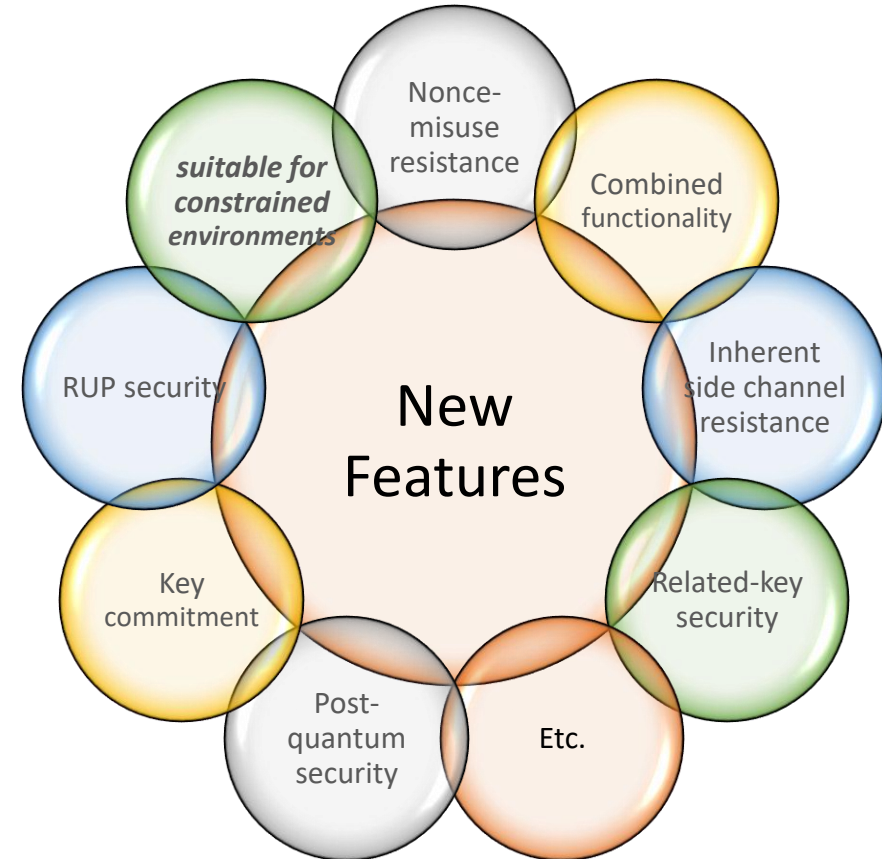
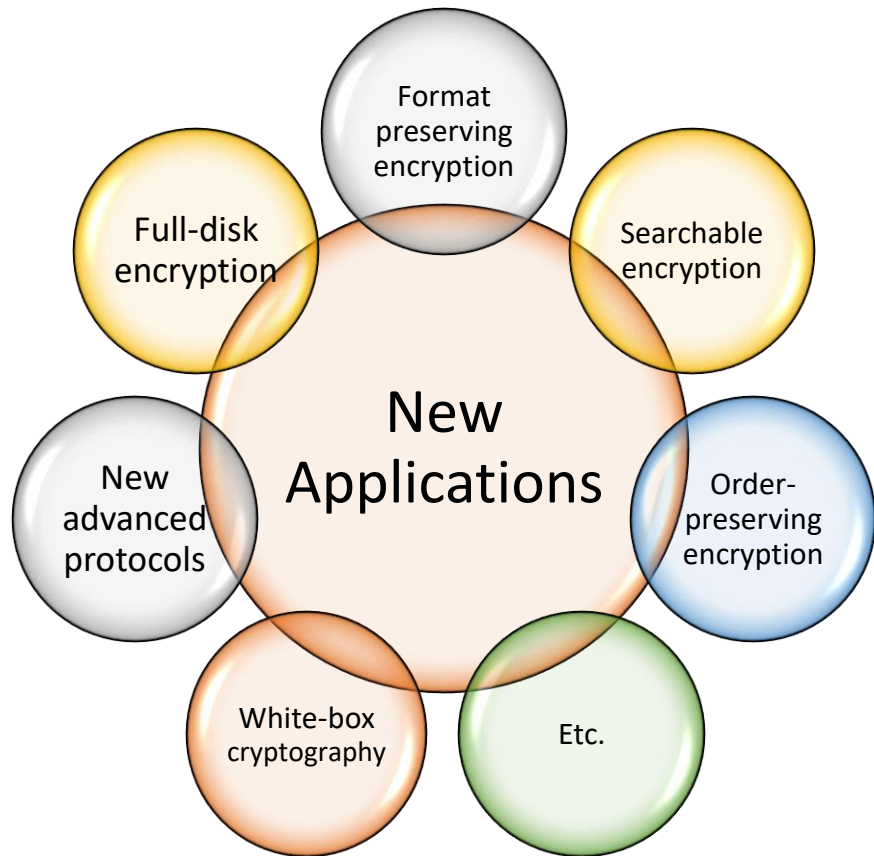
1. NIST IR 8319 & Publication Reviews <https://csrc.nist.gov/projects/crypto-publication-review-project/completed-reviews>
2. Smid, Development of the Advanced Encryption Standard, 2021
3. Leech et al., *The Economic Impacts of the Advanced Encryption Standard*, 2018

Why do we need more symmetric-key primitives?

Why do we need more symmetric-key primitives?



Why do we need more symmetric-key primitives?





- Constrained Devices
- e.g., RFID tags, sensors, IoT devices



- New Applications
- e.g., home automation, healthcare, smart city



- Private Information
- e.g., location, health data



- Lack of Cryptography Standards
- NIST crypto standards are optimized for general-purpose computers.



PROCESS

Public competition-like process with multiple rounds like AES, SHA3 and PQC standardization



GOAL

Develop new guidelines, recommendations and standards optimized for constrained devices



SCOPE

Authenticated Encryption and (optional) hashing for constrained software and hardware environments



Initial Phase
(July 2015 – August 2018)



Submission Call
(August 2018 – April 2019)



Round 1
(April 2019 – August 2019)



Round 2
(August 2019 – March 2021)



Final Round
(March 2021 – February 2023)



Initial Phase
(July 2015 – August 2018)



Submission Call
(August 2018 – April 2019)



Round 1
(April 2019 – August 2019)



Round 2
(August 2019 – March 2021)



Final Round
(March 2021 – February 2023)

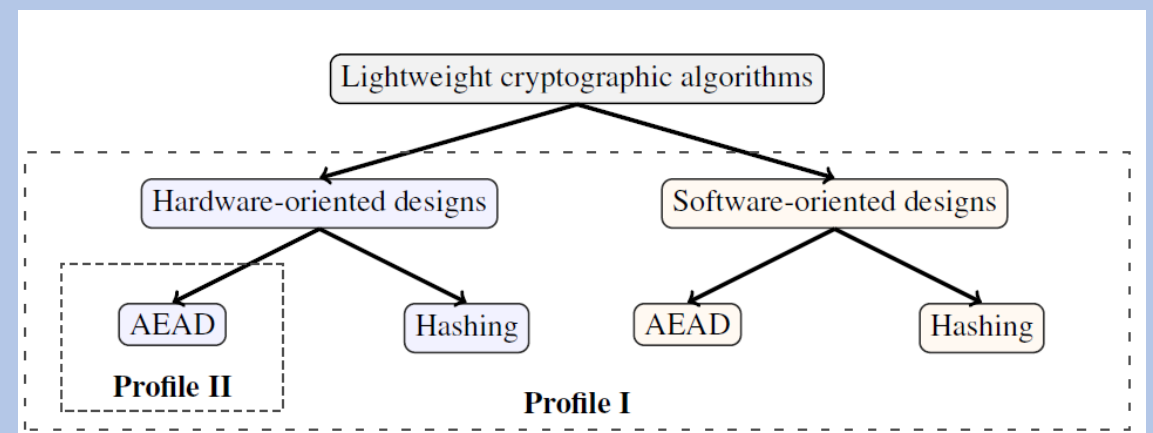
Workshops:

- First Lightweight Cryptography Workshop
July 20 – 21, 2015
- Second Lightweight Cryptography Workshop
October 17 – 18, 2016

to get feedback on target applications, industry need, requirements, etc.

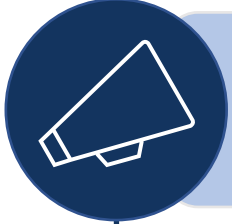
Publications:

- NISTIR 8114 *Report on Lightweight Cryptography*
- (White paper, retired) *Profiles for the Lightweight Cryptography Standardization Process*





Initial Phase
(July 2015 – August 2018)



Submission Call
(August 2018 – April 2019)



Round 1
(April 2019 – August 2019)



Round 2
(August 2019 – March 2021)



Final Round
(March 2021 – February 2023)

In August 2018, NIST published '*Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process*'.

Submission deadline: February 2019

Security requirements

At least 112-bit security level for messages up to 2^{50} bytes, (nonce respecting). Key size at least 128 bits.

Design requirements

Perform better than NIST standards (AES-GCM, SHA-2), optimized for short messages etc.

Implementation requirements

Reference and optimized implementation compatible with API etc.



Initial Phase
(July 2015 – August 2018)



Submission Call
(August 2018 – April 2019)



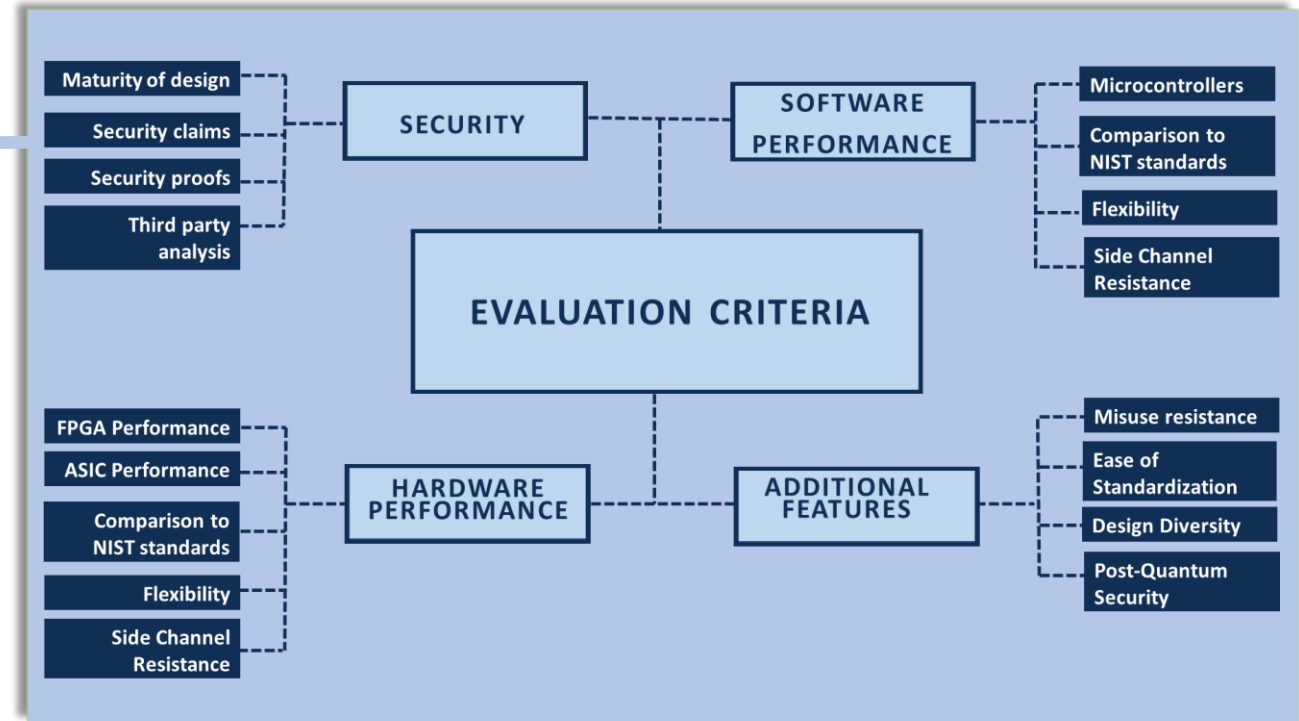
Round 1
(April 2019 – August 2019)



Round 2
(August 2019 – March 2021)



Final Round
(March 2021 – February 2023)





Initial Phase
(July 2015 – August 2018)



Submission Call
(August 2018 – April 2019)



Round 1
(April 2019 – August 2019)



Round 2
(August 2019 – March 2021)



Final Round
(March 2021 – February 2023)

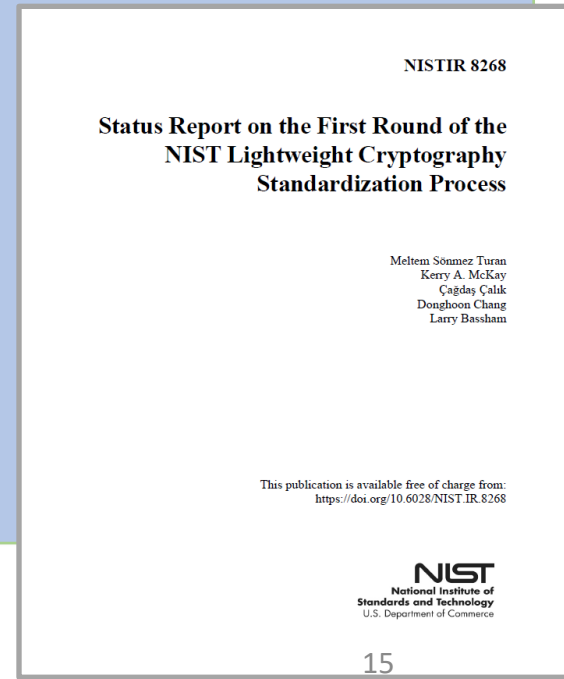
Around 4 months

56 First-round candidates

Evaluation of the candidates were done based on their security

- e.g., distinguishing attacks, practical tag forgeries, domain separation issues, new designs with no third-party analysis etc.

NIST IR 8268 explains how 32 candidates (out of 56) were selected to move forward to the second round.



NIST IR 8268

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

Meltem Sonmez Turan
Kerry A. McKay
Cağdas Çalik
Donghoon Chang
Larry Bassham

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.IR.8268>

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



Initial Phase
(July 2015 – August 2018)



Submission Call
(August 2018 – April 2019)



Round 1
(April 2019 – August 2019)



Round 2
(August 2019 – March 2021)



Final Round
(March 2021 – February 2023)

Around 20 months

32 Second-round candidates

Workshops:

- Third Lightweight Cryptography Workshop
November 4 – 6, 2019
- Fourth Lightweight Cryptography Workshop 2016
October 19 – 21, 2020

NIST IR 8369 explains how
10 finalists were selected
to move forward to
the final round.

NISTIR 8369

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

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

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



Initial Phase
(July 2015 – August 2018)



Submission Call
(August 2018 – April 2019)



Round 1
(April 2019 – August 2019)



Round 2
(August 2019 – March 2021)



Final Round
(March 2021 – February 2023)

Evaluation of ten finalists took about two years.

Ascon

Photon-Beetle

Elephant

Romulus

GIFT-COFB

Sparkle

Grain-128AEAD

TinyJambu

ISAP

Xoodyak

Fair evaluation of finalists is challenging:

- Assigning different weights for different criteria
- Different security claims, different functionality, attacks with different complexities etc.
- Limited resources (not all algorithms got the same attention from the crypto community) for security analysis and benchmarking.

Decision relied on publicly available analysis and benchmarking results.

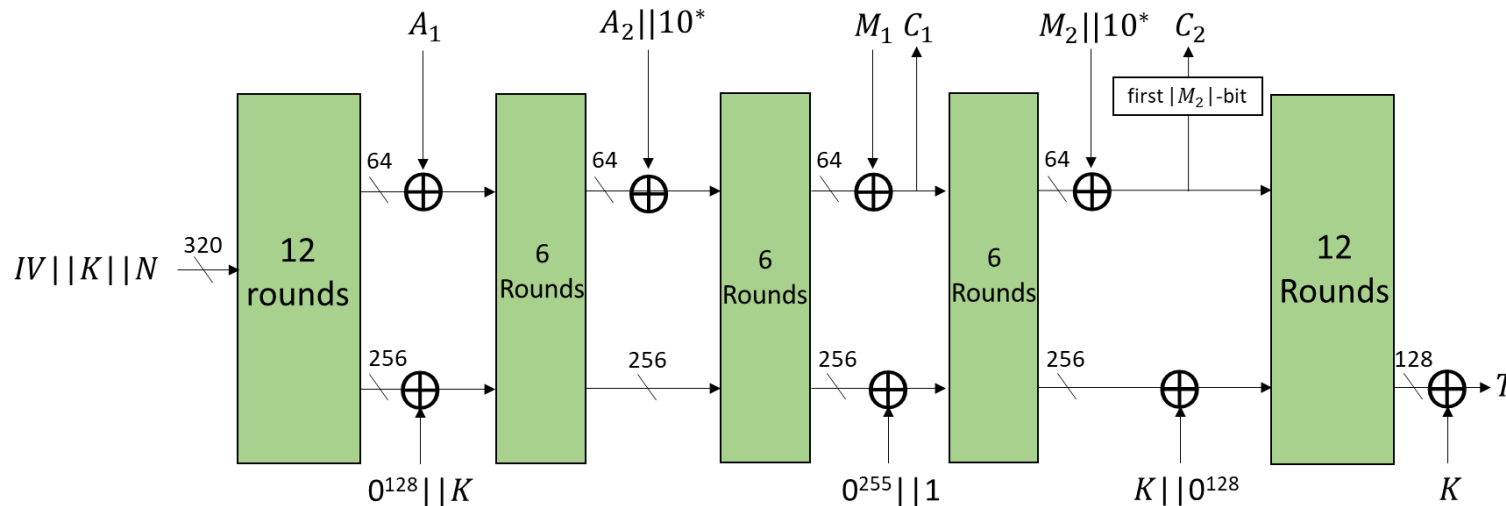
<i>Finalists</i>	<i>Variant</i>	<i>Building Block</i>	<i>Mode</i>	<i>Key size</i>	<i>Nonce Size</i>	<i>Tag Size</i>
ASCON	ASCON-128	ASCON Permutation	MonkeyDuplex	128	128	128
	ASCON-128a			128	128	128
	ASCON-80pq			160	128	128
Elephant	Dumbo	Spongent- π [160]	Encrypt-then-MAC	128	96	64
	Jumbo	Spongent- π [176]		128	96	64
	Delirium	KECCAK- f [200]		128	96	128
GIFT-COFB	GIFT-COFB	GIFT-128	Combined Feedback	128	128	128
Grain-128AEAD	Grain-128AEAD	Feedback shift register	Encrypt-and-MAC	128	96	64
ISAP	ISAP-A-128a	ASCON Permutation	Encrypt-then-MAC	128	128	128
	ISAP-K-128a	KECCAK- f [400]		128	128	128
	ISAP-A-128	ASCON Permutation		128	128	128
	ISAP-K-128	KECCAK- f [400]		128	128	128
PHOTON-Beetle	PHOTON-Beetle-AEAD[128]	PHOTON ₂₅₆ Permutation	Sponge with	128	128	128
	PHOTON-Beetle-AEAD[32]		Combined Feedback	128	128	128
Romulus	Romulus-N	Skinny-128-384+ Tweakable Block Cipher	Combined Feedback	128	128	128
	Romulus-M		MAC-then-Encrypt	128	128	128
	Romulus-T		Encrypt-then-MAC	128	128	128
SPARKLE	SCHWAEMM256-128	SPARKLE ₃₈₄	Sponge with Combined Feedback	128	256	128
	SCHWAEMM128-128	SPARKLE ₂₅₆		128	128	128
	SCHWAEMM192-192	SPARKLE ₃₈₄		192	192	192
	SCHWAEMM256-256	SPARKLE ₅₁₂		256	256	256
TinyJAMBU	TinyJAMBU-128	Keyed Permutation	Sponge	128	96	64
	TinyJAMBU-192			192	96	64
	TinyJAMBU-256			256	96	64
Xoodyak	Xoodyakv1	Xoodoo Permutation	Sponge-variant Cyclist	128	128	128

<i>Finalists</i>	<i>Variant</i>	<i>Building Block</i>	<i>Mode</i>	<i>Digest size</i>
ASCON	ASCON-Hash	ASCON Permutation	Sponge	256
	ASCON-Hasha			256
PHOTON-Beetle	PHOTON-Beetle-Hash[32]	PHOTON ₂₅₆ Permutation	Sponge	256
Romulus	Romulus-H	Skinny-128-384+	MDPH ¹	256
SPARKLE	ESCH256	SPARKLE ₃₈₄	Sponge	256
	ESCH384	SPARKLE ₅₁₂		384
Xoodyak	Xoodyak	Xoodoo Permutation	Sponge	256

ASCON

- Permutation-based (320-bit) AEAD and hashing scheme (fixed or variable output length)
- AEAD: MonkeyDuplex mode with keyed initialization and finalization, Hash: Sponge
- No design tweak, new variant added in the final round
- Included in the final portfolio of CAESAR for lightweight authenticated encryption

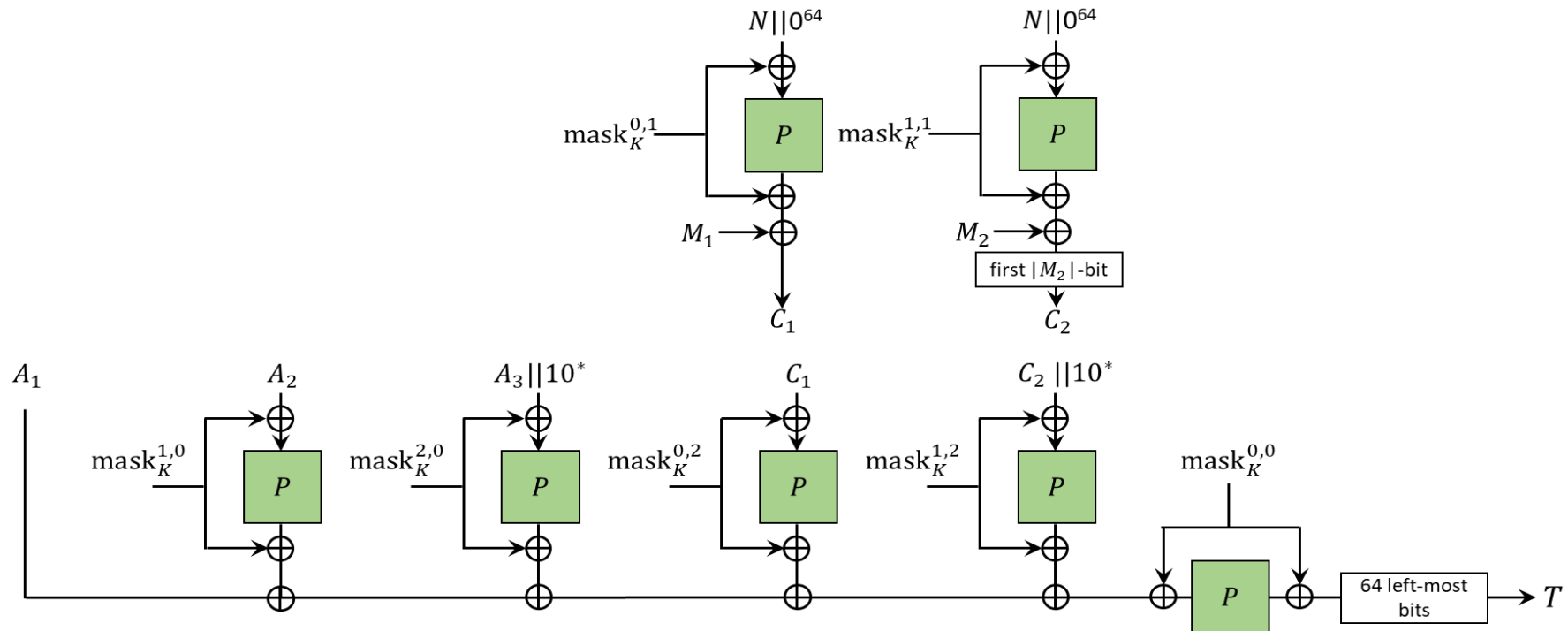
	Variant	Parameter sizes
AEAD	Ascon-128	128-bit key/nonce/tag
	Ascon-128a	128-bit key/nonce/tag
	Ascon-80-pq	160-bit key, 128-bit nonce/tag
Hash	Ascon-hash	256-bit digest
	Ascon-hasha	256-bit digest
XOF	Ascon-XOF	Arbitrary length digest
	Ascon-XOFa	Arbitrary length digest



ELEPHANT

- Permutation-based (Spongent and Keccak[200]) AEAD scheme
- Nonce-based Encrypt-then-MAC mode
- Only finalist with a parallel mode
- Design tweak: Mode slightly modified to achieve authenticity under nonce-reuse.

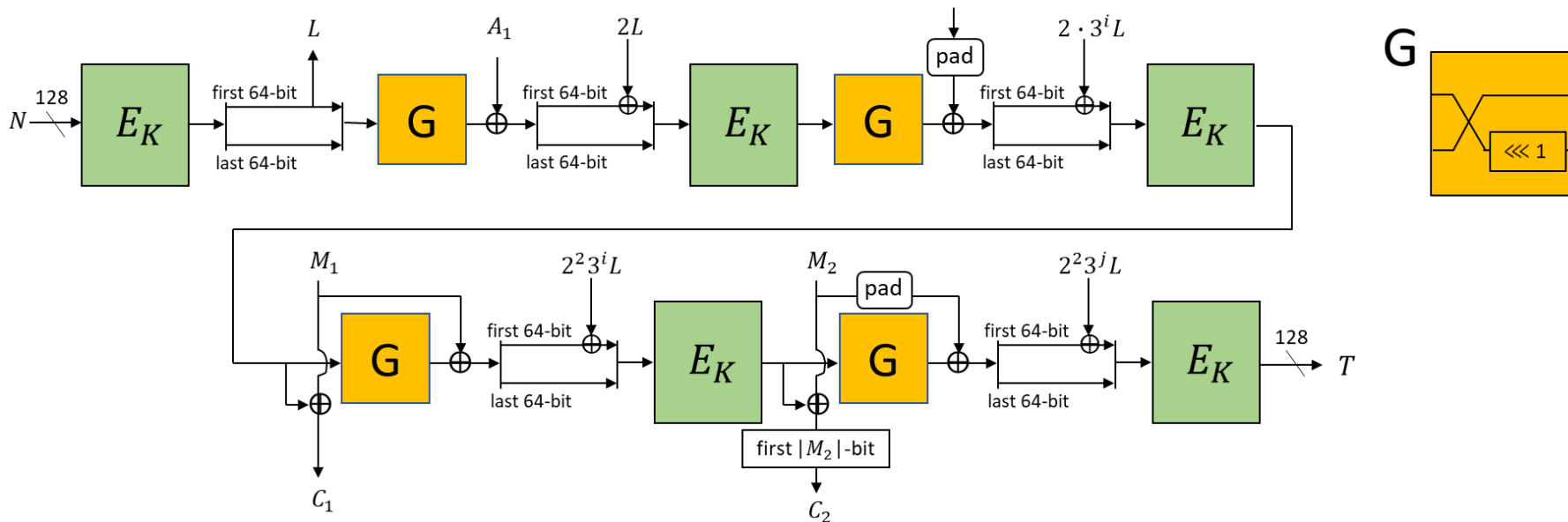
Variant	Parameter sizes
Dumbo	128-bit key, 96-bit nonce, 64-bit tag
Jumbo	128-bit key, 96-bit nonce, 64-bit tag
Delirium	128-bit key, 96-bit nonce, 128-bit tag



GIFT-COFB

- Block-cipher (GIFT-128) based AEAD scheme
- Combined Feedback (COFB) mode
- No design tweak

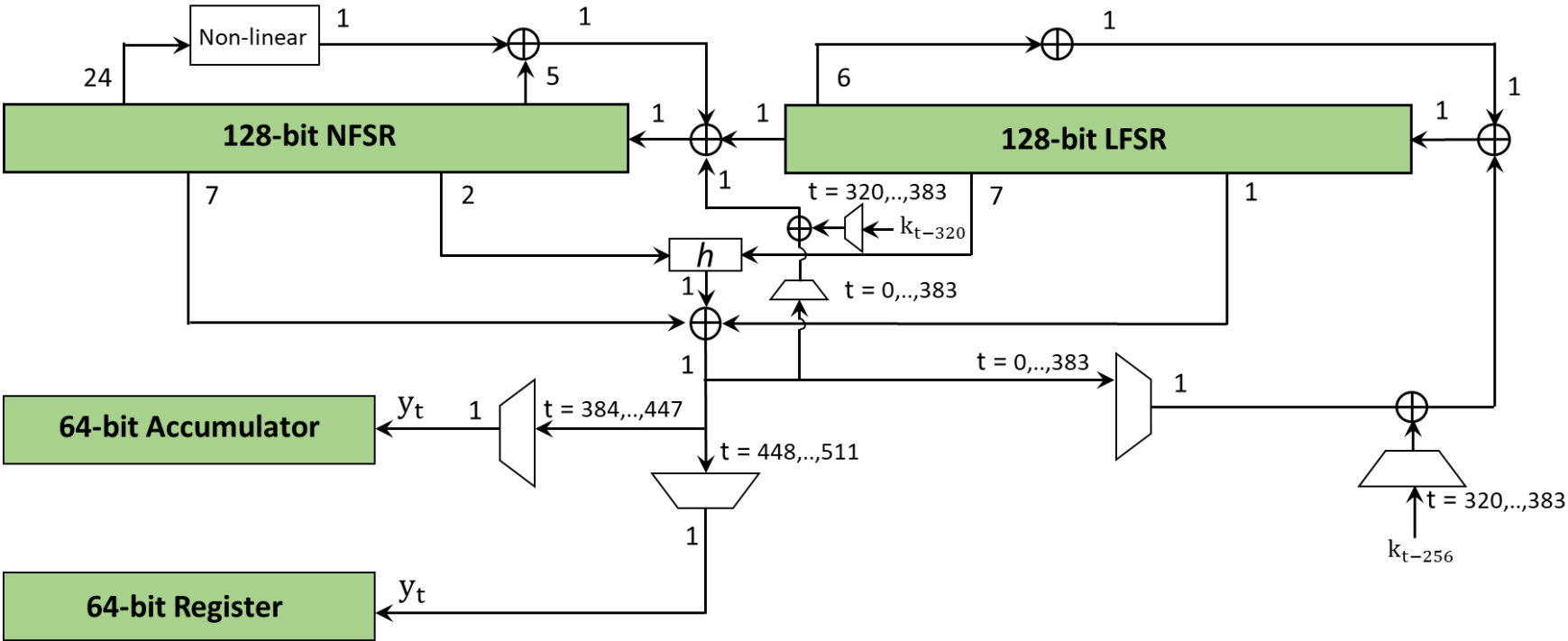
Variant	Parameter sizes
Gift-COFB	128-bit key/nonce/tag



Grain-128AEAD

- Feedback shift register based AEAD scheme
- Design tweak on the initialization part
- (Earlier versions) Part of eSTREAM portfolio, included in ISO/IEC 29167-13:2005

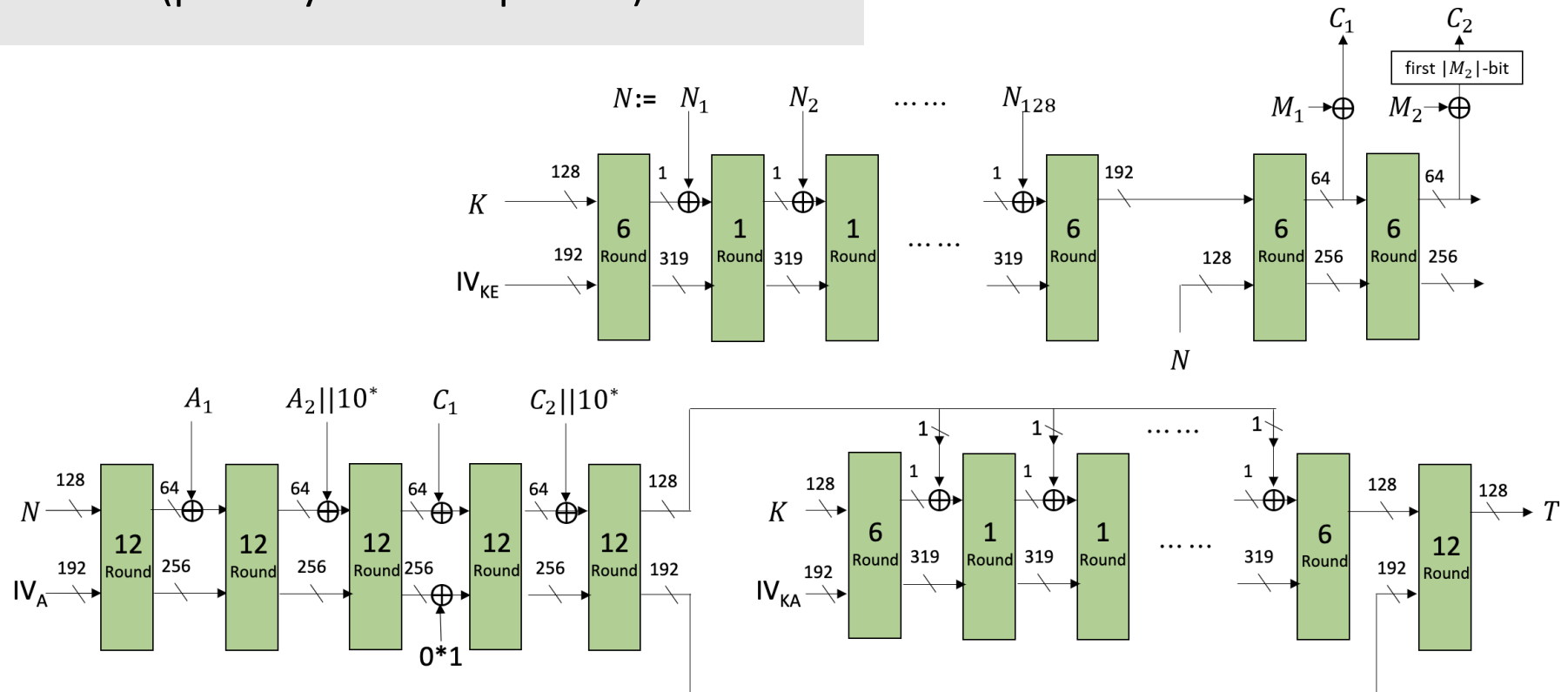
Variant	Parameter sizes
Grain-128AEAD	128-bit key, 96-bit nonce, 64-bit tag



ISAP

- Permutation-based (Ascon and Keccak permutations) AEAD scheme
- Can be paired with Ascon Hash
- Nonce-based Encrypt-then-MAC mode
- Algorithm-level security against implementation attacks
- No design tweak (primary variant updated)

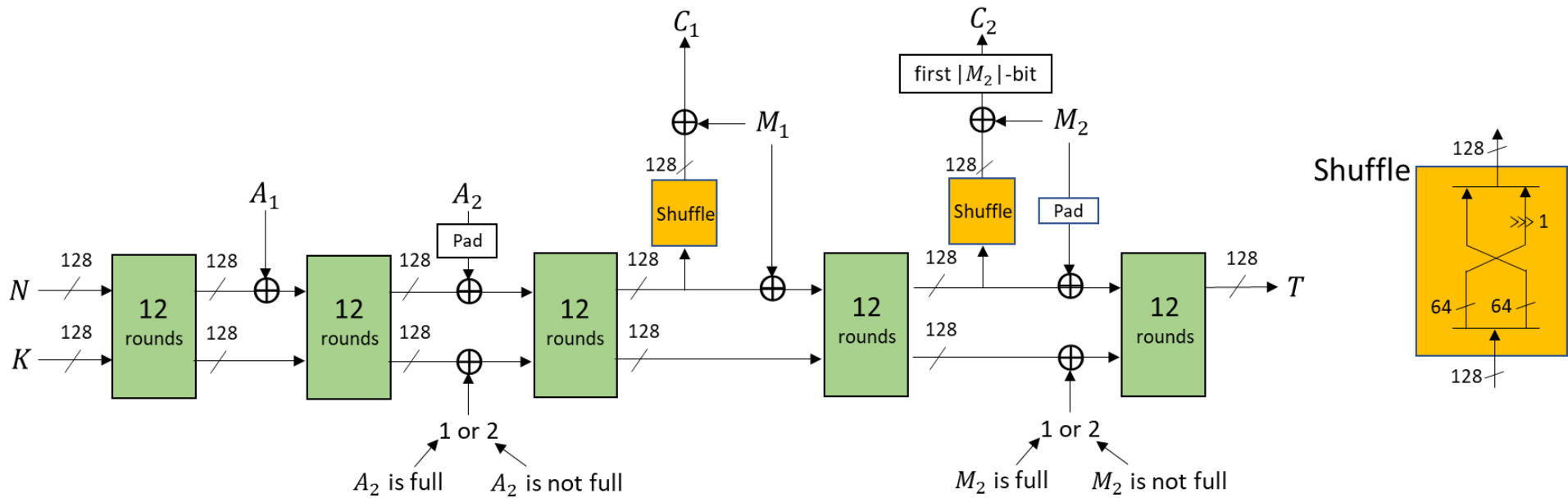
Variant	Parameter sizes
ISAP-A-128a	128-bit key/nonce/tag
ISAP-K-128a	128-bit key/nonce/tag
ISAP-A-128	128-bit key/nonce/tag
ISAP-K-128	128-bit key/nonce/tag



PHOTON-BEETLE

- Family of permutation-based (256-bit Photon permutation) AEAD & hashing scheme
- Sponge-like mode with a combined feedback.
- No design tweak

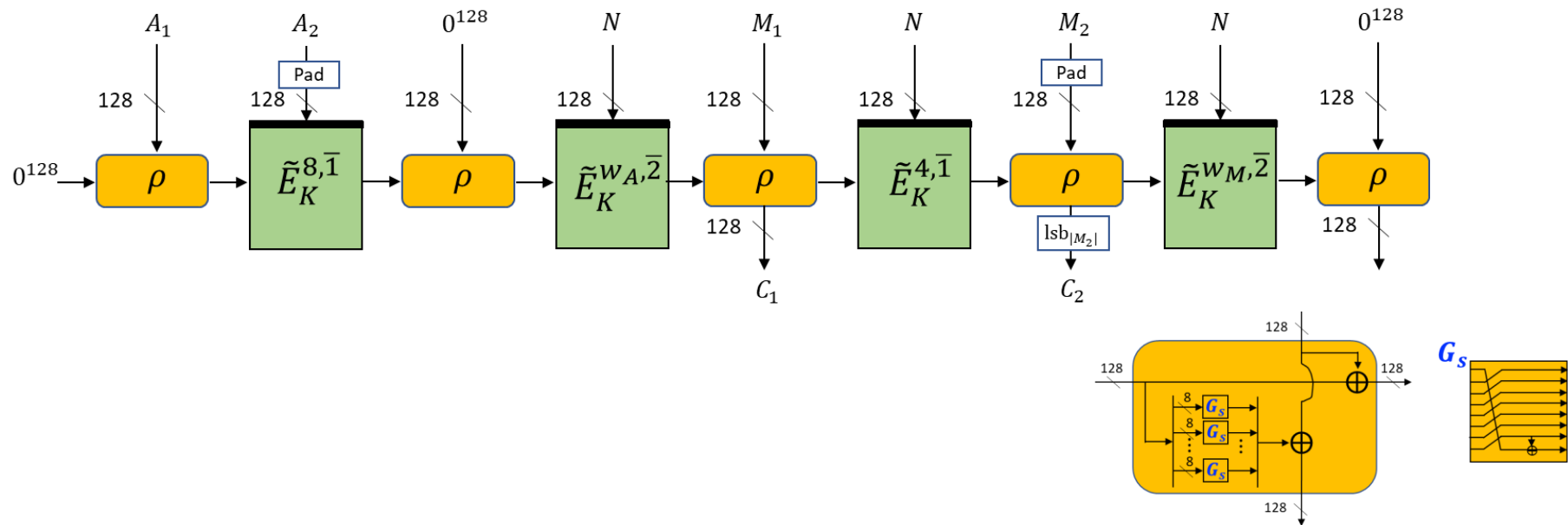
	Variant	Parameter sizes
AEAD	Photon-Beetle-AEAD[128]	128-bit key/nonce/tag
	Photon-Beetle-AEAD[32]	128-bit key/nonce/tag
Hash	Photon-Beetle-Hash[32]	256-bit digest



ROMULUS

- Family of tweakable-block-cipher (Skinny) based AEAD & hashing
- Romulus-N: rate-1 TBC-based combined feedback, Romulus-M: MAC-then-Encrypt
- Nonce-misuse and nonce-respecting variants
- Design tweak to reduce the number of rounds from 56 to 40, removal of non-primary variants, addition of new variants.

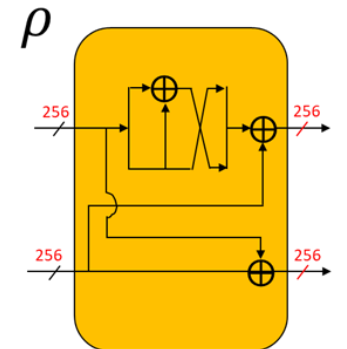
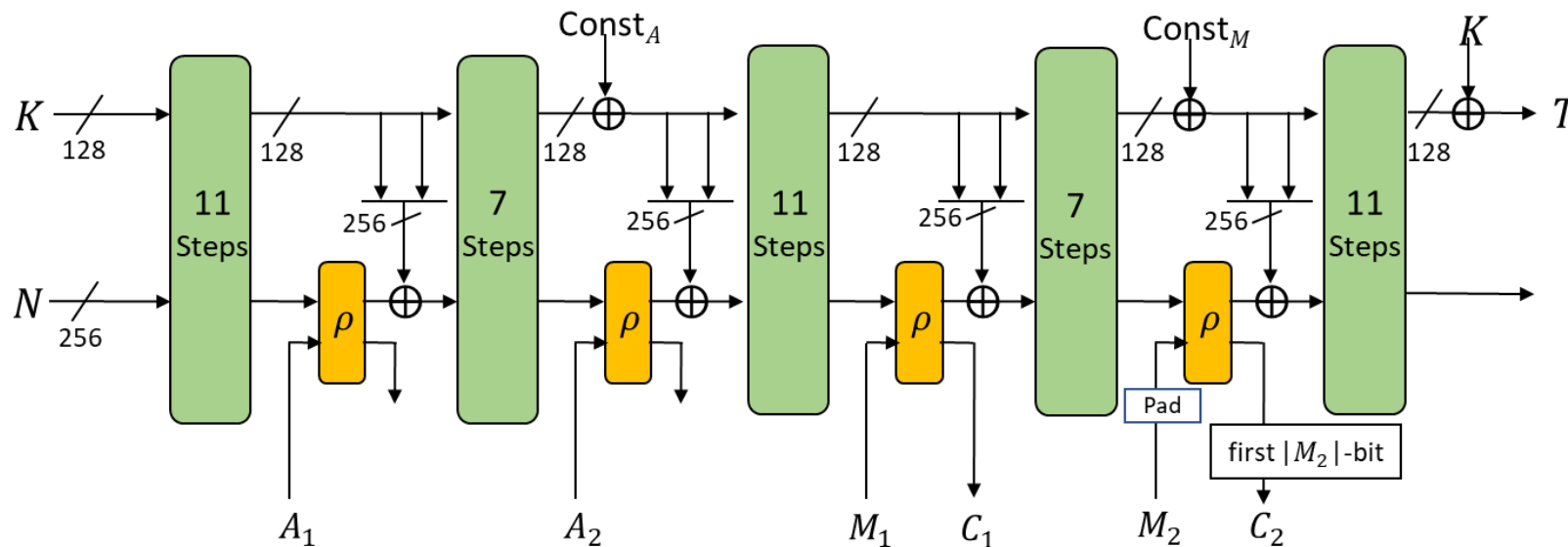
	Variant	Parameter sizes
AEAD	Romulus-N	128-bit key/nonce/tag
	Romulus-M	128-bit key/nonce/tag
	Romulus-T	128-bit key/nonce/tag
Hash	Romulus-H	256-bit digest



SPARKLE

- Family of permutation-based AEAD (SCHWAEMM) and hashing (ESCH)
- ARX based design
- Sponge construction with combined feedback
- Tweak to change the primary variant

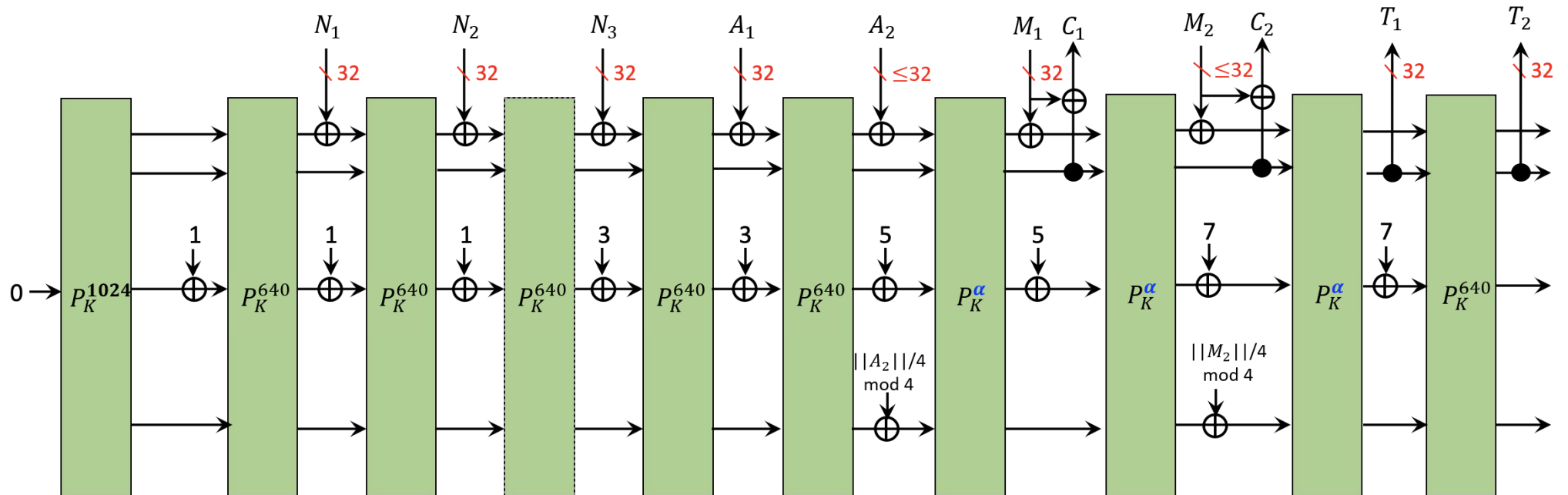
	Variant	Parameter sizes
AEAD	SCHWAEMM256-128	128-bit key/tag, 256-bit nonce
	SCHWAEMM128-128	128-bit key/nonce/tag
	SCHWAEMM192-192	192-bit key/nonce/tag
	SCHWAEMM256-256	256-bit key/nonce/tag
Hash	ESCH256	256-bit digest
	ESCH384	384-bit digest
XOF	XOESCH256	Arbitrary length digest
	XOESCH384	Arbitrary length digest



TINYJAMBU

- Keyed-permutation based AEAD scheme
- Uses 128-bit nonlinear feedback shift register
- Inspired by JAMBU (CAESAR candidate)
- Design tweak: increase in number of rounds to improve security margin.

Variant	Parameter sizes
TinyJambu-128	128-bit key, 96-bit nonce, 64-bit tag
TinyJambu-192	192-bit key, 96-bit nonce, 64-bit tag
TinyJambu-256	256-bit key, 96-bit nonce, 64-bit tag

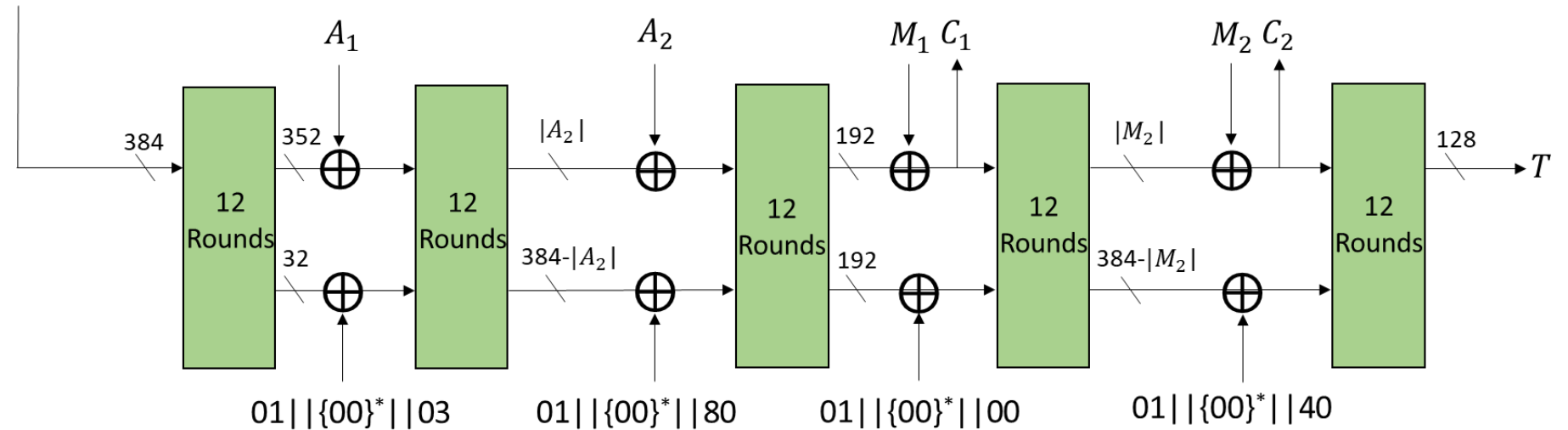


XOODYAK

- Family of permutation based AEAD & hashing scheme
- Based on 384-bit Xoodoo permutation
- Uses Cyclist mode
- Design tweak: simplified initialization to improve performance for short messages

	Variant	Parameter sizes
AEAD	Xoodyak	128-bit key/nonce/tag
Hash	Xoodyak	256-bit digest
XOF	Xoodyak	Arbitrary length digest

$K || N || (\text{byte-length of } N) || 01 || \{00\}^* || 02$



Security Margins and Claims

Security Requirements: At least 128-bit keys, input message sizes of at least $2^{50}-1$ bytes etc.

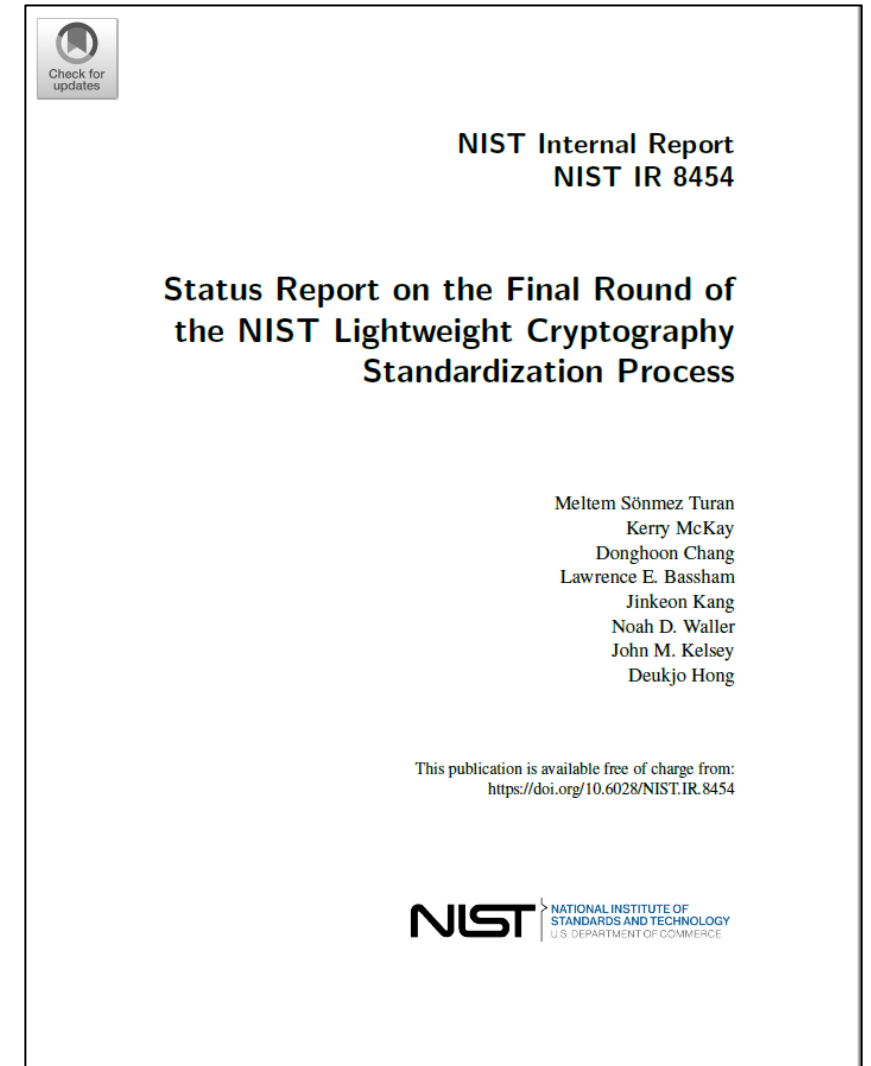
All finalists have met the security requirements and provided sufficient security margins.

- None of the security claims made by the submitters have been invalidated.
- Maturity of the design is one of the important security evaluation factors.
 - Is the finalist based on well-established design principles?
 - Did the finalist receive enough third-party analysis?
 - Are there design tweaks that invalidate the earlier security analysis?
 - Are there any additional concerns (e.g., nonce misuse, related-key, RUP security, post quantum)?

Selection of Ascon

In February 2023, NIST announced the Ascon family as the winner.

- High security margin, large number of third-party analysis (designed in 2014)
- Primary choice for the for lightweight applications in the final CAESAR portfolio (in 2019)
- No design tweaks
- Performance advantages over NIST standards (AES-GCM and SHA-2) in hardware and software
- Implementation and design flexibility
- Mode-level protection mechanism against leakage and lower additional cost for protected implementations
- Support for additional functionalities XOF, dedicated MAC, in addition to Hash



Which variants to standardize?

	Variant	Parameter sizes
AEAD	Ascon-128	128-bit key/nonce/tag
	Ascon-128a	128-bit key/nonce/tag
	Ascon-80-pq	160-bit key, 128-bit nonce/tag
Hash	Ascon-hash	256-bit digest
	Ascon-hasha	256-bit digest
XOF	Ascon-XOF	Arbitrary length digest
	Ascon-XOFa	Arbitrary length digest

Current tentative decisions:

- Either Ascon-128 or both Ascon-128 and Ascon-128a
- Do not include Ascon-80pq
- XOF standardization instead of hash functions

Possible Updates

- Support of shorter tags: 64 and 96-bit tag
- Support for customization strings
- Little endian encoding of inputs for more efficient implementations
- Support for additional functionalities (PRF, MAC, KDF, DRBG etc.)

NEXT STEPS

- Publication of the draft standards describing the Ascon family (tentative in 2023)
 - Special Publication (SP) series rather than Federal Information Processing Standards (FIPS) (tentative decision)
- Public comments period of 60 to 90 days

CONTACT US

lightweight-crypto@nist.gov

PUBLIC FORUM lwc-forum@list.nist.gov

GITHUB <https://github.com/usnistgov/Lightweight-Cryptography-Benchmarking>

WEBSITE <https://csrc.nist.gov/Projects/lightweight-cryptography>