

Shaking up Authenticated Encryption

Joan DAEMEN¹ Seth HOFFERT Silvia MELLA¹ Gilles VAN ASSCHE² Ronny VAN KEER²

¹Radboud University ²STMicroelectronics

RWC 2025, Sofia, Bulgaria, March 26–28, 2025

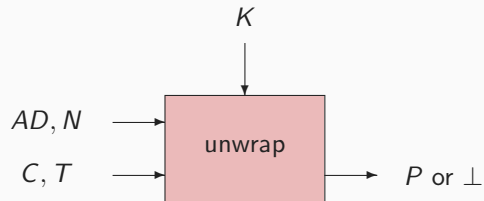
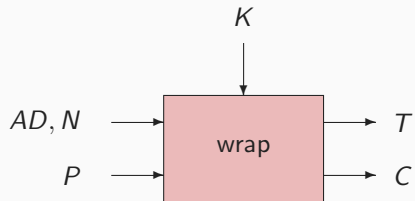
Radboud University



life.augmented

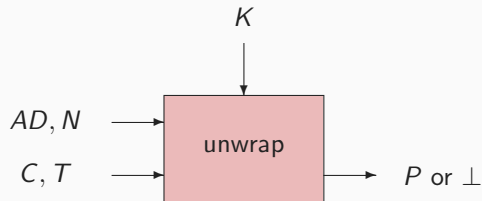
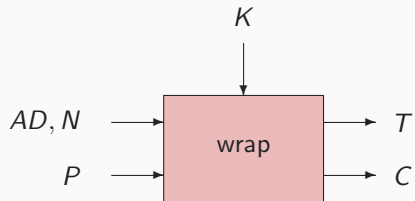
What?

Authenticated encryption...



- **wrap** takes (K, N, AD, P) and returns C, T
- **unwrap** takes (K, N, AD, C, T) and returns P or error $\perp$

Authenticated encryption...



- ▶ wrap takes (K, N, AD, P) and returns C, T
- ▶ unwrap takes (K, N, AD, C, T) and returns P or error $\perp$

Ideally

- ▶ C looks random for each input
- ▶ unwrap of invalid ciphertext fails

Authenticated encryption...



- ▶ wrap takes (K, N, AD, P) and returns C, T
- ▶ unwrap takes (K, N, AD, C, T) and returns P or error $\perp$

Ideally

- ▶ C looks random for each input
- ▶ unwrap of invalid ciphertext fails

Examples

- ▶ AES-GCM, AES-CCM, Ascon-AEAD128, ChaCha20-Poly1305, and others

SHAKE128 and SHAKE256 [FIPS 202]

FIPS PUB 202

FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION

SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions

CATEGORY: COMPUTER SECURITY SUBCATEGORY: CRYPTOGRAPHY

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

This publication is available free of charge from:
<http://dx.doi.org/10.6028/NIST.FIPS.202>

August 2015



SHAKE128 and SHAKE256 [FIPS 202]

FIPS PUB 202

FEDERAL INFORMATION PROCESSING STANDARDS
PUBLICATION

SHA-3 Standard: Permutation-Based Hash and
Extendable-Output Functions

CATEGORY: COMPUTER SECURITY SUBCATEGORY: CRYPTOGRAPHY

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

This publication is available free of charge from:
<http://dx.doi.org/10.6028/NIST.FIPS.202>

August 2015



SHAKE128 and SHAKE256 [FIPS 202]

- ▶ Sponge with KECCAK- p [**24 rounds**] [Bertoni et al., EUROCRYPT 2008]
- ▶ 15 years of public scrutiny $\Rightarrow$ 12 rounds give comfortable safety margin

FEDERAL INFORMATION PROCESSING STANDARDS

SHA-3 Standard: Permutation-Based Hash and
Extendable-Output Functions

CATEGORY: COMPUTER SECURITY SUBCATEGORY: CRYPTOGRAPHY

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

This publication is available free of charge from:
<http://dx.doi.org/10.6028/NIST.FIPS.202>

August 2015



SHAKE128 and SHAKE256 [FIPS 202]

- ▶ Sponge with KECCAK- p [**24 rounds**] [Bertoni et al., EUROCRYPT 2008]
- ▶ 15 years of public scrutiny $\Rightarrow$ 12 rounds give comfortable safety margin

TurboSHAKE128 and TurboSHAKE256

- ▶ Sponge with KECCAK- p [**12 rounds**] [Bertoni et al., ePrint 2023/342] + [RFC draft in the pipe]
- ▶ Same public scrutiny applies as all cryptanalysis is on reduced-round versions

FEDERAL INFORMATION PROCESSING STANDARDS

SHA-3 Standard: Permutation-Based Hash and
Extendable-Output Functions

CATEGORY: COMPUTER SECURITY SUBCATEGORY: CRYPTOGRAPHY

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

This publication is available free of charge from:
<http://dx.doi.org/10.6028/NIST.FIPS.202>

August 2015



...on top of SHAKE and TurboSHAKE

SHAKE128 and SHAKE256 [FIPS 202]

- ▶ Sponge with KECCAK- p [**24 rounds**] [Bertoni et al., EUROCRYPT 2008]
- ▶ 15 years of public scrutiny $\Rightarrow$ 12 rounds give comfortable safety margin

TurboSHAKE128 and TurboSHAKE256

- ▶ Sponge with KECCAK- p [**12 rounds**] [Bertoni et al., ePrint 2023/342] + [RFC draft in the pipe]
- ▶ Same public scrutiny applies as all cryptanalysis is on reduced-round versions

Security of (Turbo)SHAKE

- ▶ Unkeyed: flat sponge claim with security strength 128/256

FEDERAL INFORMATION PROCESSING STANDARDS

SHA-3 Standard: Permutation-Based Hash and
Extendable-Output Functions

CATEGORY: COMPUTER SECURITY SUBCATEGORY: CRYPTOGRAPHY

National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

This publication is available free of charge from:
<http://dx.doi.org/10.6028/NIST.FIPS.202>

August 2015



...on top of SHAKE and TurboSHAKE

SHAKE128 and SHAKE256 [FIPS 202]

- ▶ Sponge with KECCAK- p [**24 rounds**] [Bertoni et al., EUROCRYPT 2008]
- ▶ 15 years of public scrutiny $\Rightarrow$ 12 rounds give comfortable safety margin

TurboSHAKE128 and TurboSHAKE256

- ▶ Sponge with KECCAK- p [**12 rounds**] [Bertoni et al., ePrint 2023/342] + [RFC draft in the pipe]
- ▶ Same public scrutiny applies as all cryptanalysis is on reduced-round versions

Security of (Turbo)SHAKE

- ▶ Unkeyed: flat sponge claim with security strength 128/256
- ▶ Keyed:
 - When input to (Turbo)SHAKE is prefixed with a secret key K ...
 - ... it is hard to distinguish from a random oracle

FEDERAL INFORMATION PROCESSING STANDARDS

SHA-3 Standard: Permutation-Based Hash and
Extendable-Output Functions

CATEGORY: COMPUTER SECURITY SUBCATEGORY: CRYPTOGRAPHY

National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

This publication is available free of charge from:
<http://dx.doi.org/10.6028/NIST.FIPS.202>

August 2015



Why?

Search CSRC 🔍CSRC MENU

Information Technology Laboratory

COMPUTER SECURITY RESOURCE CENTER

COMPUTER SECURITY
RESOURCE CENTER
CSRC

UPDATES2024

NIST Proposes to Update FIPS 202, "SHA-3 Standard" and Revise SP 800-185, "SHA-3 Derived Functions"

September 04, 2024

[f](#) [X](#) [in](#) [✉](#)

NIST is considering whether to specify and approve one or more SHA-3 derived functions for authenticated encryption with associated data in a new, separate Special Publication.

NIST proposes to update FIPS 202 to improve its editorial quality. For example, text about SHA-1 and Triple DES will be edited to reflect the withdrawal of those techniques, as suggested in the public comments.

NIST proposes to revise SP 800-185 to provide "streaming" specifications of the two extendable output functions (XOFs) SHAKE128 and SHAKE256, to support implementations in which the length of the data output and the complete data input are not necessarily available before the XOF is called.

The public comments included suggestions that NIST specify and approve several other SHA-3 derived functions. NIST is considering

Security and Privacy Recommendations

Activities and Products: [standards development](#)

RELATED PAGES

News Item: [NIST Requests Public Comments on FIPS 202 and SP 800-185](#)

News Item: [Decision to Update FIPS 202 and Revise SP 800-](#)

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

We support NIST's plans to specify and approve additional SHA-3 derived functions, including those for authenticated encryption with associated data.

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

With

Abstract. We support NIST's potential plan to specify SHA-3 derived functions ("Keccak Modes") for Authenticated Encryption with Associated Data (AEAD). We offer security and performance arguments for a Keccak-based AEAD as an excellent

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of SHAKE to for instance encrypt and authenticate ordered streams of messages, and investigate beyond-birthday mode usages.

We
llent

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

SH Currently approved encryption methods such as AES-GCM are challenging and error-prone to deploy, primarily because of strict limits of encryption when using random nonces. An approved AEAD that can be used safely with random nonces would be of great benefit to us.

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

SH Cu the standardization method such as AES GCM and challenge and answer to deploy e
bir pri queries, and the expected number of forgeries is cubic in the number of queries. Furthermore,
be the limited nonce length and absence of nonce-resistance make both GCM and CCM unsuitable
for use with random nonces.

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

CU. Currently, the standardization methods such as AES-GCM are challenging and even more to deploy

Furthermore, there is sufficient capacity in the Keccak permutation to accommodate long nonces/IVs together with long sequence numbers. Currently the compromise is often at $\text{nonce} + \text{ctr} = 96 + 32 = 128$ in GCM[16] and CCM[15]. This is one of the reasons why AES-GCM keys are limited to 2^{32} blocks [20, 24].

more,
table

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract. We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

CU. Currently, the standardization methods such as AES-GCM are challenging and more complex to deploy

Furthermore, there is sufficient capacity in the Keccak permutation to accommodate long nonces/IVs

Limitations of AES. Essentially, all AES [27] modes are subject to a $\approx 2^{61}$ -block "birthday bound" for encryption under a given secret key; the wide permutation size of Keccak allows more long-lived keys.

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

SHAKE. Currently, the standardization methods such as AFS, GCM, and challenge-and-answer are not designed

Furthermore, there is sufficient capacity in the Keccak permutation to accommodate long nonces/IVs

Limitations of AFS Essentially, all AFS [27] modes are subject to a $\approx 2^{61}$ -block "birthday

I support streaming XOF specification. I also think that we should standardize more flexible uses of SHAKE to for instance encrypt and authenticate ordered streams of messages, and investigate beyond-birthday mode usages.

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

CU. Current encryption algorithms exhibit significant security and usability limitations. The

Furthermore, there is sufficient capacity in the Keccak permutation to accommodate long nonces/IVs

Limitations of AES Essentially all AES [27] modes are subject to a $\approx 2^{61}$ -block "birthday

box. I support streaming XOF specification. I also think that we should standardize more flexible uses of

Current encryption algorithms exhibit significant security and usability limitations. The security of GCM and CCM is limited by the narrow 128-bit block size of AES and the 128-bit digest size of GHASH. Neither provides commitment security, both are vulnerable to release of unverified plaintext, and except for CCM with short tags, they do not behave like ideal

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

CU. Current encryption algorithms exhibit significant security and usability limitations. The

Furthermore, there is sufficient capacity in the Keccak permutation to accommodate long nonces/IVs

Limitations of AES Essentially all AES [27] modes are subject to a $\approx 2^{61}$ -block "birthday

hor. I support streaming XOF specification. I also think that we should standardize more flexible uses of

Current encryption algorithms exhibit significant security and usability limitations. The

Several specifications of Keccak-based AEAD specify the use of sessions that takes care of
nonce handling and replay protection. We strongly agree with this design principle and
of suggest that the new special publication specifies the use of sessions. Users, application

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

CU. Current encryption algorithms exhibit significant security and usability limitations. The

Furthermore, there is sufficient capacity in the Keccak permutation to accommodate long nonces/IVs

Limitations of AES Essentially all AES [27] modes are subject to a $\approx 2^{61}$ -block "birthday

bound. I support streaming XOF specification. I also think that we should standardize more flexible uses of

Current encryption algorithms exhibit significant security and usability limitations. The

several

digital

of v

suggest standardizing a parallelizable AEAD mode. NIST may also consider abstract APIs that allow "sessions" that simultaneously provide transcripts of communications and allow lightweight full-duplex protocols [19, 32].

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

CU. Current encryption algorithms exhibit significant security and usability limitations. The

Furthermore, there is sufficient capacity in the Keccak permutation to accommodate long nonces/IVs

Limitations of AES Essentially all AES [27] modes are subject to a $\approx 2^{61}$ -block "birthday

box. I support streaming XOF specification. I also think that we should standardize more flexible uses of

Current encryption algorithms exhibit significant security and usability limitations. The

Keccak seems more secure in the long run. After more than 15 years of intense cryptanalysis, the security margin of Keccak-p[1600, 24] remains very large. The best of relevant attacks apply to at most seven of 24 rounds [18, 23, 35], and halving the number of rounds to 12 would still offer a reasonable security margin [5]. AES, on the other hand, has hardly any security margin left, as is apparent in NIST's own 2021 review [25].

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

CU. Current encryption algorithms exhibit significant security and usability limitations. The

Furthermore, there is sufficient capacity in the Keccak permutation to accommodate long nonces/IVs

Limitations of AES Essentially all AES [27] modes are subject to a $\approx 2^{61}$ -block "birthday

box. I support streaming XOF specification. I also think that we should standardize more flexible uses of

Current encryption algorithms exhibit significant security and usability limitations. The

Keccak seems more secure in the long run. After more than 15 years of intense cryptanalysis, the security margin of Keccak-p[1600, 24] remains very large. The best

on SHA3, we suggest NIST uses the 12-round permutation, put forward by the Keccak team in 2016 and reconfirmed [1] in 2022.

has hardly any security margin left, as is apparent in NIST's own 2021 review [25].

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

CU. Current encryption algorithms exhibit significant security and usability limitations. The

Furthermore, there is sufficient capacity in the Keccak permutation to accommodate long nonces/IVs

Limitations of AES Essentially all AES [27] modes are subject to a $\approx 2^{61}$ -block "birthday

box. I support streaming XOF specification. I also think that we should standardize more flexible uses of

Current encryption algorithms exhibit significant security and usability limitations. The

Keccak seems more secure in the long run. After more than 15 years of intense

cryptanalysis, the security margin of Keccak-p[1600, 24] remains very large. The best

of SHA-2, we suggest NIST uses the 12-round permutation put forward by the Keccak team in 2016 and

There is broad agreement that the number of rounds in FIPS 202 is too high by at least a factor of two. (1 2).

has hardly any security margin left, as is apparent in NIST's own 2021 review [25].

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

CU. Current encryption algorithms exhibit significant security and usability limitations. The

Furthermore, there is sufficient capacity in the Keccak permutation to accommodate long nonces/IVs

Limitations of AES Essentially all AES [27] modes are subject to a $\approx 2^{61}$ -block "birthday

bound. I support streaming XOF specification. I also think that we should standardize more flexible uses of

Current encryption algorithms exhibit significant security and usability limitations. The

Keccak seems more secure in the long run. After more than 15 years of intense cryptanalysis, the security margin of Keccak-p[1600, 24] remains very large. The best

round, resulting in less significant gains. With increased architectural support, we can expect Keccak-based AEAD schemes to clearly outperform their AES counterparts, as they do in pure hardware.

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

CU. Current encryption algorithms exhibit significant security and usability limitations. The

Furthermore, there is sufficient capacity in the Keccak permutation to accommodate long nonces/IVs

Limitations of AES Essentially all AES [27] modes are subject to a $\approx 2^{61}$ -block "birthday

box. I support streaming XOF specification. I also think that we should standardize more flexible uses of

Current encryption algorithms exhibit significant security and usability limitations. The

se **Keccak seems more secure in the long run.** After more than 15 years of intense

di cryptanalysis, the security margin of Keccak-p[1600, 24] remains very large. The best

o c

re **Most experts agree that the Keccak permutation is relatively straightforward to**

of **protect against power- and emissions-based side-channel attacks.** This was one of the

t original design considerations of Keccak [6, 11], and there has been much subsequent

work.

Public Comments

From <https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/fips202-sp800-185-decision-proposal-comments-2024.pdf>

Abstract We support NIST's potential plan to specify SHA-2 derived functions

I support streaming XOF specification. I also think that we should standardize more flexible uses of

CU. Current encryption algorithms exhibit significant security and usability limitations. The

Furthermore, there is sufficient capacity in the Keccak permutation to accommodate long nonces/IVs

Limitations of AES Essentially all AES [27] modes are subject to a $\approx 2^{61}$ -block "birthday

box. I support streaming XOF specification. I also think that we should standardize more flexible uses of

Current encryption algorithms exhibit significant security and usability limitations. The

Keccak seems more secure in the long run. After more than 15 years of intense

cryptanalysis, the security margin of Keccak-p[1600, 24] remains very large. The best

Most experts agree that the Keccak permutation is relatively straightforward to protect against power- and emissions-based side-channel attacks. This was one of the original design considerations of Keccak [6, 11], and there has been much subsequent work.

Summary of desired properties

- ▶ Long nonce and nonce-resistance

Summary of desired properties

- ▶ Long nonce and nonce-resistance
- ▶ Beyond 2^{64} birthday bound

Summary of desired properties

- ▶ Long nonce and nonce-resistance
- ▶ Beyond 2^{64} birthday bound
- ▶ Committing security

Summary of desired properties

- ▶ Long nonce and nonce-resistance
- ▶ Beyond 2^{64} birthday bound
- ▶ Committing security
- ▶ Security against release of unverified plaintext

Summary of desired properties

- ▶ Long nonce and nonce-resistance
- ▶ Beyond 2^{64} birthday bound
- ▶ Committing security
- ▶ Security against release of unverified plaintext
- ▶ Support for sessions

Summary of desired properties

- ▶ Long nonce and nonce-resistance
- ▶ Beyond 2^{64} birthday bound
- ▶ Committing security
- ▶ Security against release of unverified plaintext
- ▶ Support for sessions
- ▶ Faster than SHAKE

How?

hashing

SHAKE128

TurboSHAKE128

SHAKE256

TurboSHAKE256

incremental hashing

hashing

overwrite duplex (OD)

SHAKE128

TurboSHAKE128

SHAKE256

TurboSHAKE256

authenticated encr.

ODW_{RAP}

incremental hashing

overwrite duplex (OD)

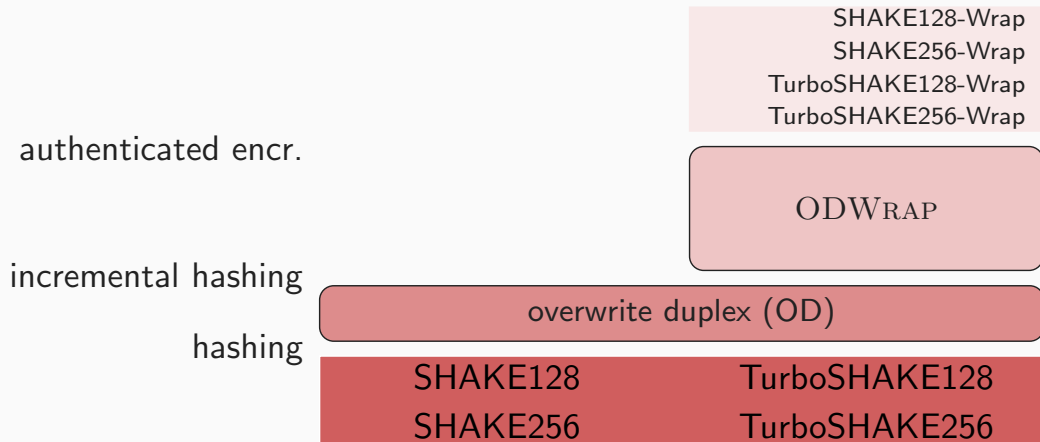
hashing

SHAKE128

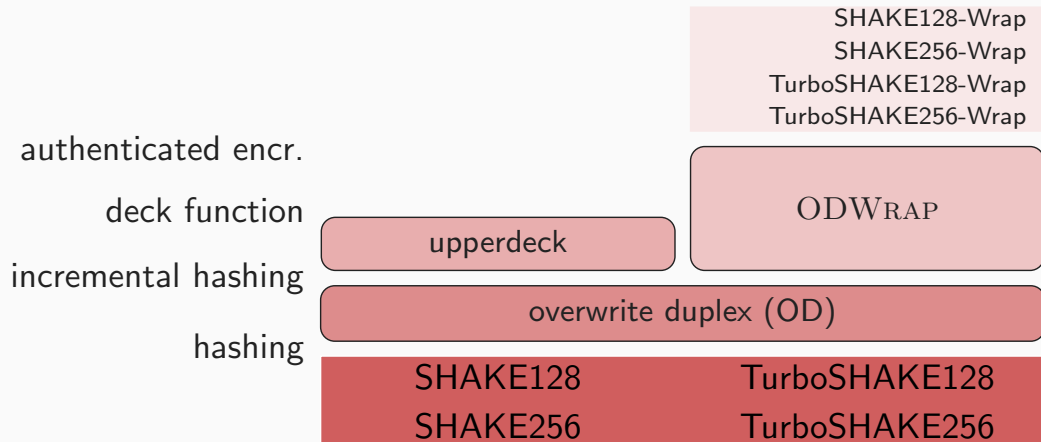
TurboSHAKE128

SHAKE256

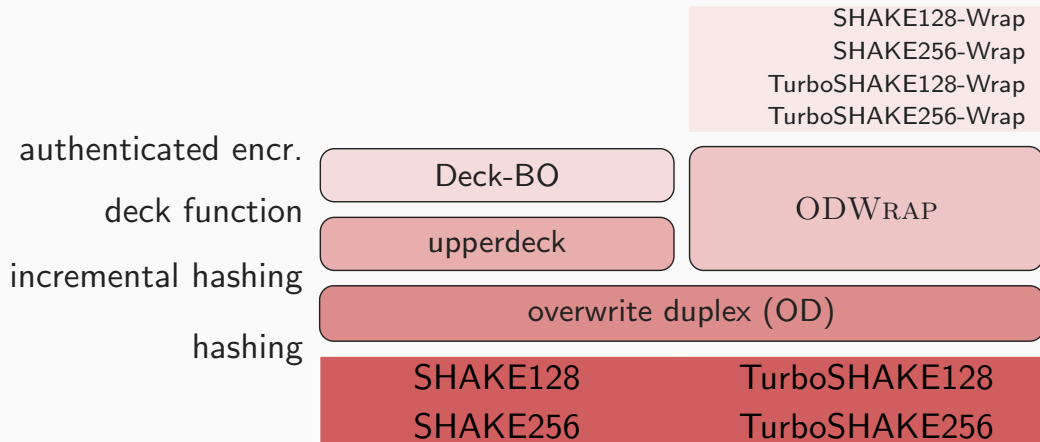
TurboSHAKE256



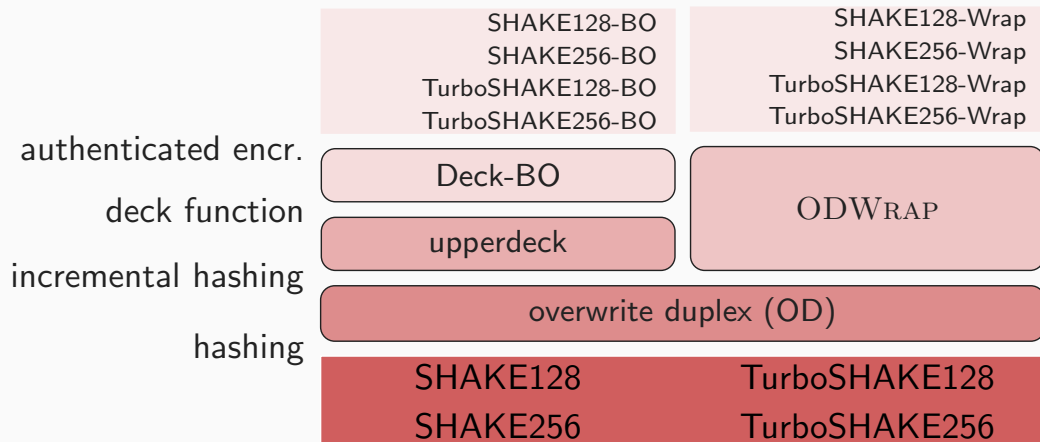
How we do it



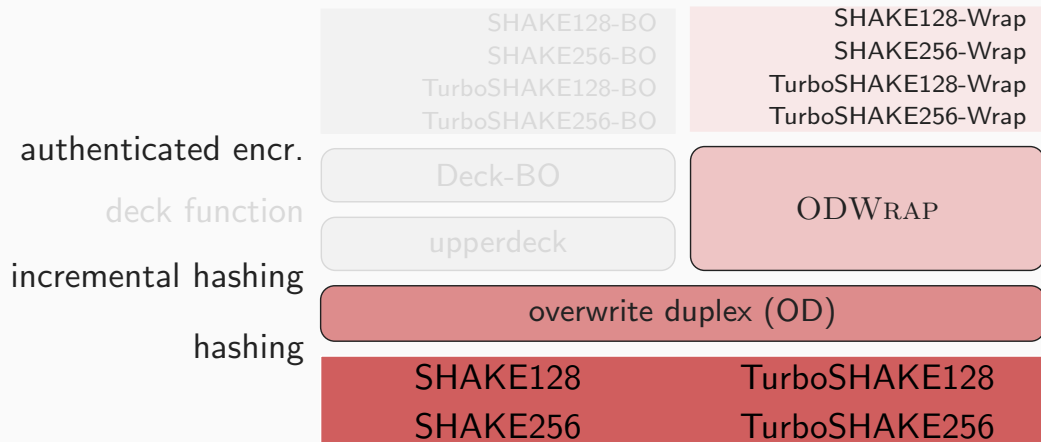
How we do it



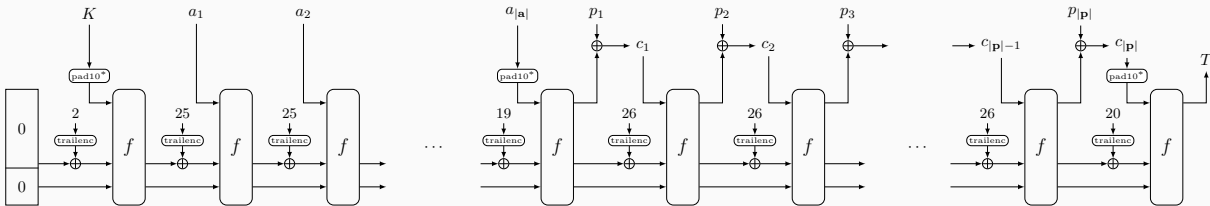
How we do it



Duplex-based authenticated encryption



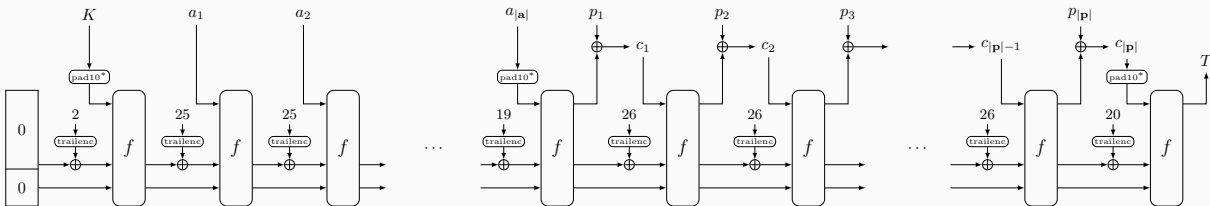
(Turbo)SHAKE-Wrap: nonce-based AE with sessions



► Duplex-based mode similar to SPONGEWrap [Bertoni et al., SAC 2011]

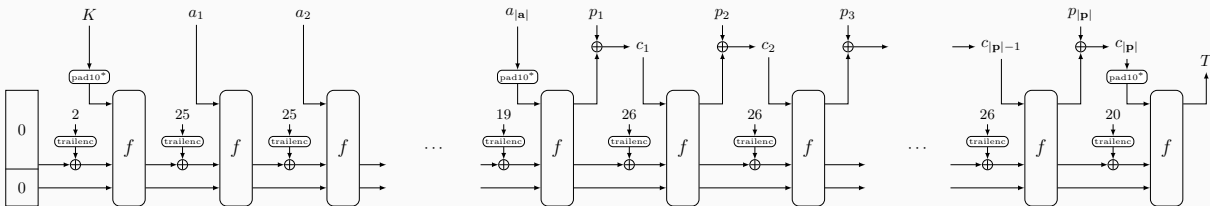
- First AD of a session to be a nonce

(Turbo)SHAKE-Wrap: nonce-based AE with sessions



- Duplex-based mode similar to SPONGEWrap [Bertoni et al., SAC 2011]
 - First AD of a session to be a nonce
- Confidentiality and integrity $\Leftarrow$ PRF security of keyed (Turbo)SHAKE

(Turbo)SHAKE-Wrap: nonce-based AE with sessions

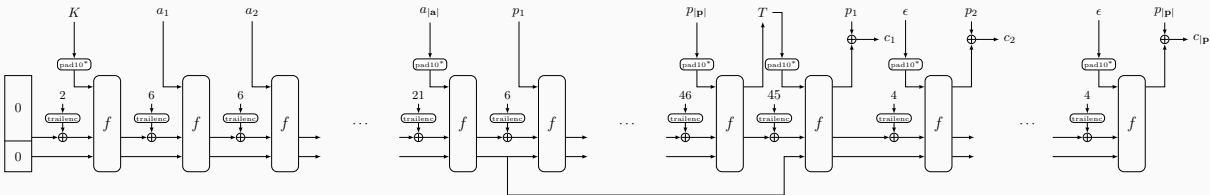


- Duplex-based mode similar to SPONGEWRAP [Bertoni et al., SAC 2011]
 - First AD of a session to be a nonce
- Confidentiality and integrity $\Leftarrow$ PRF security of keyed (Turbo)SHAKE
- Committing security $\Leftarrow$ collision resistance of (Turbo)SHAKE

Deck-based authenticated encryption

	SHAKE128-BO SHAKE256-BO TurboSHAKE128-BO TurboSHAKE256-BO	SHAKE128-Wrap SHAKE256-Wrap TurboSHAKE128-Wrap TurboSHAKE256-Wrap
authenticated encr.	Deck-BO	ODWRAP
deck function	upperdeck	
incremental hashing	overwrite duplex (OD)	
hashing	SHAKE128 SHAKE256	TurboSHAKE128 TurboSHAKE256

(Turbo)SHAKE-BO: SIV-type AE with sessions



- Based on Deck-BO [Băcuieți et al., ASIACRYPT 2022]
 - A session-supporting version of Synthetic Initialization Value (SIV) AE modes
 - Does not require a nonce
- Confidentiality and integrity $\Leftarrow$ PRF security of keyed (Turbo)SHAKE
- Committing security $\Leftarrow$ collision resistance of (Turbo)SHAKE

Solution	Security strength	Nonce-misuse resistance	Session support	Committing security
(Turbo)SHAKE128-Wrap	128 bits	no	yes	128 bits
(Turbo)SHAKE256-Wrap	256 bits	no	yes	256 bits
(Turbo)SHAKE128-BO	128 bits	yes	yes	128 bits
(Turbo)SHAKE256-BO	256 bits	yes	yes	256 bits
Ascon-AEAD128	128 bits	no	no	64 bits
ChaCha20-Poly1305	106 bits	no	no	no
AES128-GCM	64 bits	no	no	no
AES128-GCM-SIV	64 bits	yes	no	no

Table: Comparison with standard AE

	...-Wrap	...-BO	
		AD	<i>P</i> or <i>C</i>
TurboSHAKE128	3.33	3.04	6.23
TurboSHAKE256	4.06	3.84	7.82
SHAKE128	6.41	6.27	12.58
SHAKE256	8.07	7.80	15.72
ChaCha20-Poly1305	3.72		
AES128-GCM	32.32		
AES256-GCM	41.69		
Ascon-128a	4.60 ¹		

Table: Performance (ns/byte) on Raspberry Pi 4 equipped with ARM Cortex-A72 running at 1.5 GHz.

¹from <https://ascon.iaik.tugraz.at/implementations.html>

Where?

Available at <https://eprint.iacr.org/2024/1618>



Cryptology ePrint Archive

Papers ▾ Submissions ▾ About ▾ 🔍

Paper 2024/1618

Shaking up authenticated encryption

Joan Daemen, Radboud University Nijmegen, The Netherlands
Seth Hoeffert, Nebraska, USA
Silvia Mella, Radboud University Nijmegen, The Netherlands
Gilles Van Assche, STMicroelectronics Diegem, Belgium
Ronny Van Keer, STMicroelectronics Diegem, Belgium

Abstract

Authenticated encryption (AE) is a cryptographic mechanism that allows communicating parties to protect the confidentiality and integrity of messages exchanged over a public channel, provided they share a secret key. In this work, we present new AE schemes leveraging the SHA-3 standard functions SHAKE128 and SHAKE256, offering 128 and 256 bits of security strength, respectively, and their “Turbo” counterparts. They support session-based communication, where a ciphertext authenticates the sequence of messages since the start of the session. The chaining in the session allows decryption in segments, avoiding the need to buffer the entire deciphered cryptogram between decryption and validation. And, thanks to the collision resistance of (Turbo)SHAKE, they provide so-called CMT-4 committing security, meaning that they provide strong guarantees that a ciphertext uniquely binds to the key, plaintext and associated data. The AE schemes we propose have the unique combination of advantages that 1) their security is based on the security claim of SHAKE, that has received a large amount of public scrutiny, that 2) they make use of the standard KECCAK-p permutation that not only receives more and more dedicated hardware support, but also allows competitive software-only implementations thanks to the TurboSHAKE instances, and that 3) they do not suffer from a 64-bit birthday bound like most AES-based schemes.

Metadata

Available format(s)

 PDF

Category

Secret-key cryptography

Publication info

Preprint.

Keywords

authenticated encryption SHA-3 TurboSHAKE
permutation-based cryptography

Contact author(s)

joan @ cs.ru.nl
silvia.mella @ ru.nl
gilles-iacr @ noekeon.org

History

2024-10-11: approved
2024-10-10: received
[See all versions](#)

Short URL

Will appear at EuroS&P 2025

EuroS&P 2025 Home Call For... Workshops Organization Attending Past Editions



Venice, June 30 - July 4, 2025

10th IEEE European Symposium on Security and Privacy

Two approaches for **committing** and **session-supporting** AE with (Turbo)SHAKE:

- ▶ performance of duplex-based mode
- ▶ robustness and flexibility of deck-based modes, see also
 - JAMBO, BOREE, and JAMBOREE modes [Băcuieti et al., ASIACRYPT 2022]
 - **nonce-encrypting** modes [Hoffert, ePrint 2022/1711]

And **simplicity** of the modes once the layers are merged

Two approaches for **committing** and **session-supporting** AE with (Turbo)SHAKE:

- ▶ performance of duplex-based mode
- ▶ robustness and flexibility of deck-based modes, see also
 - JAMBO, BOREE, and JAMBOREE modes [Băcuieti et al., ASIACRYPT 2022]
 - **nonce-encrypting** modes [Hoffert, ePrint 2022/1711]

And **simplicity** of the modes once the layers are merged

Thanks for your attention!