Maastricht University

Accordion Modes: Basic Idea, Known Constructions, and Recent Developments

Bart Mennink

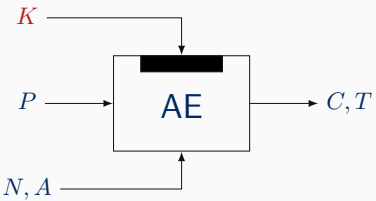
Gausta

April 14, 2026

Outline

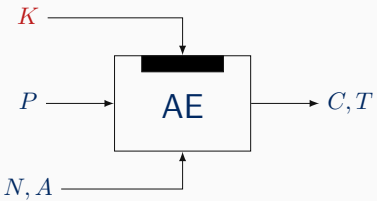
Authenticated Encryption and GCM

Authenticated Encryption (AE): Wrap and Unwrap



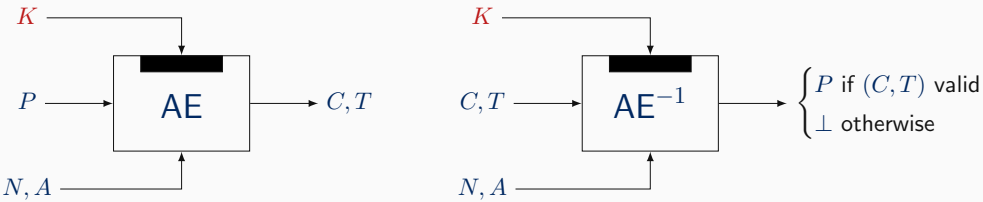
- Input: nonce N, associated data A, and plaintext P
- Ciphertext C encrypts P, and tag T authenticates (N, A, P)

Authenticated Encryption (AE): Wrap and Unwrap



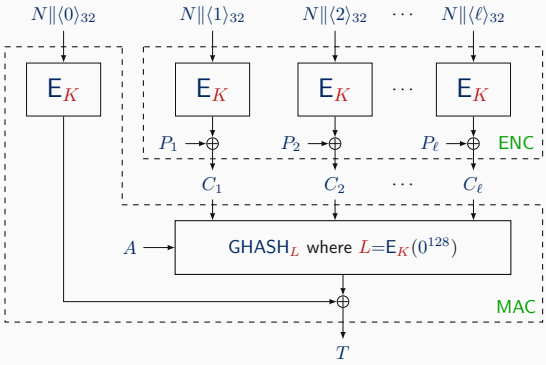
- Input: nonce N, associated data A, and plaintext P
- Ciphertext C encrypts P, and tag T authenticates (N, A, P)
- Schemes typically require uniqueness of N
  - Otherwise, ciphertexts typically leak plaintext info... or even key material!

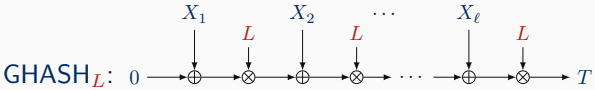
Authenticated Encryption (AE): Wrap and Unwrap



- Input: nonce N, associated data A, and plaintext P
- Ciphertext C encrypts P, and tag T authenticates (N, A, P)
- Schemes typically require uniqueness of N
  - Otherwise, ciphertexts typically leak plaintext info... or even key material!
- Unwrapping needs to satisfy that
  - Plaintext P disclosed if (C, T) comes from an evaluation of wrap
  - Error symbol otherwise

GCM for 96-bit Nonce N [MV04]

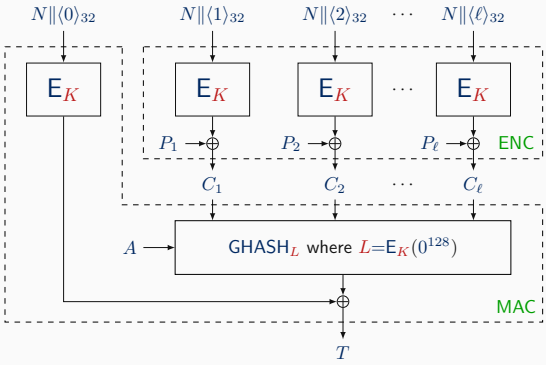


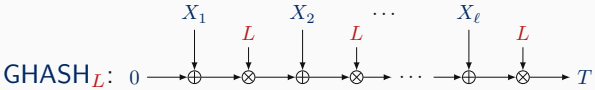


Features

- Efficient, parallelizable, inverse-free
- Widely used:
  - TLS, WPA3, IPsec, ...
- Cooler variant: ChaCha20-Poly1305!

GCM for 96-bit Nonce N [MV04]





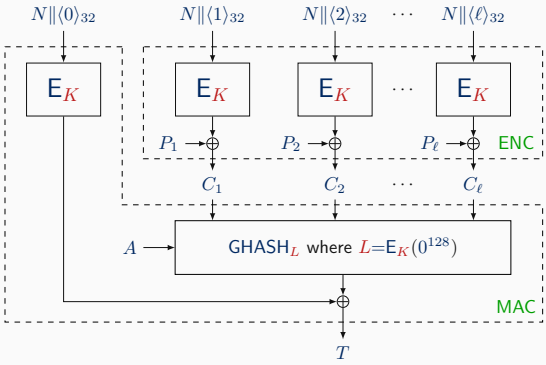
Features

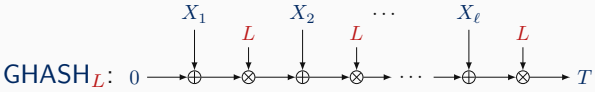
- Efficient, parallelizable, inverse-free
- Widely used:
  - TLS, WPA3, IPsec, ...
- Cooler variant: ChaCha20-Poly1305!

Secure as Long as...

- E_K is a secure block cipher
- Number of blocks doesn't exceed $2^{n/2}$
- N is never repeated...

GCM for 96-bit Nonce N [MV04]





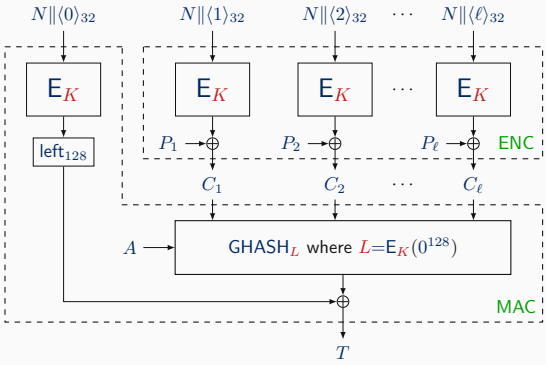
Features

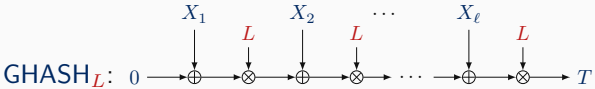
- Efficient, parallelizable, inverse-free
- Widely used:
  - TLS, WPA3, IPsec, ...
- Cooler variant: ChaCha20-Poly1305!

Secure as Long as...

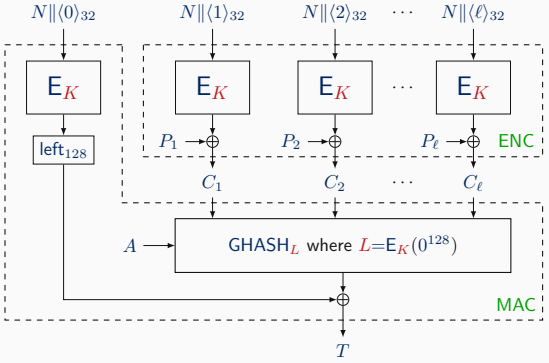
- E_K is a secure block cipher
- Number of blocks doesn't exceed $2^{n/2}$
- N is never repeated...
- ... but nonce reuse is devastating
  - Leaks $P \oplus P' = C \oplus C'$ and L

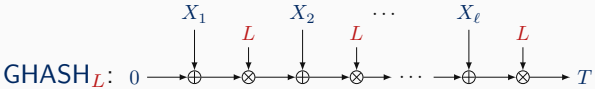
What Rijndael-256 Would Solve





What Rijndael-256 Would Solve

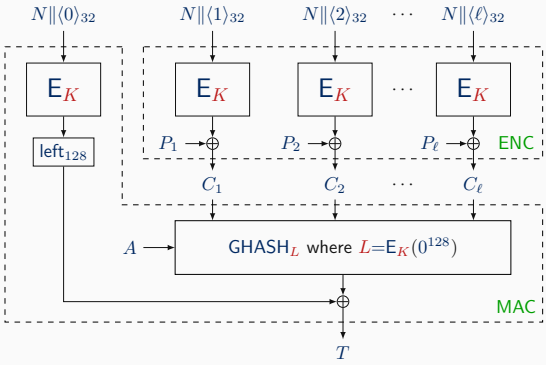


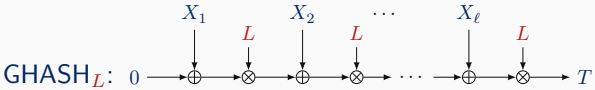


The Obvious

- Nonce$\|$counter now 256 bits
- Problem of short nonces vanishes

What Rijndael-256 Would Solve





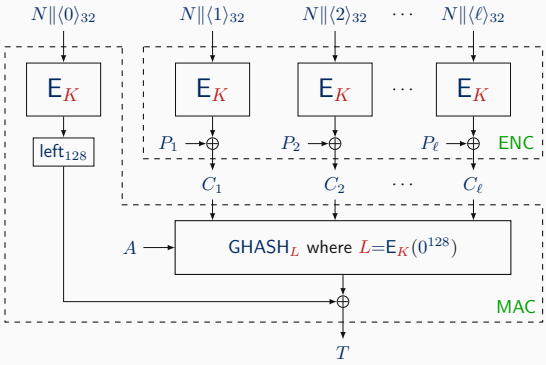
The Obvious

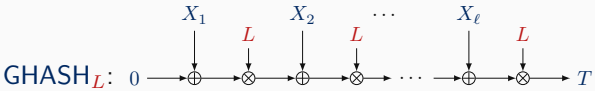
- Nonce$\|$counter now 256 bits
- Problem of short nonces vanishes

What's More?

- Nonce reuse still reveals L...
- ... but $\mathsf{E}_K(N\|\langle 0\rangle_{32})$ outputs 256-bit subkey material
  - Use half of it as GHASH-key L
  - Use other half as masking key

What Rijndael-256 Would Solve





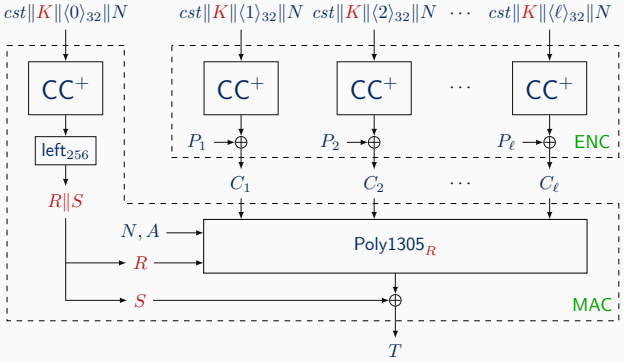
The Obvious

- Nonce$\|$counter now 256 bits
- Problem of short nonces vanishes

What's More?

- Nonce reuse still reveals L...
- ... but $\mathsf{E}_K(N\|\langle 0\rangle_{32})$ outputs 256-bit subkey material
  - Use half of it as GHASH-key L
  - Use other half as masking key
- Would give nonce misuse resilience

ChaCha20-Poly1305

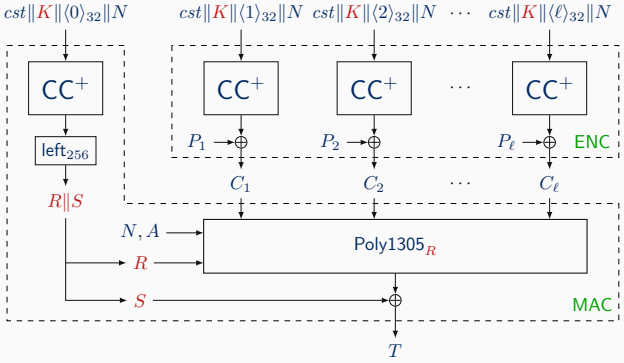


- Bernstein [Ber05, Ber08]
- RFC 8439 [NL18]

- CC: 512-bit permutation
- CC$^+$: CC plus feed-forward

$$\text{Poly1305}_R(X) = (1\|X_1)\cdot R^\ell + (1\|X_2)\cdot R^{\ell-1} + \cdots + (1\|X_{\ell-1})\cdot R^2 + X_\ell \cdot R \bmod 2^{130} - 5$$

ChaCha20-Poly1305



- Bernstein [Ber05, Ber08]
- RFC 8439 [NL18]

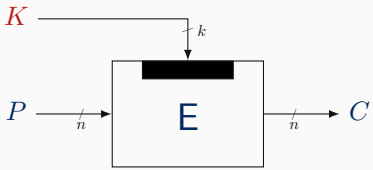
- CC: 512-bit permutation
- CC$^+$: CC plus feed-forward

- Both key and mask of Poly1305 are nonce-dependent

$$\text{Poly1305}_R(X) = (1\|X_1)\cdot R^\ell + (1\|X_2)\cdot R^{\ell-1} + \cdots + (1\|X_{\ell-1})\cdot R^2 + X_\ell \cdot R \bmod 2^{130}-5$$

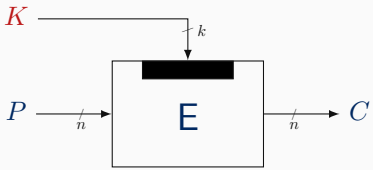
Accordion Modes

Block Ciphers



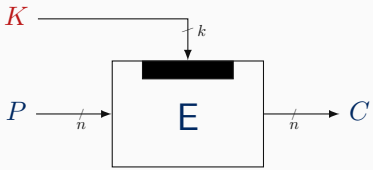
- Plaintext P is encrypted to ciphertext C using secret key K

Block Ciphers



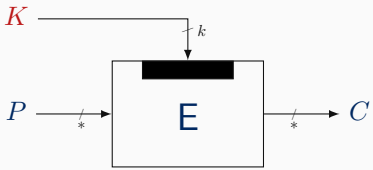
- Plaintext P is encrypted to ciphertext C using secret key K
- Fixed block size

Block Ciphers



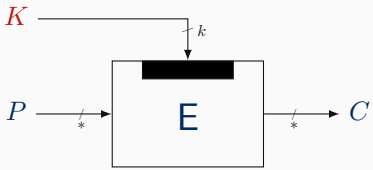
- Plaintext P is encrypted to ciphertext C using secret key K
- Fixed block size
- In order to encrypt variable sized messages, we need a mode of operation
  - These modes require a nonce

Wide Block Ciphers



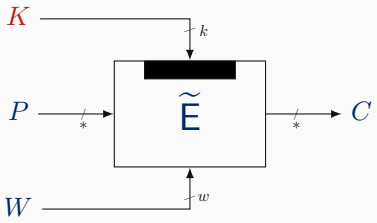
- Alternatively, we can design a wide block cipher
- A wide block cipher is a block cipher with a variable block size

Wide Block Ciphers



- Alternatively, we can design a wide block cipher
- A wide block cipher is a block cipher with a variable block size
- Every part of the output (ideally) depends on every part of the input

Tweakable Wide Block Ciphers



- A tweakable wide block cipher additionally has a tweak
- Tweak W public, ciphertext completely changes with a different tweak

NIST's Incentive to Develop Accordion Mode

- March 2024: NIST announced quest for tweakable wide block ciphers

NIST's Incentive to Develop Accordion Mode

- March 2024: NIST announced quest for tweakable wide block ciphers
- There was a workshop (June 2024) aimed to discuss ideas on requirements, designs, security goals, targets, ...

NIST's Incentive to Develop Accordion Mode

- March 2024: NIST announced quest for tweakable wide block ciphers
- There was a workshop (June 2024) aimed to discuss ideas on requirements, designs, security goals, targets, ...
  - Quote from the website: *NIST plans to develop a new mode of the AES that is a tweakable, variable-input-length-strong pseudorandom permutation (VIL-SPRP) with a reduction proof to the security of the underlying block cipher.*

NIST's Incentive to Develop Accordion Mode

- March 2024: NIST announced quest for tweakable wide block ciphers
- There was a workshop (June 2024) aimed to discuss ideas on requirements, designs, security goals, targets, ...
  - Quote from the website: *NIST plans to develop a new mode of the AES that is a tweakable, variable-input-length-strong pseudorandom permutation (VIL-SPRP) with a reduction proof to the security of the underlying block cipher.*
- A list of desired requirements was posted in April 2025

NIST's Incentive to Develop Accordion Mode

- March 2024: NIST announced quest for tweakable wide block ciphers
- There was a workshop (June 2024) aimed to discuss ideas on requirements, designs, security goals, targets, ...
  - Quote from the website: *NIST plans to develop a new mode of the AES that is a tweakable, variable-input-length-strong pseudorandom permutation (VIL-SPRP) with a reduction proof to the security of the underlying block cipher.*
- A list of desired requirements was posted in April 2025
- June 2025: NIST proposes to use HCTR2-style modes

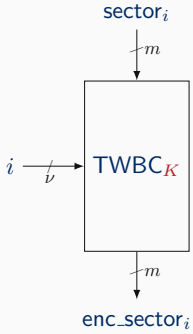
NIST's Incentive to Develop Accordion Mode

- March 2024: NIST announced quest for tweakable wide block ciphers
- There was a workshop (June 2024) aimed to discuss ideas on requirements, designs, security goals, targets, ...
  - Quote from the website: *NIST plans to develop a new mode of the AES that is a tweakable, variable-input-length-strong pseudorandom permutation (VIL-SPRP) with a reduction proof to the security of the underlying block cipher.*
- A list of desired requirements was posted in April 2025
- June 2025: NIST proposes to use HCTR2-style modes

Rest of the Talk: Why? How? Really, Why?

Applications

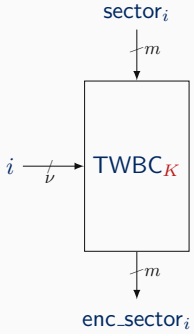
Disk Encryption



Disk Encryption Idea

- Disks are separated in sectors
- Block size is equal to the sector size
  - Typically 512 to 4096 bytes
- Physical sector number used as tweak i

Disk Encryption



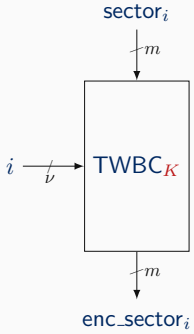
Disk Encryption Idea

- Disks are separated in sectors
- Block size is equal to the sector size
  - Typically 512 to 4096 bytes
- Physical sector number used as tweak i

Features

- Security/efficiency tradeoff: bit-flip affects entire sector
- Note: XTS further granulates disk into 128-bit blocks

Disk Encryption



Disk Encryption Idea

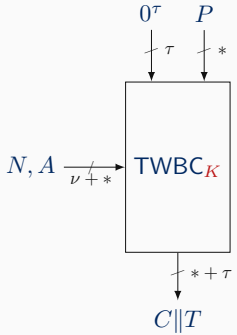
- Disks are separated in sectors
- Block size is equal to the sector size
  - Typically 512 to 4096 bytes
- Physical sector number used as tweak i

Features

- Security/efficiency tradeoff: bit-flip affects entire sector
- Note: XTS further granulates disk into 128-bit blocks

Usage

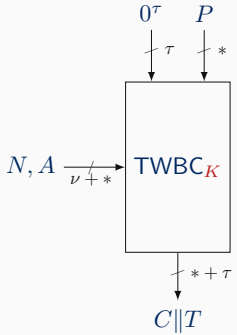
- XTS-AES is standardized as IEEE P1619 and widely used
- Android and dm-crypt support accordion mode Adiantum



Encode-Then-Encipher (ete) [BR00, HKR15]

- Encryption:
  - Prepend τ zeros to P
  - Evaluate with TWBC$_K$ to obtain $C \| T$
- Decryption:
  - Decrypt $C \| T$ using TWBC$_K^{-1}$
  - If result starts with τ zeros: output P

Encode-Then-Encipher from Tweakable Wide Block Ciphers



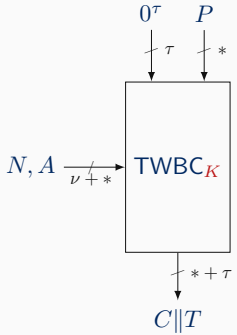
Encode-Then-Encipher (ete) [BR00, HKR15]

- Encryption:
  - Prepend τ zeros to P
  - Evaluate with TWBC$_K$ to obtain $C \| T$
- Decryption:
  - Decrypt $C \| T$ using TWBC$_K^{-1}$
  - If result starts with τ zeros: output P

Security Properties

- No security degradation relative to TWBC$_K$
- Nonce reuse has limited impact

Encode-Then-Encipher from Tweakable Wide Block Ciphers

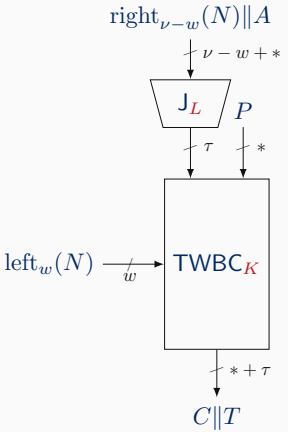


Encode-Then-Encipher (ete) [BR00, HKR15]

- Encryption:
  - Prepend τ zeros to P
  - Evaluate with TWBC$_K$ to obtain $C \| T$
- Decryption:
  - Decrypt $C \| T$ using TWBC$_K^{-1}$
  - If result starts with τ zeros: output P

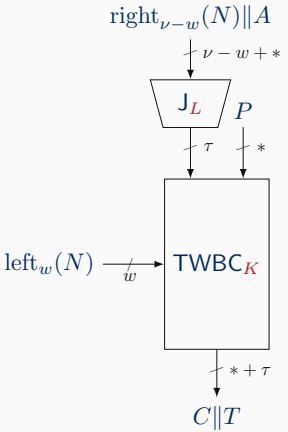
Security Properties

- No security degradation relative to TWBC$_K$
- Nonce reuse has limited impact
- What if TWBC$_K$ only supports fixed tweak size?



Advanced AE with AD (aaa) [DMMT25]

- TWBC_K: tweakable wide block cipher
- J_L: universal hash



Advanced AE with AD (aaa) [DMMT25]

- TWBC_K: tweakable wide block cipher
- J_L: universal hash

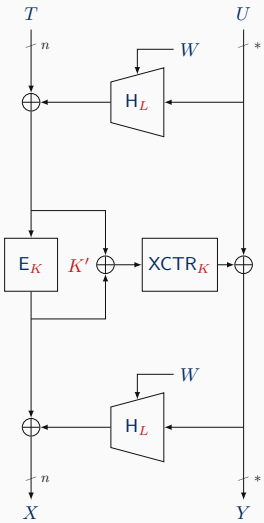
Rationale

- N partially entered into tweak
- Rest of N and A hashed into τ-bit string
- Recall: nonce reuse has limited impact

Features

- Similar to ete
- Potentially cheaper tweak processing

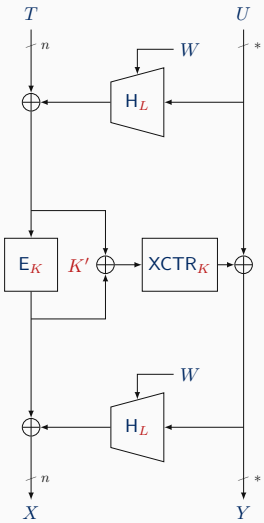
Notable Constructions



Building Blocks

- E_K: block cipher
- XCTR_K: XOR-based Counter
- H_L: universal hash

HCTR2 [CHB21]



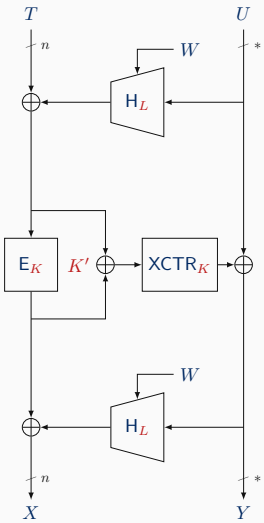
Building Blocks

- E_K: block cipher
- XCTR_K: XOR-based Counter
- H_L: universal hash

Construction

- Feistel-like structure but with a twist
- Left lane of fixed size
- Right lane of variable size

HCTR2 [CHB21]



Building Blocks

- E_K: block cipher
- XCTR_K: XOR-based Counter
- H_L: universal hash

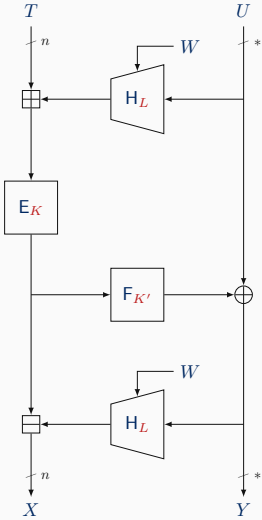
Construction

- Feistel-like structure but with a twist
- Left lane of fixed size
- Right lane of variable size

Features

- Supports inputs of $\geq n$ bits
- Birthday bound secure

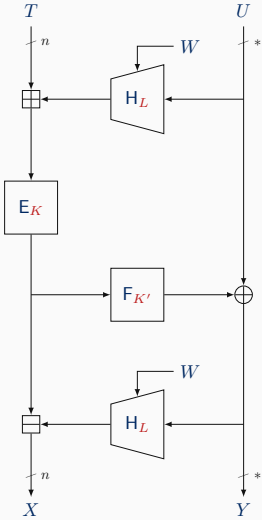
HCTR2's Sibling Adiantum [CB18]



Building Blocks

- E_K: block cipher
- $\mathsf{F}_{K'}$: stream cipher
- H_L: universal hash

HCTR2's Sibling Adiantum [CB18]



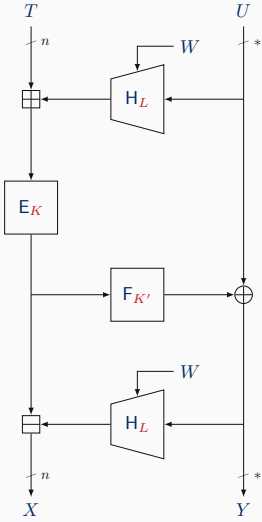
Building Blocks

- E_K: block cipher
- $\mathsf{F}_{K'}$: stream cipher
- H_L: universal hash

Construction

- Inspired by HCTR2 but for cool schemes:
  - NH/Poly1305 for universal hash
  - XChaCha12 for streaming

HCTR2's Sibling Adiantum [CB18]



Building Blocks

- E_K: block cipher
- $\mathsf{F}_{K'}$: stream cipher
- H_L: universal hash

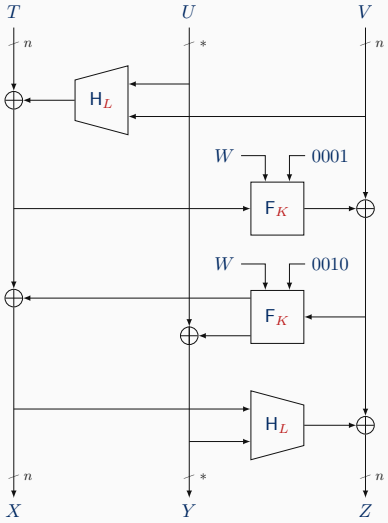
Construction

- Inspired by HCTR2 but for cool schemes:
  - NH/Poly1305 for universal hash
  - XChaCha12 for streaming

Features

- Supports inputs of $\geq n$ bits
- Birthday bound secure
- Massively used for Android device encryption

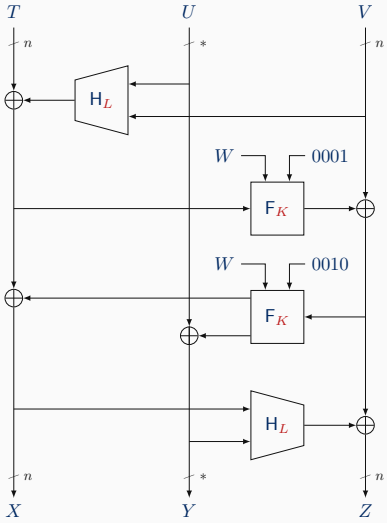
Docked Double Decker [GDM19]



Building Blocks

- F_K: stream cipher
- H_L: universal hash

Docked Double Decker [GDM19]



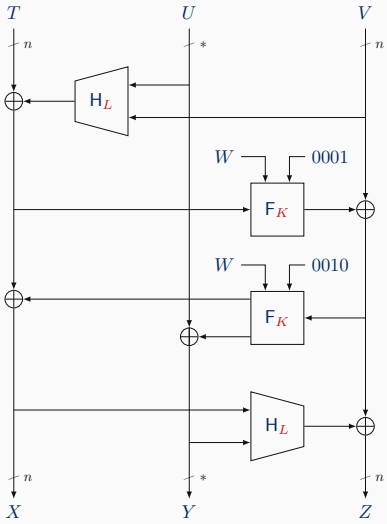
Building Blocks

- F_K: stream cipher
- H_L: universal hash

Construction

- Feistel-like structure
- Outer lanes of fixed size
- Inner lane of variable size

Docked Double Decker [GDM19]



Building Blocks

- F_K: stream cipher
- H_L: universal hash

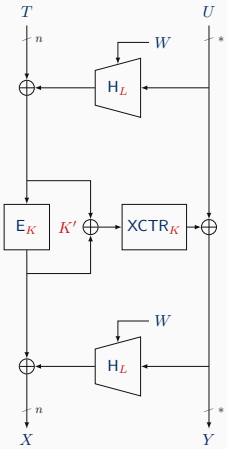
Construction

- Feistel-like structure
- Outer lanes of fixed size
- Inner lane of variable size

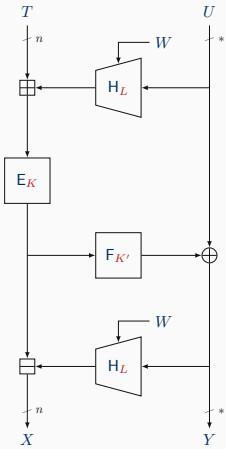
Features

- Supports inputs of $\geq 2n$ bits
- Birthday bound secure...
- ... but improved if tweaks are not reused too often

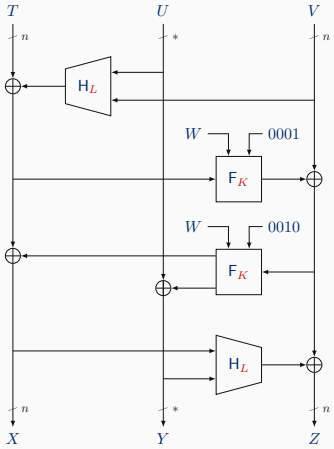
Comparison of HCTR2, Adiantum, and Docked Double Decker



HCTR2 [CHB21]



Adiantum [CB18]



Docked double decker [GDM19]

Is This Really Efficient?

Efficient Instances of Docked Double Decker With AES, and Application to Authenticated Encryption

Christoph Dobraunig[1], Krystian Matusiewicz[2], Bart Mennink[3] and Alexander Tereschenko[2]

Our Goals in [DMMT25]

- Instantiating ddd using components as used in NIST standardized schemes:
  - AES [DR02, DR20]
  - Operations in binary extension fields, e.g., as in GHASH [MV04]

Efficient Instances of Docked Double Decker With AES [DMMT25]

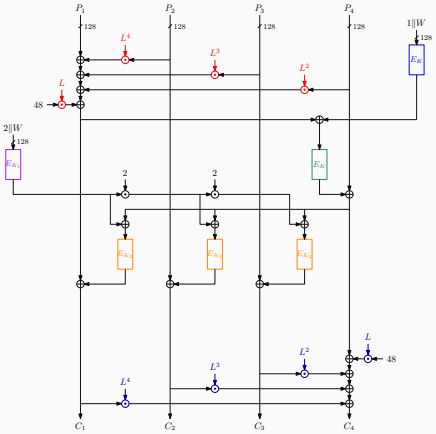
Efficient Instances of Docked Double Decker With AES, and Application to Authenticated Encryption

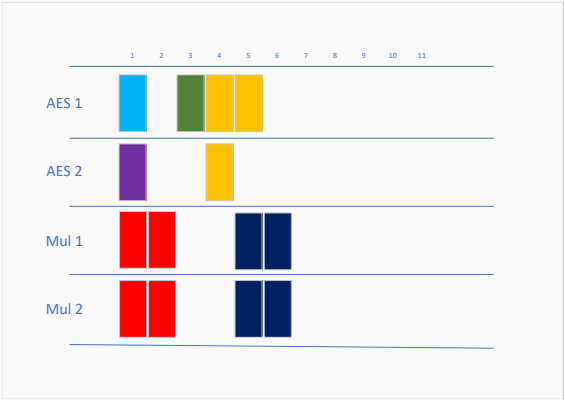
Christoph Dobraunig[1], Krystian Matusiewicz[2], Bart Mennink[3] and Alexander Tereschenko[2]

Our Goals in [DMMT25]

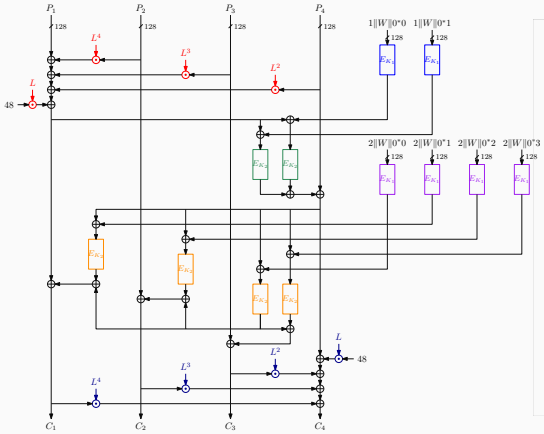
- Instantiating *ddd* using components as used in NIST standardized schemes:
  - AES [DR02, DR20]
  - Operations in binary extension fields, e.g., as in GHASH [MV04]
- Presented birthday bound secure *ddd-AES* and beyond birthday bound secure *bbb-ddd-AES* that fit NIST's accordion idea

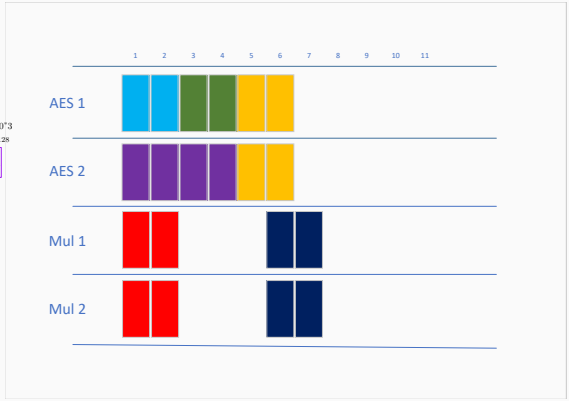
Implementation Design of $ddd\text{-}AES$ (512-Bit Message)





Implementation Design of $bbb\text{-}ddd\text{-}AES$ (512-Bit Message)





Benchmarks

- $ddd\text{-}AES$ and $bbb\text{-}ddd\text{-}AES$ on an Intel® Core™ i7-10610U
- C implementation using AES-NI and PCLMULQDQ

Message length (bytes)	32	48	64	96	128	256	512	1024	2048
$ddd\text{-}AES$ x1	6	4.3	3.4	2.8	2.5	2.3	2.2	2.1	2.1
$ddd\text{-}AES$ x2	6	3.9	3.2	2.5	2.0	1.7	1.5	1.3	1.3
$ddd\text{-}AES$ x3	9	4.6	3.1	2.5	2.1	1.4	1.2	1.1	1.0
$ddd\text{-}AES$ x4	7	4.3	3.5	2.6	2.3	1.6	1.3	1.1	1.0
$ddd\text{-}AES$ x5	8	4.6	3.8	2.4	2.2	1.5	1.2	1.1	1.0
$ddd\text{-}AES$ x6	7	4.6	3.6	2.9	2.1	1.7	1.2	1.1	1.0
$bbb\text{-}ddd\text{-}AES$ x1	8	5.0	4.0	3.2	2.9	2.6	2.5	2.5	2.5
$bbb\text{-}ddd\text{-}AES$ x2	9	5.1	3.9	3.0	2.6	1.9	1.6	1.4	1.3
$bbb\text{-}ddd\text{-}AES$ x3	8	5.2	3.8	3.0	2.5	1.7	1.4	1.2	1.1
$bbb\text{-}ddd\text{-}AES$ x4	8	5.0	4.1	3.0	2.8	1.9	1.4	1.2	1.1
$bbb\text{-}ddd\text{-}AES$ x5	9	5.9	4.1	2.8	2.8	1.7	1.5	1.3	1.2
$bbb\text{-}ddd\text{-}AES$ x6	9	5.2	4.4	3.3	2.6	2.0	1.4	1.3	1.2

Benchmarks

- *ddd-AES* and *bbb-ddd-AES* on an Intel® Core™ i7-10610U
- C implementation using AES-NI and PCLMULQDQ

Message length (bytes)	32	48	64	96	128	256	512	1024	2048
ddd-AES x1	6	4.3	3.4	2.8	2.5	2.3	2.2	2.1	2.1
ddd-AES x2	6	3.9	3.2	2.5	2.0	1.7	1.5	1.3	1.3
ddd-AES x3	9	4.6	3.1	2.5	2.1	1.4	1.2	1.1	1.0
ddd-AES x4	7	4.3	3.5	2.6	2.3	1.6	1.3	1.1	1.0
ddd-AES x5	8	4.6	3.8	2.4	2.2	1.5	1.2	1.1	1.0
ddd-AES x6	7	4.6	3.6	2.9	2.1	1.7	1.2	1.1	1.0
bbb-ddd-AES x1	8	5.0	4.0	3.2	2.9	2.6	2.5	2.5	2.5
bbb-ddd-AES x2	9	5.1	3.9	3.0	2.6	1.9	1.6	1.4	1.3
bbb-ddd-AES x3	8	5.2	3.8	3.0	2.5	1.7	1.4	1.2	1.1
bbb-ddd-AES x4	8	5.0	4.1	3.0	2.8	1.9	1.4	1.2	1.1
bbb-ddd-AES x5	9	5.9	4.1	2.8	2.8	1.7	1.5	1.3	1.2
bbb-ddd-AES x6	9	5.2	4.4	3.3	2.6	2.0	1.4	1.3	1.2

- For comparison, CBC encryption takes ≈ 1.4 cpb for 2048 byte messages

Conclusion

Conclusion

NIST's Plans

- Acc128 with birthday bound security using AES
- Acc256 with birthday bound security using a 256-bit block cipher
- BBBAcc with beyond birthday bound security using AES

Conclusion

NIST's Plans

- Acc128 with birthday bound security using AES
- Acc256 with birthday bound security using a 256-bit block cipher
- BBBAcc with beyond birthday bound security using AES
- NIST plans to standardize these using variants of HCTR2

Conclusion

NIST's Plans

- Acc128 with birthday bound security using AES
- Acc256 with birthday bound security using a 256-bit block cipher
- BBBAcc with beyond birthday bound security using AES
- NIST plans to standardize these using variants of HCTR2
- If you have any questions about this: ask Meltem!

Conclusion

NIST's Plans

- Acc128 with birthday bound security using AES
- Acc256 with birthday bound security using a 256-bit block cipher
- BBBAcc with beyond birthday bound security using AES
- NIST plans to standardize these using variants of HCTR2
- If you have any questions about this: ask Meltem!

Thank you for your attention!

References i

Daniel J. Bernstein.

The Poly1305-AES Message-Authentication Code.

In Henri Gilbert and Helena Handschuh, editors, *Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers*, volume 3557 of *Lecture Notes in Computer Science*, pages 32–49. Springer, 2005.

Daniel J. Bernstein.

The ChaCha family of stream ciphers.

`https://cr.yp.to/chacha.html`, January 2008.

References ii


Mihir Bellare and Phillip Rogaway.
Encode-Then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography.
In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, Lecture Notes in Computer Science, pages 317–330. Springer, 2000.


Paul Crowley and Eric Biggers.
Adiantum: length-preserving encryption for entry-level processors.
IACR Trans. Symmetric Cryptol., 2018(4):39–61, 2018.


Paul Crowley, Nathan Huckleberry, and Eric Biggers.
Length-preserving encryption with HCTR2.
Cryptology ePrint Archive, Paper 2021/1441, 2021.

References iii


Christoph Dobraunig, Krystian Matusiewicz, Bart Mennink, and Alexander Tereschenko.
Efficient Instances of Docked Double Decker with AES, and Application to Authenticated Encryption.
In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology - EUROCRYPT 2025 - 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4-8, 2025, Proceedings, Part I*, volume 15601 of *Lecture Notes in Computer Science*, pages 62–92. Springer, 2025.


Joan Daemen and Vincent Rijmen.
The Design of Rijndael: AES - The Advanced Encryption Standard.
Information Security and Cryptography. Springer, 2002.


Joan Daemen and Vincent Rijmen.
The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition.
Information Security and Cryptography. Springer, 2020.

References iv


Aldo Gunsing, Joan Daemen, and Bart Mennink.
Deck-Based Wide Block Cipher Modes and an Exposition of the Blinded Keyed Hashing Model.
IACR Trans. Symmetric Cryptol., 2019(4):1–22, 2019.


Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway.
Robust Authenticated-Encryption AEZ and the Problem That It Solves.
In Elisabeth Oswald and Marc Fischlin, editors, *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*, volume 9056 of *Lecture Notes in Computer Science*, pages 15–44. Springer, 2015.


David A. McGrew and John Viega.
The Security and Performance of the Galois/Counter Mode (GCM) of Operation.
In Anne Canteaut and Kapalee Viswanathan, editors, *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355. Springer, 2004.


Yoav Nir and Adam Langley.
ChaCha20 and Poly1305 for IETF Protocols.
Request for Comments (RFC) 8439, June 2018.
`https://tools.ietf.org/html/rfc8439`.