# Towards Optimally Secure Deterministic Authenticated Encryption Schemes

soft merge with

# Making GCM Great Again: Toward Full Security and Longer Nonces

Ashwin Jha
RUB

Byeonghak Lee
Samsung SDS

Eurocrypt 2025

6 May, 2025

# Towards Optimally Secure Deterministic Authenticated Encryption Schemes

Yu Long Chen
KU Leuven

Avijit Dutta
TCG CREST

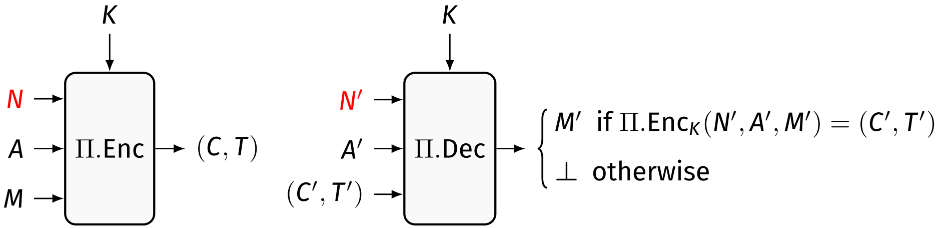
Ashwin Jha
RUB

Mridul Nandi
ISI Kolkata

Eurocrypt 2025

6 May, 2025

# Authenticated Encryption with Associated Data

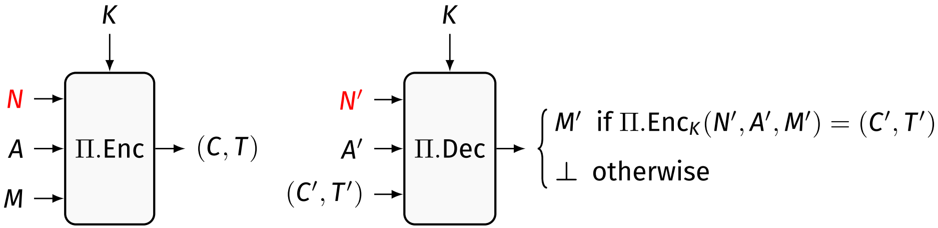



- AEAD *encrypts* the message $M$ + *authenticates* the metadata & message $(A, M)$
- Widely deployed (TLS, IPsec, wireless standards)

GCM     CCM     ChaCha20-Poly1305     Ascon

- Nonce is supposed to be unique in encryption

# Authenticated Encryption with Associated Data

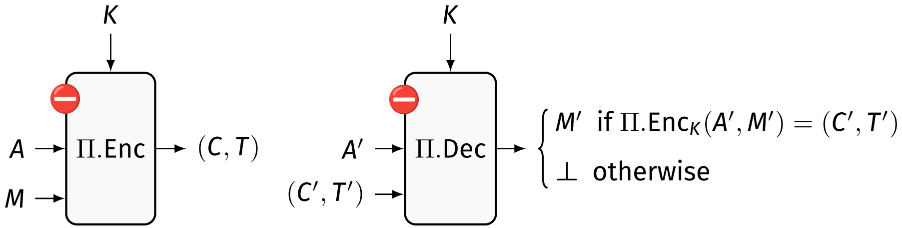



- AEAD *encrypts* the message $M$ + *authenticates* the metadata & message $(A, M)$
- Widely deployed (TLS, IPsec, wireless standards)

GCM CCM ChaCha20-Poly1305 Ascon

- Nonce is supposed to be unique in encryption

# Deterministic AEAD [RS: EC '06]




- AEAD *without* a nonce [can be absorbed in the associated data]
- Encryption at rest (iCloud, AWS) and tokenization (PCI-compliant systems)

SIV GCM-SIV

# Why Use a Nonce?

Uniqueness of nonce in encryption ensures security and efficiency

- Security:
  - DAEAD *leaks equality* when message + metadata repeat.
  - Nonce ensures *fresh* randomness per encryption query
- Efficiency:
  - DAEAD are inherently *two-pass* (rate[1] is capped at $0.5$)
  - Nonce allows for *single-pass* schemes

[1]The ratio of number of $n$-bit blocks in the input to the number of primitive calls.

# Why Use a Nonce?

Uniqueness of nonce in encryption ensures security and efficiency

- Security:
  - DAEAD *leaks equality* when message + metadata repeat.
  - Nonce ensures *fresh* randomness per encryption query
- Efficiency:
  - DAEAD are inherently *two-pass* (rate[1] is capped at $0.5$)
  - Nonce allows for *single-pass* schemes

[1]The ratio of number of *n*-bit blocks in the input to the number of primitive calls.

# Why Use a Nonce?

Uniqueness of nonce in encryption ensures security and efficiency

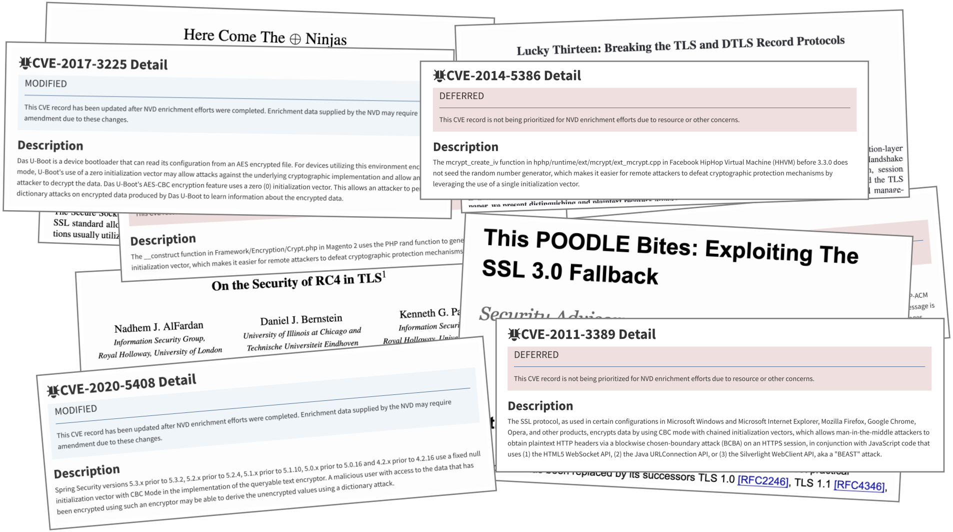
- Security:
  - DAEAD *leaks equality* when message + metadata repeat.
  - Nonce ensures *fresh* randomness per encryption query
- Efficiency:
  - DAEAD are inherently *two-pass* (rate[1] is capped at $0.5$)
  - Nonce allows for *single-pass* schemes

⚠️ Nonce-reuse is *strictly* prohibited!

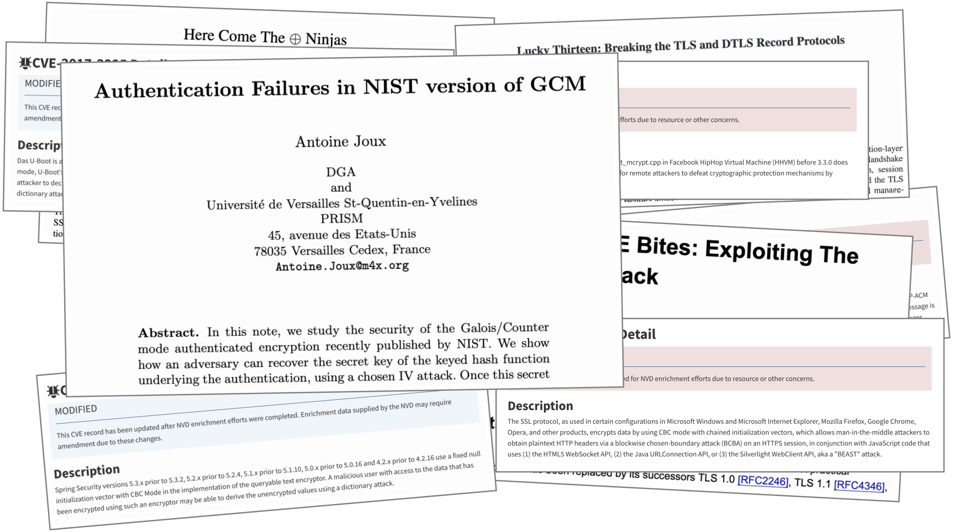
[1] The ratio of number of $n$-bit blocks in the input to the number of primitive calls.

# The Curse of Nonce-Misuse

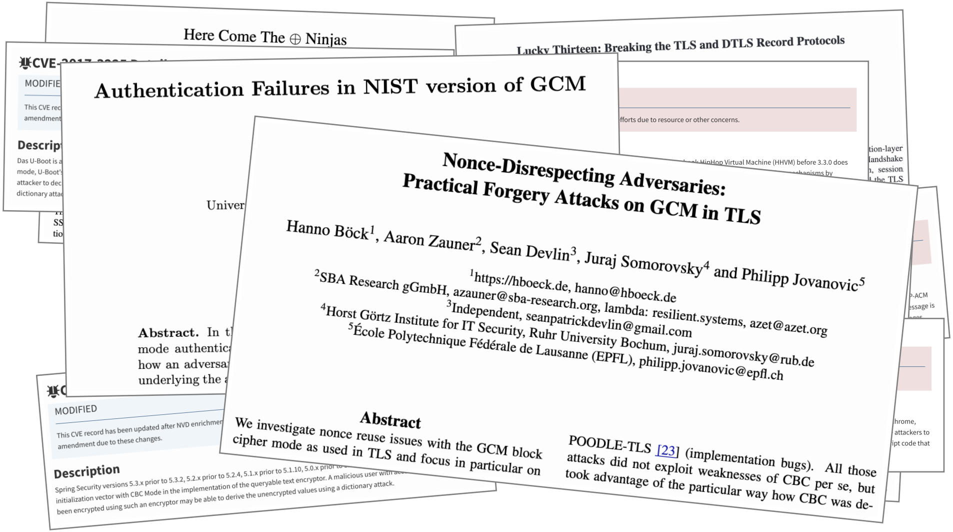
# The Curse of Nonce-Misuse



# The Curse of Nonce-Misuse



# The Curse of Nonce-Misuse



# Need for Beyond-the-Birthday Bound Security

- GCM, CCM, and OCB† are *limited* to birthday bound security

  AES-{GCM,CCM,OCB†} is secure up to $2^{64}$ queries

- $64$-bit security might be *insufficient*
  - exabyte-scale ($\simeq 2^{60}$) in use, zetabyte-scale ($\simeq 2^{70}$) expected
  - Limited *generic* multi-user security

- Standardise a bigger block cipher [an effective long term solution(?)]
  - Replacing AES-128 might not be viable
  - Noticeable setup time expected

- BBB secure (nonce-based) AEAD modes
  - CHM: full $n$-bit security
  - SCM: *graceful* degradation (*limited to $n/2$-bit security for arbitrary misuse*)
  - SIV$_r$: BBB nonce-misuse security (*highly inefficient*)

# Need for Beyond-the-Birthday Bound Security

- GCM, CCM, and OCB$^\dagger$ are *limited* to birthday bound security

  AES-{GCM,CCM,OCB$^\dagger$} is secure up to $2^{64}$ queries

- $64$-bit security might be *insufficient*
  - exabyte-scale $(\simeq 2^{60})$ in use, zetabyte-scale $(\simeq 2^{70})$ expected
  - Limited *generic* multi-user security

- Standardise a bigger block cipher [an effective long term solution(?)]
  - Replacing AES-128 might not be viable
  - Noticeable setup time expected [hardware support, general confidence]

- BBB secure (nonce-based) AEAD modes
  - CHM: full $n$-bit security
  - SCM: *graceful* degradation (*limited to $n/2$-bit security for arbitrary misuse*)
  - SIV$_r$: BBB nonce-misuse security (*highly inefficient*)

# Need for Beyond-the-Birthday Bound Security

- GCM, CCM, and OCB† are *limited* to birthday bound security

  AES-{GCM,CCM,OCB†} is secure up to $2^{64}$ queries

- $64$-bit security might be *insufficient*
  - exabyte-scale ($\simeq 2^{60}$) in use, zetabyte-scale ($\simeq 2^{70}$) expected
  - Limited *generic* multi-user security

- Standardise a bigger block cipher [an effective long term solution(?)]
  - Replacing AES-128 might not be viable
  - Noticeable setup time expected

- BBB secure (nonce-based) AEAD modes [CHM, SCM, $\mathrm{SIV}_r$, GCM-SIV, ΘCB, Romulus, LightOCB]
  - CHM: full $n$-bit security
  - SCM: *graceful* degradation (*limited to $n/2$-bit security for arbitrary misuse*)
  - $\mathrm{SIV}_r$: BBB nonce-misuse security (*highly inefficient*)

# Need for Beyond-the-Birthday Bound Security

- GCM, CCM, and OCB[†] are *limited* to birthday bound security

  AES-{GCM,CCM,OCB[†]} is secure up to $2^{64}$ queries

- $64$-bit security might be *insufficient*
  - exabyte-scale $(\simeq 2^{60})$ in use, zetabyte-scale $(\simeq 2^{70})$ expected
  - Limited *generic* multi-user security

- Standardise a bigger block cipher [an effective long term solution(?)]
  - Replacing AES-128 might not be viable
  - Noticeable setup time expected

- BBB secure (nonce-based) AEAD modes
  - CHM: full *n*-bit security (*insecure with a single misuse*)
  - SCM: *graceful* degradation (*limited to $n/2$-bit security for arbitrary misuse*)
  - $\text{SIV}_r$: BBB nonce-misuse security (*highly inefficient*)

# The Goal

To solve two problems:

- *Unique* nonce requirement
- *Limited* security (birthday bound)

# The Goal

To solve two problems:

- *Unique* nonce requirement
- *Limited* security (birthday bound)

*Design a block cipher-based efficient, misuse-resistant BBB-secure AEAD mode*

# Security Notions: A Quick Recap

## The Distinguishing Game

Real World (RW)

$$\begin{tikzcd} \mathcal{A} \arrow[r, shift left, "X_i"] & \mathcal{O}_{re} \arrow[l, shift left, "Y_i"] \end{tikzcd}$$

Ideal World (IW)

$$\begin{tikzcd} \mathcal{A} \arrow[r, shift left, "X_i"] & \mathcal{O}_{id} \arrow[l, shift left, "Y_i"] \end{tikzcd}$$

$$\mathbf{Adv}^{\text{game}}_{\mathcal{O}_{re}}(\mathcal{A}) := \left| \Pr\left(\mathcal{A} \text{ returns } 1 \text{ in RW}\right) - \Pr\left(\mathcal{A} \text{ returns } 1 \text{ in IW}\right) \right|$$

- Adversary's resources: $q$ (query), $\ell$ (max. length), $\sigma$ (total data) etc.
- Game: ideal world *functionality* + adversary's *power*

# Security Notions: A Quick Recap

## Pseudorandom Function (PRF)

RW

$$\begin{tikzcd} \mathcal{A} \arrow[r, shift left, "X_i"] & \mathsf{F}_K \arrow[l, shift left, "Y_i"] \end{tikzcd}$$

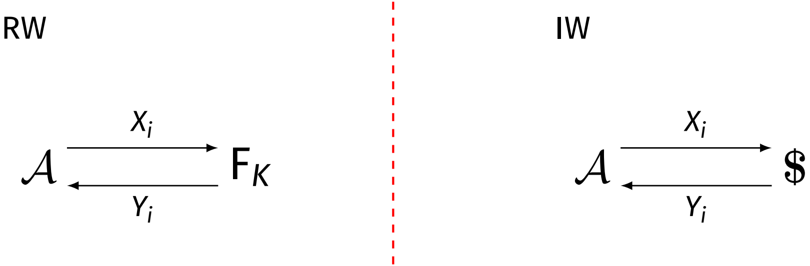
IW

$$\begin{tikzcd} \mathcal{A} \arrow[r, shift left, "X_i"] & \$ \arrow[l, shift left, "Y_i"] \end{tikzcd}$$

- Ideal world: a uniform random function $\$$
- $\mathcal{A}$ makes *chosen* plaintext queries
- $\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{F}}(\mathcal{A})$: the PRF advantage of $\mathcal{A}$ against F

# Security Notions: A Quick Recap

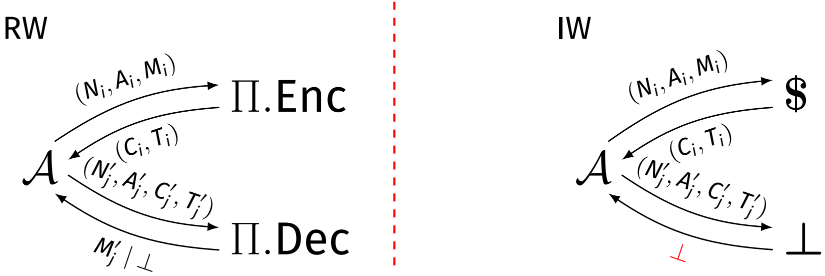
## Random IV-based PRF ($-PRF)




- Ideal world: a uniform random function $\$$
- $\mathcal{A}$ makes random plaintext queries
- $\mathbf{Adv}_{\mathsf{F}}^{\$\text{-prf}}(\mathcal{A})$: the $\$$-PRF advantage of $\mathcal{A}$ against F

# Security Notions: A Quick Recap

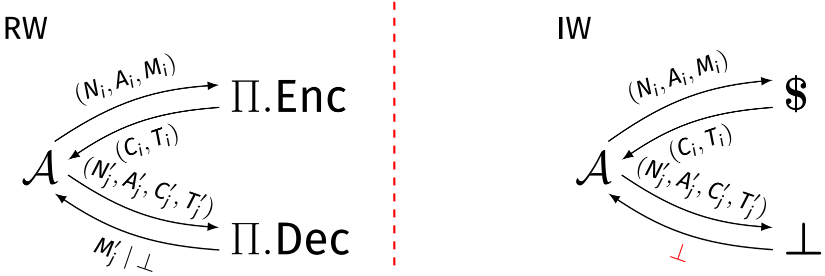
## Misuse-resistant AE (MRAE)




- Ideal world: a uniform random function $\$$ and the *reject* oracle $\perp$
- $\mathcal{A}$'s queries must satisfy $(N'_j, A'_j, C'_j, T'_j) \neq (N_i, A_i, C_i, T_i)$
- $\mathbf{Adv}^{\mathsf{mrae}}_{\Pi}(\mathcal{A})$: the MRAE advantage of $\mathcal{A}$ against $\Pi$
- DAEADs achieve MRAE security naturally!

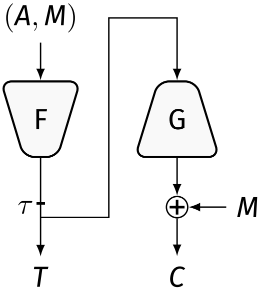
# Security Notions: A Quick Recap

## Misuse-resistant AE (MRAE)




- Ideal world: a uniform random function $\$$ and the *reject* oracle $\perp$
- $\mathcal{A}$'s queries must satisfy $(N'_j, A'_j, C'_j, T'_j) \neq (N_i, A_i, C_i, T_i)$
- $\mathbf{Adv}_{\Pi}^{\mathsf{mrae}}(\mathcal{A})$: the MRAE advantage of $\mathcal{A}$ against $\Pi$
- DAEADs achieve MRAE security naturally!

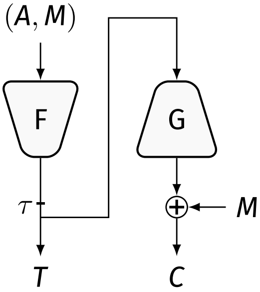
# Synthetic IV [RS: EC '06]




- Two main components:
  - F: a PRF
  - G: a random IV-based PRF
- Inverse-free
- Parallelizable
- Composition Bound [RS: EC '06, IM: ToSC '16]:

$$\mathbf{Adv}^{\mathsf{mrae}}_{\mathsf{SIV}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{prf}}_{\mathsf{F}}(\mathcal{B}) + \mathbf{Adv}^{\$\text{-}\mathsf{prf}}_{\mathsf{G}}(\mathcal{C}) + \frac{q}{2^{\tau}}$$

# Synthetic IV [RS: EC '06]




- Two main components:
  - F: a PRF
  - G: a random IV-based PRF
- Inverse-free
- Parallelizable
- Composition Bound [RS: EC '06, IM: ToSC '16]:

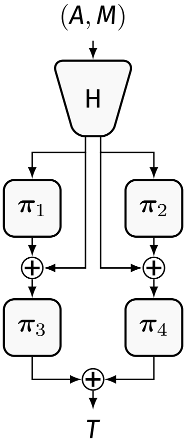
$$\mathbf{Adv}_{\mathsf{SIV}}^{\mathsf{mrae}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathsf{F}}^{\mathsf{prf}}(\mathcal{B}) + \mathbf{Adv}_{\mathsf{G}}^{\$\text{-}\mathsf{prf}}(\mathcal{C}) + \frac{q}{2^{\tau}}$$

TODOs:

1. A BBB secure PRF component with $\tau > n$ bits of output
2. A BBB secure random IV-based PRF component

# The PRF Component

## Revisiting **HtmB-p2** [CJN: AC '20]




- Hashing solves two purposes:
  - Handling arbitrary length inputs
  - Inputs to $\pi_{\{1,2\}}$ have *controlled collisions*

  $\implies$ Optimal Security for HtmB

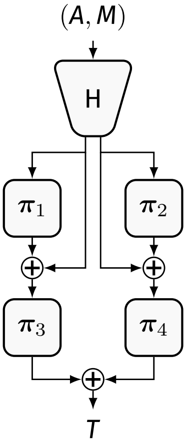
- HtmB-p2 PRF Bound [CJN: AC '20, CDNPS EC '23]:

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{HtmB\text{-}p2}}(\mathcal{A}) = O\left(\frac{q}{2^n} + q^2\epsilon_{coll}\right)$$

- Limitation: only $n$-bit outputs

# The PRF Component

## Revisiting **HtmB-p2** [CJN: AC '20]




- Hashing solves two purposes:
  - Handling arbitrary length inputs
  - Inputs to $\pi_{\{1,2\}}$ have *controlled collisions*

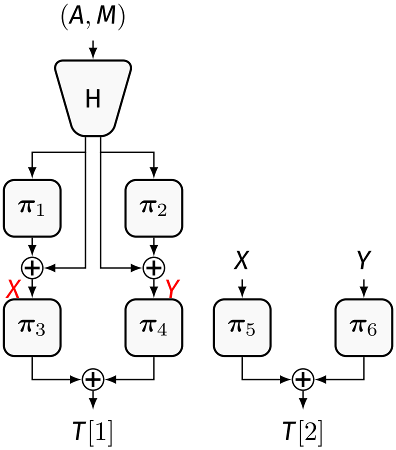
    $\Longrightarrow$ Optimal Security for HtmB
- HtmB-p2 PRF Bound [CJN: AC '20, CDNPS EC '23]:

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{HtmB\text{-}p2}}(\mathcal{A}) = O\left(\frac{q}{2^n} + q^2 \epsilon_{coll}\right)$$

- Limitation: only $n$-bit outputs

# The PRF Component

## $\mathsf{F}^*$: A BBB secure PRF with $2n$-bit outputs



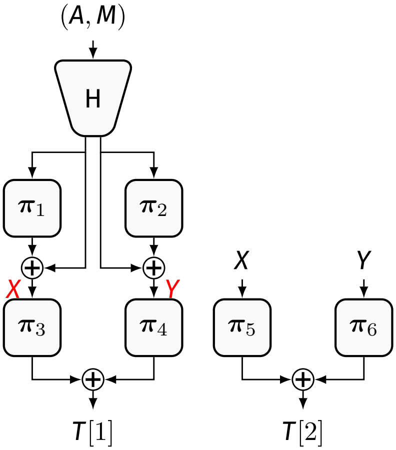

- HtmB-p2*:
  - Duplicates the HtmB-p2 finalization
  - Additional $n$ bits at the cost of two calls
- $\mathsf{F}^*$: HtmB-p2* with a PMAC+ like hash

$\mathsf{F}^*$ is optimally secure [for lengths up to $\sqrt{2^n}$]

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{F}^*}(\mathcal{A}) = O\left(\frac{\sigma}{2^n}\right)$$

# The PRF Component

## F$^*$: A BBB secure PRF with $2n$-bit outputs



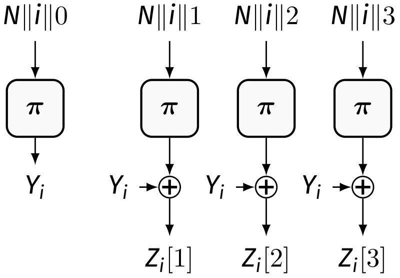

- HtmB-p2$^*$:
  - Duplicates the HtmB-p2 finalization
  - Additional $n$ bits at the cost of two calls
- F$^*$: HtmB-p2$^*$ with a PMAC+ like hash

**F$^*$ is optimally secure** [for lengths up to $\sqrt{2^n}$]

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{F}^*}(\mathcal{A}) = O\left(\frac{\sigma}{2^n}\right)$$

# The Random-IV PRF Component (Option 1)

## Revisiting CENC [Iwata: FSE '06]

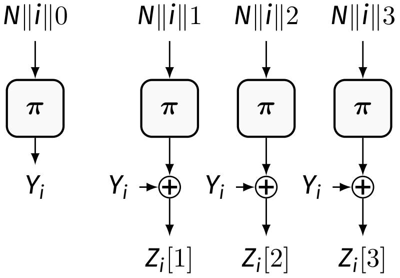



The $i$-th chunk of keystream (r=3)

- Keystream is generated in chunks of $r$ blocks
- Fully parallelizable
- Rate $\approx \left(\frac{r}{r+1}\right)$
- Optimally secure if IVs are *unique* [IMV: ePrint '16]
- Limitations:
  - $|N| < n$ (we require $\approx 2n$)
  - Only *birthday-bound* \$-PRF secure

# The Random-IV PRF Component (Option 1)

## Revisiting CENC [Iwata: FSE '06]

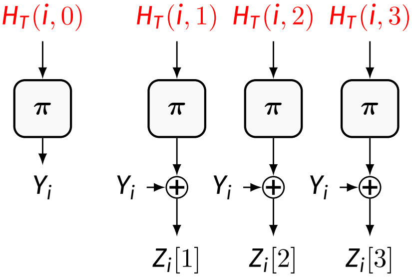



The $i$-th chunk of keystream (r=3)

- Keystream is generated in chunks of $r$ blocks
- Fully parallelizable
- Rate $\approx \left(\frac{r}{r+1}\right)$
- Optimally secure if IVs are *unique* [IMV: ePrint '16]
- Limitations:
  - $|N| < n$ (we require $\approx 2n$)
  - Only *birthday-bound* \$-PRF secure

# The Random-IV PRF Component (Option 1)

## GiantStar: A BBB secure random IV-based PRF




The $i$-th chunk of keystream (r=3)

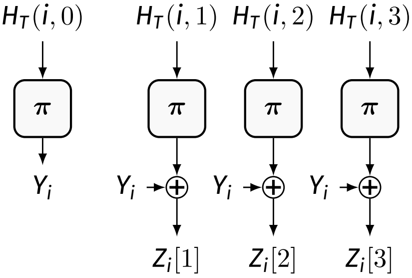
- CTR-based encoding $\rightarrow$ *lightweight* hash
  - Use the random IV as key
- Inherits all the the efficiency traits of CENC
- Secure if hash is $2$-wise independent

**GiantStar is BBB secure** [for *moderately* large $\ell$]

$$\mathbf{Adv}^{\text{\$-prf}}_{\text{GiantStar}}(\mathcal{A}) = O\left(\frac{r\sigma}{2^n} + \frac{r\sigma^2\ell}{2^{2n}}\right)$$

# The Random-IV PRF Component (Option 1)

## GiantStar: A BBB secure random IV-based PRF




The $i$-th chunk of keystream (r=3)

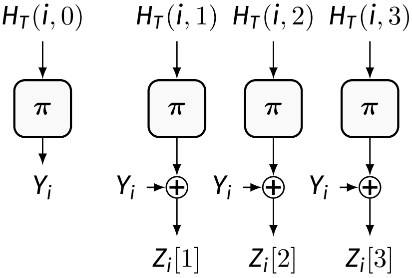
- CTR-based encoding $\rightarrow$ *lightweight* hash
  - Use the random IV as key
- Inherits all the the efficiency traits of CENC
- Secure if hash is $2$-wise independent

**GiantStar is BBB secure** [for *moderately* large $\ell$]

$$\mathbf{Adv}^{\$\text{-prf}}_{\text{GiantStar}}(\mathcal{A}) = O\left(\frac{r\sigma}{2^n} + \frac{r\sigma^2\ell}{2^{2n}}\right)$$

# The Random-IV PRF Component (Option 1)

## GiantStar: A BBB secure random IV-based PRF




The $i$-th chunk of keystream (r=3)

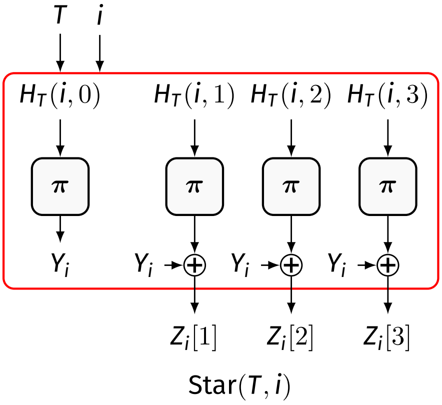
- CTR-based encoding $\rightarrow$ *lightweight* hash
  - Use the random IV as key
- Inherits all the the efficiency traits of CENC
- Secure if hash is $2$-wise independent

**GiantStar is BBB secure** [for *moderately* large $\ell$]

$$\mathbf{Adv}^{\$\text{-prf}}_{\text{GiantStar}}(\mathcal{A}) = O\left(\frac{r\sigma}{2^n} + \frac{r\sigma^2\ell}{2^{2n}}\right)$$

# The Random-IV PRF Component (Option 2)

## **Star:** A fixed-length BBB secure random IV-based PRF



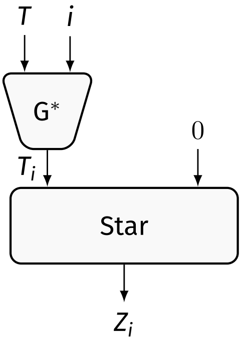

- Star $\equiv$ GiantStar with
  - Fixed chunk index $i$
  - Restricted to $\leq r$-block outputs

**Star is optimally secure**

$$\mathbf{Adv}^{\$\text{-prf}}_{\mathsf{Star}}(\mathcal{A}) = O\left(\frac{rq}{2^n}\right)$$

# The Random-IV PRF Component (Option 2)

## Snowflake: A length-independent BBB secure random IV-based PRF




The $i$-th chunk of keystream

- Fresh $2n$-bit randomness per chunk

$$\mathbf{Adv}^{\$\text{-prf}}_{\text{Snowflake}}(\mathcal{A}) \leq \mathbf{Adv}^{\$\text{-prf}}_{\text{Star}}(\mathcal{B}) + \mathbf{Adv}^{\$\text{-prf}}_{\text{G}^*}(\mathcal{C})$$

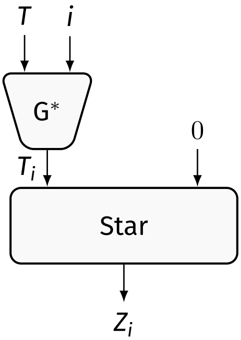
- $\text{G}^*$ must have *length-independent* bound!
- $\text{G}^*$ can be relatively heavier
  - in the paper: $6$ calls per chunk

**Snowflake is optimally secure**

$$\mathbf{Adv}^{\$\text{-prf}}_{\text{Snowflake}}(\mathcal{A}) = O\left(\frac{r\sigma}{2^n}\right)$$

# The Random-IV PRF Component (Option 2)

## Snowflake: A length-independent BBB secure random IV-based PRF




The $i$-th chunk of keystream

- Fresh $2n$-bit randomness per chunk

$$\mathbf{Adv}^{\$\text{-prf}}_{\mathsf{Snowflake}}(\mathcal{A}) \leq \mathbf{Adv}^{\$\text{-prf}}_{\mathsf{Star}}(\mathcal{B}) + \mathbf{Adv}^{\$\text{-prf}}_{\mathsf{G}^*}(\mathcal{C})$$

- $\mathsf{G}^*$ must have *length-independent* bound!
- $\mathsf{G}^*$ can be relatively heavier
  - in the paper: $6$ calls per chunk

**Snowflake is optimally secure**

$$\mathbf{Adv}^{\$\text{-prf}}_{\mathsf{Snowflake}}(\mathcal{A}) = O\left(\frac{r\sigma}{2^n}\right)$$

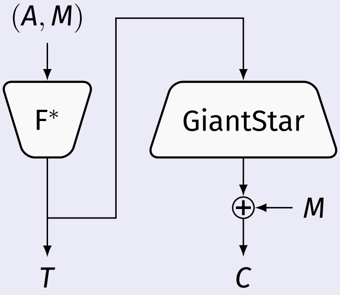
# Our Contributions

## Two misuse-resistant BBB-secure AEAD modes

# Our Contributions

## Two misuse-resistant BBB-secure AEAD modes

### DENC1



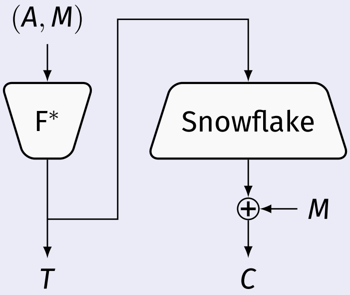

- Highly *parallelizable*
- Tag size $\tau = 2n$-bit
- Max. input length $\ell \leq \sqrt{2^n}$-block
- Rate $\geq \left(\frac{r}{2r+0.5}\right)$ ($\approx 0.498$ for $r = 64$)
- BBB secure for *moderate* message lengths

$$\mathbf{Adv}^{\mathsf{mrae}}_{\mathsf{DENC1}}(\mathcal{A}) = O\left(\frac{r\sigma^2\ell}{2^{2n}}\right)$$

# Our Contributions

## Two misuse-resistant BBB-secure AEAD modes

### DENC2




- Highly parallelizable
- Tag size $\tau = 2n$-bit
- Max. input length $\ell \leq \sqrt{2^n}$-block
- Rate $\geq \left(\frac{r}{2r+3.5}\right)$ ($\approx 0.486$ for $r = 64$)
- Length-independent optimal security

$$\mathbf{Adv}^{\mathsf{mrae}}_{\mathsf{DENC1}}(\mathcal{A}) = O\left(\frac{r\sigma}{2^n}\right)$$

# Making GCM Great Again: Toward Full Security and Longer Nonces

Woohyuk Chung[1] Seongha Hwang[1] Seongkwang Kim[2]
**Byeonghak Lee**[2] Jooyoung Lee[1]

[1]KAIST, Korea [2]Samsung SDS, Korea

Eurocrypt 2025

2025. 05. 06.

# Same Motivation, Different Goal

Recall: We require BBB-secure AEAD with low nonce misusing risk.

1. Design a misuse-resistant AE
   - AES-GCM-SIV, DENC1, DENC2, ...
   - Best for security, but inherently two pass
2. Design a nonce-based AE with extended nonces
   - DNDK-GCM: requires carefully generated nonces and BC with 2n-bit key

# Same Motivation, Different Goal

Recall: We require BBB-secure AEAD with low nonce misusing risk.

1. Design a misuse-resistant AE
    - AES-GCM-SIV, DENC1, DENC2, ...
    - Best for security, but inherently two pass
2. Design a nonce-based AE with extended nonces
    - DNDK-GCM: requires carefully generated nonces and BC with 2n-bit key

# Same Motivation, Different Goal

Recall: We require BBB-secure AEAD with low nonce misusing risk.

1. Design a misuse-resistant AE
   - AES-GCM-SIV, DENC1, DENC2, ...
   - Best for security, but inherently two pass
2. Design a nonce-based AE with extended nonces
   - DNDK-GCM: requires carefully generated nonces and BC with 2n-bit key

Our Goal:

- Block cipher based AE with full security
  \+ Provably secure under standard PRP assumption
- Efficiency is comparable to GCM
- Support extended nonces or provide nonce misuse resistance
- Support arbitrary length message

# Starting Point: CENC

Cipher-based ENCryption (CENC)

- CTR-type encryption mode with full security

$$\mathrm{Adv}^{\mathrm{prf}}_{\mathrm{CENC}[E,r]}(q,\sigma,l) \leq O\left(\frac{\sigma}{2^n}\right)$$

- limitation: $|\text{nonce}| + |\text{counter}| \leq n$
  $\Rightarrow$ still have nonce misusing risk and short length limitation

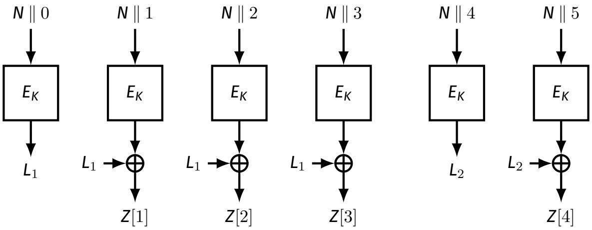



Figure: The first $4$ keystream blocks from $\mathrm{CENC}[E_K, w](N, \cdot)$ with $w = 3$.

# Building Blocks - eCTR

enhanced CTR (eCTR) ($\simeq$ GiantStar !)

- almost fully secure variable output length PRF (VOL-PRF) with 2n-bit random IV

$$\mathbf{Adv}^{\$\text{-prf}}_{\text{eCTR}[E,r]}(\mathcal{A}) \leq O\left(\frac{r\sigma}{2^n} + \frac{r\sigma^2 l}{2^{2n}}\right)$$

- limitation: requires random IV
  $\Rightarrow$ enough for iv-based AE, but we want nonce-based

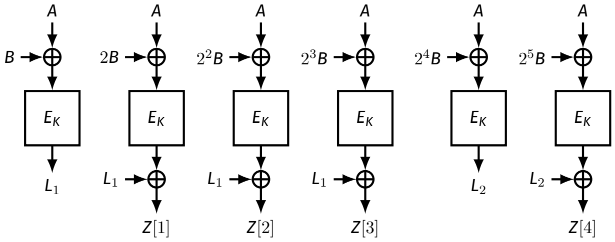



Figure: The first $4$ blocks from eCTR$[E_K, w](A, B)$ with $w = 3$.

# Building Blocks - HteC

Hash-then-eCTR (HteC)

- almost fully secure variable input/output length PRF (VIL-VOL-PRF)

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{HteC}[H,E,w]}(\mathcal{A}) \le O\left(\frac{w\sigma}{2^n} + \frac{w\sigma^2 l}{2^{2n}}\right)$$

  where $H$ is $\delta$-universal hash (UH)
- UH-then-PRP outputs $(= A, B)$ are not fully random but enough for eCTR input

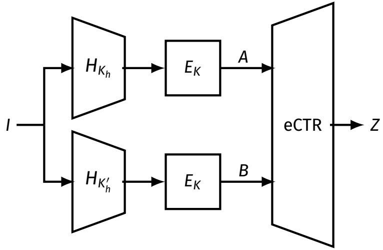
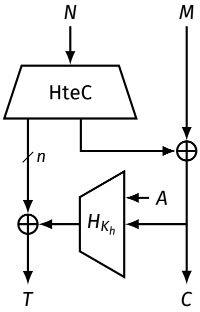


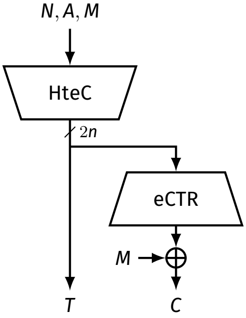

Figure: The HteC VIL-VOL pseudorandom function.

# Our Contribution

eGCM/eGCM-SIV: enhanced variant of GCM/GCM-SIV




(a) Encryption of eGCM

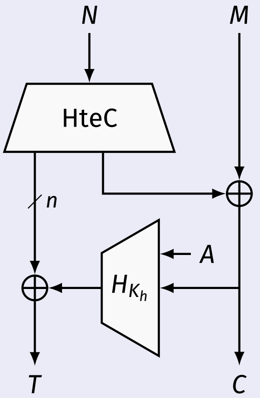



(b) Encryption of eGCM-SIV

Figure: The eGCM and eGCM-SIV AE schemes. A nonce, an associated data, and a message are denoted *N*, *A* and *M*, respectively

# Our Contribution

## eGCM

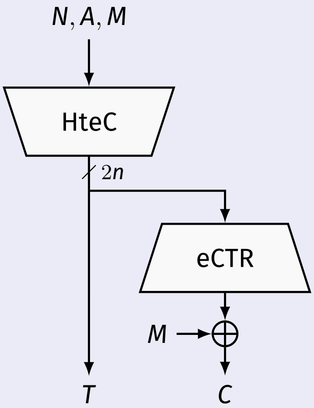



- Replace CTR to HteC
- Support extended nonces
- Support encrypting arbitrary length messages
- BBB secure for *moderate* message lengths

$$\mathrm{Adv}^{\mathrm{nae}}_{\mathrm{eGCM}}(\mathcal{A}) = O\left(\frac{r\sigma^2\ell}{2^{2n}}\right)$$

# Our Contribution

## eGCM-SIV




- Use HteC as PRF and replace CTR to eCTR
- Support encrypting arbitrary length messages
- BBB secure for *moderate* message lengths

$$\mathsf{Adv}^{\mathsf{dae}}_{\mathsf{eGCM\text{-}SIV}}(\mathcal{A}) = O\left(\frac{r\sigma^2\ell}{2^{2n}}\right)$$

# Comparison

| AEAD | Rate | Security | |
|---|---|---|---|
| | | NR | NM |
| OCB3 | 1 | $n/2$ | - |
| GCM | 1/2 | $n/2$ | - |
| CIP, CHM, mGCM, eGCM | $\lesssim 1/2^{\dagger}$ | $n$ | - |
| AES-GCM-SIV | 1/2 | $n$ | $n/2$ |
| SCM | 1/2 | $n$ | $n/2$ |
| CWC+ | $\lesssim 1/2^{\dagger}$ | $3n/4$ | $n/2$ (auth only) |
| eGCM-SIV, DENC1, DENC2 | $\lesssim 1/2^{\dagger}$ | $n$ | $n$ |

‡ Depends on the parameter $w$, while we write $\lesssim 1/2$ since the rate approaches $1/2$ as $w$ increases and $w$ can be set to a large enough value.

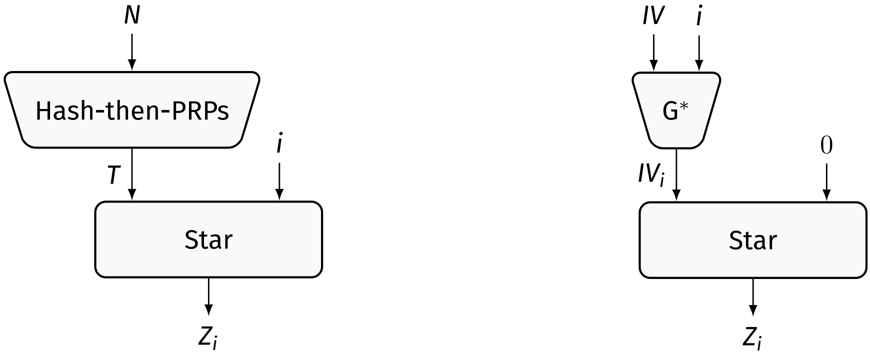
Table: Comparion of eGCM, eGCM-SIV, DENC1 and DENC2 and other block cipher based AE schemes. The maximum message length $(= l)$ is assumed to be a small constant. Note that DENC2 has length-independent security.

# Benchmark

| AEAD | Message | | |
|---|---|---|---|
| | 1KB | 4KB | 64KB |
| OCB3 | 0.52 | 0.47 | 0.45 |
| GCM | 1.65 | 1.02 | 0.83 |
| eGCM | 0.93 | 0.89 | 0.88 |
| AES-GCM-SIV | 1.33 | 1.07 | 0.99 |
| SCM | 1.19 | 1.11 | 1.07 |
| eGCM-SIV | 1.33 | 1.15 | 1.12 |
| DENC1 | 1.31 | 1.20 | 1.18 |
| DENC2 | 1.42 | 1.38 | 1.32 |

Table: Benchmark of eGCM, eGCM-SIV, DAE1 and DAE2 and other block cipher based AE schemes. Throughput is measured in cycles per byte, for empty associated data.

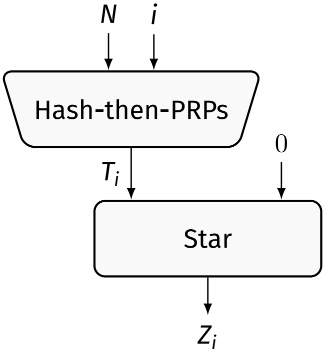
# HteC vs SnowFlake




- Use arbitrary length nonces
- Simpler compressing function
- Length-dependent security

- Use random IVs (enough for SIV)
- Length-independent security
- $G^*$ is heavy!

# Combining Two Papers: **HteC + SnowFlake**




- G* is replaced by Hash-then-PRPs $\Rightarrow$ faster and support nonce!
- VIL-VOL-PRF with (output) length-independent security
- Can be used to construct fully secure NAE and DAE

# Conclusion

Towards Optimally Secure DAEs

- DENC1: almost fully secure DAE
- DENC2: fully secure DAE (length-independent security)

Making GCM Great Again

- HteC: almost fully secure VIL-VOL-PRF
- eGCM: almost fully secure NAE with extended nonces
- eGCM-SIV: almost fully secure DAE

Our results can also be applied to:

- Accordion ciphers: Hash-CTR-Hash $\Rightarrow$ Hash-(eCTR/SnowFlake)-Hash
- Nonce-key derivation: HteC and HteC+SnowFlake are PRF

# Thank you for your attention!