The Committing Security of MACs with Applications to Generic Composition

Ritam Bhaumik Bishwajit Chakraborty Wonseok Choi Avijit Dutta
Jérôme Govinden Yaobin Shen

CRYPTO 2024

Committing Security

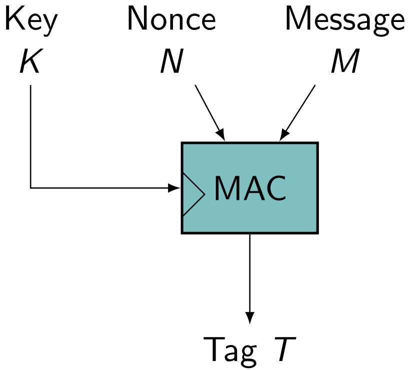
Committing Security

- Notion originating from the public-key encryption literature [ABN10]
- Adapted to symmetric primitives by Farshim et al. [FOR17]
- Latest research focuses on authenticated encryption with associated data (AEAD)
  - **Vulnerable settings:** moderation in encrypted messaging apps, key rotation mechanisms, password-based encryption, etc. [GLR17, DGRW18, LGR21, ADG+22]
  - **Vulnerable schemes:** AES-GCM, AES-GCM-SIV, ChaCha20-Poly1305, OCB3, CCM, EAX, SIV [LGR21, ADG+22, MLGR23]

→ Almost all standardized AEAD are vulnerable to committing attacks

What about MACs (Message Authentication Codes)?

- Provide only authentication
- Many MAC standards based on:
  - Universal hash functions
  - Block ciphers
  - Hash functions
  - Permutations
- Used in a wide variety of scenarios:
  - Message authentication and integrity checks
  - Authentication protocols and authenticated encryption schemes
  - Pseudorandom functions
  - Key derivation functions



→ Committing security scarcely studied for MACs!

Settings Requiring Committing MACs

Practical Applications of Committing MACs

- We found four practical settings needing committing security:
  - The OPAQUE Augmented PAKE Protocol
  - Authentication without key identification
  - Collision Resistant KDF
  - Timed Efficient Stream Loss-Tolerant Authentication (TESLA)

- Three of them implicitly assumed it to be guaranteed by their underlying MAC

The OPAQUE Augmented PAKE Protocol

- Password-authenticated key exchanges (PAKE) recommended by the CFRG
- Explicitly requires *random-key robustness*

$$\Pr_{K_1,K_2 \leftarrow\$}[M \leftarrow \mathcal{A}(K_1, K_2)\,;\, \mathsf{MAC}(K_1, M) = \mathsf{MAC}(K_2, M)] \leq \epsilon$$

$\rightarrow$ we capture this property in the MAC key-committing notion CMT_k

- Proposed instantiation by HMAC

Key Derivation Function (KDF)

- HMAC, CMAC, and KMAC are recommended in NIST SP800-108r1 for sub-keys derivation
- Used to derive a sub-key $K_{\mathsf{OUT}} = \mathsf{MAC}(K_{\mathsf{IN}}, \mathsf{Ctx})$ from a key K_{IN} and a context Ctx

NIST SP800-108r1

> *If those parties have different understandings, then they will derive different keying material.*

$\rightarrow$ not guaranteed by standard MAC security
$\rightarrow$ we capture this property in the MAC context-committing notion CMT

- CMAC is not context-committing

Timed Efficient Stream Loss-Tolerant Authentication (TESLA)

- Provide sender authentication for broadcast streams and defined in RFC 4082
- Uses a one-way chain:

Generate (←)

$$K_0 \xleftarrow{F(K_1)} K_1 \xleftarrow{F(K_2)} \cdots \quad \cdots \xleftarrow{F(K_\ell)} K_\ell \xleftarrow{F(K_{\ell+1})} K_{\ell+1}$$

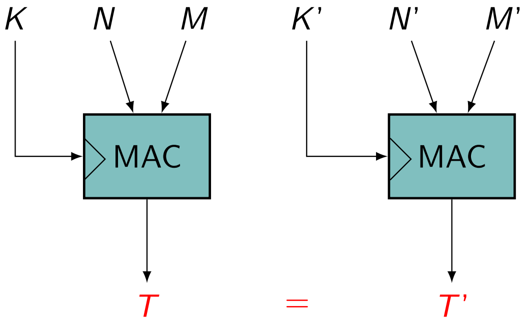
Reveal (→)

- F is defined as $F(K) = \mathrm{MAC}(K, 0)$
- Given K_i, it should be hard to find x such that $F(x) = K_i$

$\rightarrow$ not guaranteed by standard MAC security
$\rightarrow$ we capture this property in the MAC context-discovery notion CDY

Committing and CDY Security Notions for MACs

MACs Committing Security (CMT)



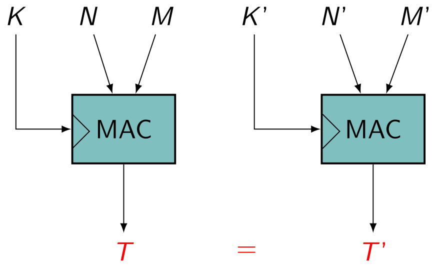
Notion	Requirement
CMT_k	$K \neq K'$
CMT	$(K, N, M) \neq (K', N', M')$

← Key Commitment

← Context Commitment

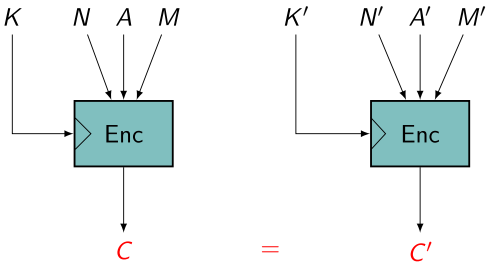
MACs Committing Security (CMT)

MAC



Notion	Requirement
CMT_k	$K \neq K'$
CMT	$(K, N, M) \neq (K', N', M')$

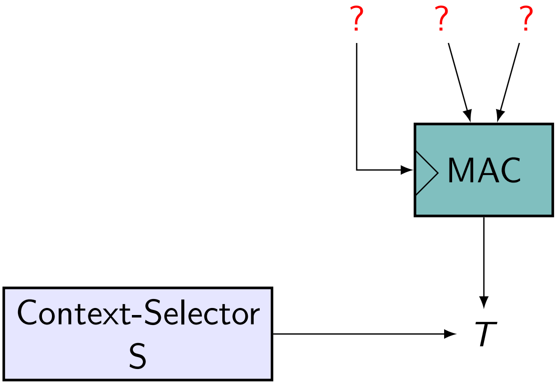
AEAD



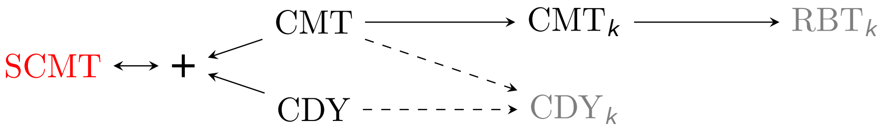
Notion	Requirement	[BH22]
CMT_k	$K \neq K'$	← CMT-1
CMT	$(K, N, A) \neq (K', N', A')$	← CMT-3

MACs Context-Discovery Security (CDY)

→ Adaptation of the context-discovery notion for AEAD from Menda et al. [MLGR23]



Relations between Commitment and Context-Discovery Notions



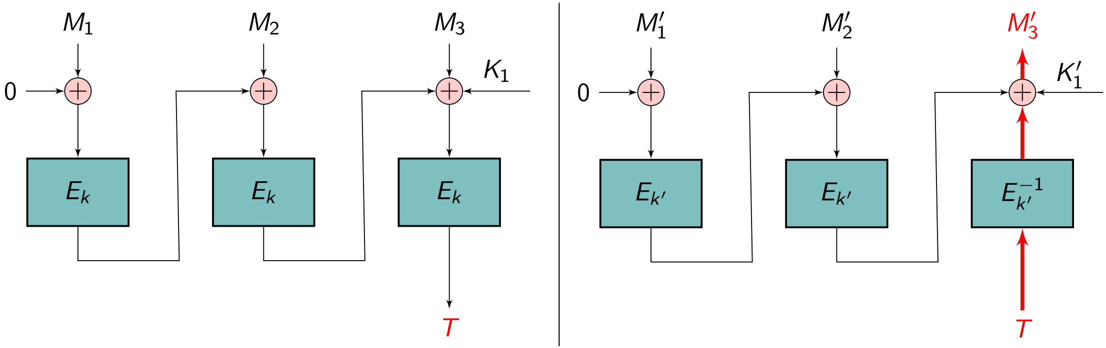
Security Analysis of Standardized MACs

Summary Table

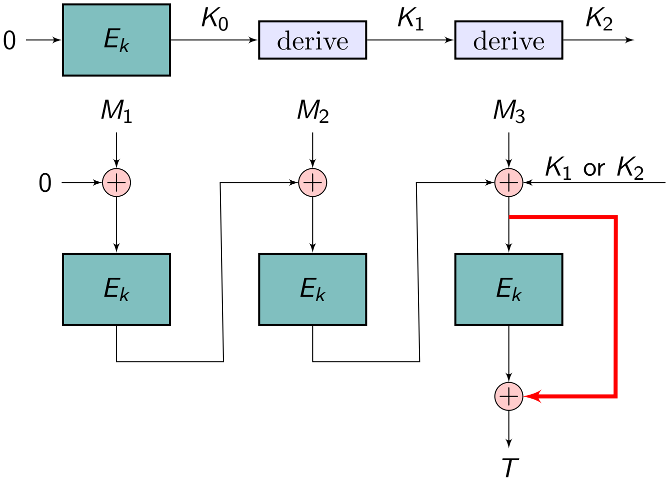
Scheme	CMT_k	CMT	CDY
CBC-type MACs	no	no	no
HMAC with variable-length keys	no	no	?
Badger	no	no	no
Poly1305-AES	no	no	no
GMAC	no	no	no
LightMAC	no	no	no
Chaskey	no	no	no
CBC-MAC-C1 [this work]	yes	no	yes
CMAC-C1 [this work]	yes	no	yes
HMAC with fixed-length keys	yes	yes	yes

Key-Committing Attack on CMAC

$\rightarrow$ Choose the values k, k' such that $k \neq k'$



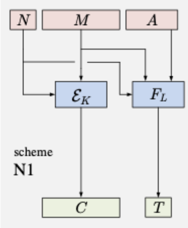
CMAC-C1: a Key-Committing Secure Variant of CMAC



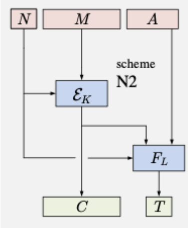
Applications to Generic Composition

Generic Composition Paradigms [NRS14]

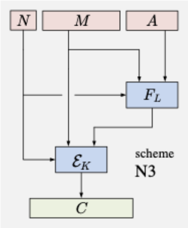
Encrypt-and-MAC (EaM)



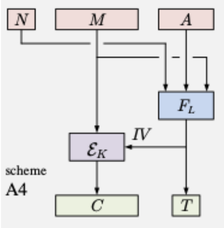
Encrypt-then-MAC (EtM)



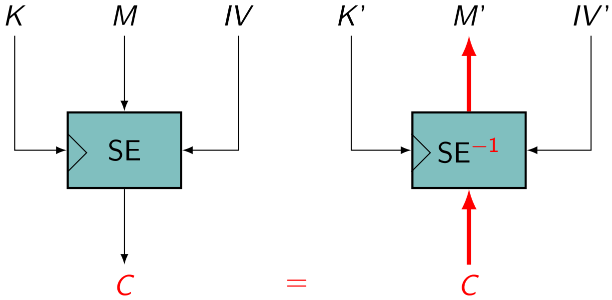
MAC-then-Encrypt (MtE)



SIV



Key-Committing Insecurity of IV-Based Symmetric Encryption



Generic Composition without Assumptions on IV-Based Encryption

If the MAC is CDY

$\rightarrow$

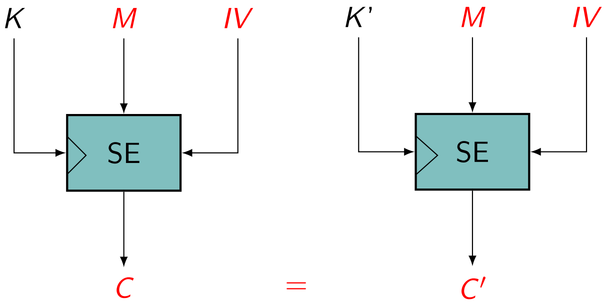
Encrypt-then-MAC, Encrypt-and-MAC and SIV are CDY

If the MAC is CMT_k or CMT

$\rightarrow$

Encrypt-then-MAC and Mac-then-Encrypt are not necessarily

Key-Robustness Security (RBT_k) of IV-Based Encryption



- CTR and CBC encryption mode are RBT_k
- If SE is RBT_k and the MAC is CMT $\rightarrow$ Encrypt-and-MAC and SIV are CMT
  $\rightarrow$ Encrypt-then-MAC and MAC-then-Encrypt are not

Generic Composition with a Key Schedule

- Keys for MAC and SE are derived from a single key with a Key Schedule function:

$$\mathsf{KS}(K) = (K_m, K_e)$$

If Key Schedule is COLL and the MAC is CMT_k, CMT or CDY

$\rightarrow$

Encrypt-then-MAC, Encrypt-and-MAC and SIV are CMT_k, CMT or CDY

Summary of Analyses

	MAC Assumption	SCMT	CDY	CDY_k	CMT	CMT_k
Scheme	SE/KS Assumption					
MtE	none	?	?	?	no	no
MtE	RBT_k	?	?	?	no	no
EtM	none	no	yes	yes	no	no
EtM	RBT_k	no	yes	yes	no	no
KEtM	COLL	yes	yes	yes	yes	yes
EaM	none	?	yes	yes	?	?
EaM	RBT_k	yes	yes	yes	yes	?
KEaM	COLL	yes	yes	yes	yes	yes
SIV	none	?	yes	yes	?	?
SIV	RBT_k	yes	yes	yes	yes	?
KSIV	COLL	yes	yes	yes	yes	yes

Future Work

- Analyze the remaining generic composition combinations
- Identify further settings requiring MAC commitment and CDY security
- Design efficient key/context-committing MACs
- Design MAC schemes with BBB committing security

Full version available on IACR ePrint:

https://ia.cr/2024/928

References I

Michel Abdalla, Mihir Bellare, and Gregory Neven.
Robust encryption.
In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 480–497. Springer, Heidelberg, February 2010.

Ange Albertini, Thai Duong, Shay Gueron, Stefan Kölbl, Atul Luykx, and Sophie Schmieg.
How to abuse and fix authenticated encryption without key commitment.
In Kevin R. B. Butler and Kurt Thomas, editors, *USENIX Security 2022*, pages 3291–3308. USENIX Association, August 2022.

Mihir Bellare and Viet Tung Hoang.
Efficient schemes for committing authenticated encryption.
In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 845–875. Springer, Heidelberg, May / June 2022.

References II

Yevgeniy Dodis, Paul Grubbs, Thomas Ristenpart, and Joanne Woodage.
Fast message franking: From invisible salamanders to encryptment.
In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 155–186. Springer, Heidelberg, August 2018.

Pooya Farshim, Claudio Orlandi, and Răzvan Roșie.
Security of symmetric primitives under incorrect usage of keys.
IACR Trans. Symm. Cryptol., 2017(1):449–473, 2017.

Paul Grubbs, Jiahui Lu, and Thomas Ristenpart.
Message franking via committing authenticated encryption.
In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 66–97. Springer, Heidelberg, August 2017.

Julia Len, Paul Grubbs, and Thomas Ristenpart.
Partitioning oracle attacks.
In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 195–212. USENIX Association, August 2021.

References III

Sanketh Menda, Julia Len, Paul Grubbs, and Thomas Ristenpart.
Context discovery and commitment attacks - how to break CCM, EAX, SIV, and more.
In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part IV*, volume 14007 of *LNCS*, pages 379–407. Springer, Heidelberg, April 2023.

Chanathip Namprempre, Phillip Rogaway, and Thomas Shrimpton.
Reconsidering generic composition.
In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 257–274. Springer, Heidelberg, May 2014.