Efficient Schemes for Committing Authenticated Encryption

Mihir Bellare[1][0000000287655573] and Viet Tung Hoang[2]

[1] Department of Computer Science and Engineering, University of California San Diego, USA. cseweb.ucsd.edu/~mihir/ mihir@eng.ucsd.edu
[2] Department of Computer Science, Florida State University, USA. cs.fsu.edu/~tvhoang/ tvhoang@cs.fsu.edu

Abstract. This paper provides efficient authenticated-encryption (AE) schemes in which a ciphertext is a commitment to the key. These are extended, at minimal additional cost, to schemes where the ciphertext is a commitment to all encryption inputs, meaning key, nonce, associated data and message. Our primary schemes are modifications of GCM (for basic, unique-nonce AE security) and AES-GCM-SIV (for misuse-resistant AE security) and add both forms of commitment without any increase in ciphertext size. We also give more generic, but somewhat more costly, solutions.

1 Introduction

Symmetric encryption is the canonical primitive of cryptography, with which the field is often identified in the popular mind. Over time, the primitive has evolved. Failures of privacy-only schemes lead to the understanding that the goal should be *authenticated* encryption [10, 32]. The underlying syntax, meanwhile, has gone from randomized or counter-based [7] to nonce-based [40, 39].

Recent attacks and applications [34, 27, 3, 25, 4] motivate another evolution. Namely, a ciphertext should be a commitment to the key, and beyond that, possibly even to other or all the inputs to the encryption process.

In this paper we contribute definitions and new schemes for such committing authenticated encryption. Our schemes combine efficiency, security and practicality attributes that may make them attractive for inclusion in cryptographic software libraries or for standardization.

Background. In a nonce-based symmetric encryption scheme SE, encryption takes key K, nonce N, associated data A and message M to deterministically return a ciphertext $C \leftarrow \mathsf{SE.Enc}(K, N, A, M)$, with decryption recovering via $M \leftarrow \mathsf{SE.Dec}(K, N, A, C)$ [40, 39]. AE security asks for both privacy and authenticity of the message. In its most basic form, called UNAE (Unique-Nonce AE security) this is under the assumption that nonces are unique, meaning never reused across encryptions [40, 39]. MRAE (Misuse-resistant AE security) is stronger, asking in addition for best-possible security under any reuse of an encryption nonce [41].

A central scheme is GCM [36]. It is a government standard [23] and is used in TLS [42]. Other standardized and widely-used schemes are XSalsa20/Poly1305 and ChaCha20/Poly1305 [17, 15, 16]. All these are UNAE-secure. AES-GCM-SIV [43, 28] is a leading MRAE scheme poised for standardization.

For both UNAE and MRAE, proofs are the norm, but the bar is now high: not only multi-user (mu) security [13] —reflecting that deployment settings like TLS have millions of users— but with bounds that are good, meaning almost the same as for the single-user setting. Dedicated analyses show that GCM has such UNAE security [13, 35, 29], and likewise for the MRAE security of AES-GCM-SIV [20]. Henceforth when we refer to UNAE or MRAE, it means in the mu setting

Committing security. We formalize, in a systematic way, different notions of what it means for a ciphertext $C \leftarrow \mathsf{SE.Enc}(K, N, A, M)$ to be a commitment. For the purposes of this Introduction, we can confine attention to two notions, CMT-1 and CMT-4. The primary, CMT-1 notion asks that the commitment be to the key K. In the game formalizing this, the adversary returns a pair $((K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2))$ satisfying $K_1 \neq K_2$, and is successful if $\mathsf{SE.Enc}(K_1, N_1, A_1, M_1) = \mathsf{SE.Enc}(K_2, N_2, A_2, M_2)$. Extending this, CMT-4 asks that the commitment be, not just to the key, but to K, N, A, M, meaning to *all* the inputs to SE.Enc. The game changes only in the requirement $K_1 \neq K_2$ being replaced by $(K_1, N_1, A_1, M_1) \neq (K_2, N_2, A_2, M_2)$. As a mnemonic, think of the integer ℓ in the notation CMT-ℓ as the number of inputs of SE.Enc to which we commit.

Clearly CMT-4 $\rightarrow$ CMT-1, meaning any scheme that is CMT-4-secure is also CMT-1-secure, and it is easy to see that the implication is strict. (There exist CMT-1-secure schemes that are not CMT-4-secure.)

In Section 3, we also consider CMT-3, simpler than, but equivalent to, CMT-4; we give alternative, decryption-based formulations of all these definitions but show the two equivalent for schemes that, like all the ones we consider, satisfy the syntactic requirement of tidiness [38]; and finally we extend the notions from 2-way committing security to s-way committing security for a parameter $s \geq 2$ which will enter results.

Simple counterexamples show that neither UNAE nor MRAE security imply even CMT-1-security. And the gap is real: attacks from [34, 4, 27] show that GCM, XSalsa20/Poly1305, ChaCha20/Poly1305 and OCB [40] are all CMT-1-insecure.

Prior notions. The notion of key-committing (KC) security, asking that a ciphertext is a commitment to the key, starts with Abdalla, Bellare and Neven (ABN) [3], who called it robustness and studied it for PKE and IBE. Their definitions were strengthened by Farshim, Libert, Paterson and Quaglia [24]. Now calling it key-robustness, Farshi, Orlandi and Rǫsie (FOR) [25] bring it to randomized symmetric encryption. Albertini, Duong, Gueron, Kölbl, Luykx and Schmieg (ADGKLS) [4] and Len, Grubbs and Ristenpart (LGR) [34] consider it for nonce-based symmetric encryption, giving definitions slightly weaker than CMT-1.

Grubbs, Lu and Ristenpart (GLR) [27] consider committing to the header and message. CMT-4 is stronger in that it asks for the commitment to be not just to these but also to the key and nonce. However, we do not consider or require what GLR [27] call compact commitment.

WHY COMMIT TO THE KEY? The canonical method for password-based encryption (PKCS#5 [31]) uses a symmetric encryption scheme SE, such as GCM, as a tool. In a surprising new attack, LGR [34] show that absence of key-committing (KC) security in SE leads to a break of the overlying password-based encryption scheme. This attack is circumvented if SE is CMT-1-secure.

Broadly, we have seen protocols failing due to absence of key-committing security in an underlying encryption scheme and then fixed by its being added. ABN [3] illustrate this when the protocol is PEKS [19]; they also note that when encryption strives to be anonymous, key-committing security is necessary for unambiguous decryption. FOR [25] illustrate the issue for an encryption-using Oblivious Transfer protocol and note that encryption not being key-committing has lead to attacks on Private Set Intersection protocols [33]. ADGKLS [4] describe in detail three real-world security failures —the domains are key rotation, envelope encryption and subscribe-with-Google— arising from lack of key-committing security.

WHY COMMIT TO EVERYTHING? CMT-4 is a simple, optimally-strong goal: we commit to everything. This means all 4 of the inputs to the encryption algorithm: key, nonce, associated data and message. Some motivation comes from applications; for example, GLR [27] show that committing to header and message is needed for an AE scheme to provide message franking, a capability in messaging systems that allows a receiver to report the receipt of abusive content. But the larger benefit is to increase ease of use and decrease risk of error or misuse. An application designer is spared the burden of trying to understand to exactly which encryption inputs the application needs a commitment; with CMT-4, she is covered.

PATH TO SCHEMES. Our starting points are existing AE schemes. Given one such, call it SE, we will modify it to a CMT-1 scheme SE-1 and then further into a CMT-4 scheme SE-4. These modifications must of course retain AE security: for $\mathrm{XX} \in \{\mathrm{UN}, \mathrm{MR}\}$, if SE is XXAE-secure then so are SE-1, SE-4. The ciphertext overhead (length of ciphertext in new scheme minus that in old) is kept as small as possible, and is zero for our primary schemes. Computational overhead will always be independent of the length of the message.

Proofs of AE security for our schemes are in the multi-user setting, with bounds as good as those for the starting schemes. This requires significant analytical effort.

Modern encryption standards are purely blockcipher based, meaning do not use a cryptographic hash function like SHA256; this allows them to most effectively exploit the AES-NI instructions for speed, and also lowers their real-estate in hardware. We aim, as much as possible, to retain this. For CMT-1, we succeed, reaching this without cryptographic hash functions. The extension to CMT-4

Scheme	AE security	Committing security	Ciphertext overhead	Starts from
CAU-C1	UNAE	CMT-1	0	GCM
HtE[CAU-C1, ·]	UNAE	CMT-4	0	GCM
CAU-SIV-C1	MRAE	CMT-1	0	AES-GCM-SIV
HtE[CAU-SIV-C1, ·]	MRAE	CMT-4	0	AES-GCM-SIV
UtC[SE, ·]	UNAE	CMT-1	1 block	any UNAE SE
HtE[UtC[SE, ·], ·]	UNAE	CMT-4	1 block	any UNAE SE
RtC[SE, ·, ·]	MRAE	CMT-1	1 block	any MRAE SE
HtE[RtC[SE, ·, ·], ·]	MRAE	CMT-4	1 block	any MRAE SE

Fig. 1. Summary of attributes of our schemes. Ciphertext overhead is length of ciphertext in our scheme minus that in the scheme from which it starts. Computational overhead is always independent of message length. A "·" as an argument to a transform refers to some suitable auxiliary primitive discussed in the text.

---

however requires a function H that we would instantiate via a cryptographic hash function.

The step from CMT-1 to CMT-4 is done via a general, zero ciphertext-overhead transform, called HtE, that we discuss next. Figure 1 summarizes the attributes of the different new schemes that we give and will discuss below.

From CMT-1 to CMT-4 via HtE. We give a generic way to turn a CMT-1 scheme into into a CMT-4 one. (That is, once you can commit to the key, it is easy to commit to everything.) The transform incurs no ciphertext overhead and preserves both UNAE and MRAE security. The computational overhead involves processing only the nonce and associated data, and is independent of message length.

We now give some detail. Given a symmetric encryption scheme SE-1, and a function H, our HtE (Hash then Encrypt) transform defines the scheme SE-4 $\leftarrow$ HtE[SE-1, H] in which SE-4.Enc(K, N, A, M) lets $L \leftarrow H(K, (N, A))$ and returns SE-1.Enc(L, N, ε, M). Here outputs of H have the same length as keys of SE-1. There is no ciphertext overhead: ciphertexts in SE-4 have the same length as in SE-1. The computational overhead, namely the computation of H, is independent of message length. Theorem 1 shows that SE-4 is CMT-4 assuming SE-1 is CMT-1 and H is collision resistant. Theorem 2 shows that if H is a PRF then (1) If SE-1 is UNAE then so is SE-4, and (2) If SE-1 is MRAE then so is SE-4. All these results are with good bounds.

We stress that we avoid assuming H is a random oracle; we instead make the standard-model assumption that it is a collision-resistant PRF. Section 3 discusses instantiations of H based on HMAC [5], SHA256 or SHA3.

CAU schemes. GCM [36] is a UNAE scheme that, due to its standardization [23] and use in TLS [42], is already widely implemented. Attacks [34, 4, 27] however show that it is not CMT-1-secure. Making only a tiny modification to GCM,

we obtain a new scheme, that we CAU-C1, that is UNAE and CMT-1 secure. Theorem 3 establishes CMT-1 security of CAU-C1, and Theorem 4 establishes UNAE security with good mu bounds.

CAU-C1 changes only how the last block GCM block is encrypted so that the tag is a Davies-Meyer hash. (See Figure 9.) The locality and minimality of the change means that it should be easy to modify existing GCM code to obtain CAU-C1 code, making CAU-C1 attractive for implementation. With regard to performance, CAU-C1 incurs essentially no overhead; in particular, the ciphertext size remains the same as in GCM.

We can obtain a UNAE and CMT-4-secure scheme, that we call CAU-C4, by applying our above-discussed HtE transform to CAU-C1 and a suitable collision-resistant PRF H. Ciphertext overhead continues to be zero: CAU-C4 ciphertexts have the same size as CAU-C1, and thus GCM, ones.

With the above, we have obtained CMT-1 and CMT-4 UNAE schemes that offer minimal overhead, good quantitative security and ease of implementation. We now turn to MRAE, doing the same. Here our starting point is AES-GCM-SIV [43, 28], a leading MRAE scheme poised for standardization. We give CAU-SIV-C1, a tiny modification of AES-GCM-SIV that is MRAE and CMT-1-secure. Theorem 5 establishes CMT-1 security of CAU-SIV-C1, and Theorem 6 establishes MRAE security with good mu bounds. Again, applying HtE to CAU-SIV-C1 yields a MRAE and CMT-4 scheme CAU-SIV-C4 that continues to be a small modification of AES-GCM-SIV. There is no growth in ciphertext size.

Generic transforms. With the four schemes discussed above, we have obtained CMT-1 and CMT-4 security for both UNAE and MRAE schemes, with zero ciphertext overhead and almost zero computational overhead. These schemes however are intrusive, making small modifications to GCM or AES-GCM-SIV. We now give ways to add committing security via generic transforms that invoke the given scheme only in a blackbox way. The price we will pay is some ciphertext overhead.

We give a generic transform UtC that takes any UNAE scheme SE and returns a scheme $\overline{\mathsf{SE}} \leftarrow \mathsf{UtC}[\mathsf{SE}, \mathsf{F}]$ that is UNAE and CMT-1-secure. Here F is a committing PRF, a primitive we introduce that generalizes the notion of a key-robust PRF from FOR [25]. We build a cheap committing PRF, that we call CX, from (only) a blockcipher. Proposition 5 proves its security with good bounds. Theorem 7 establishes CMT-1 security of $\overline{\mathsf{SE}}$, and also shows that $\overline{\mathsf{SE}}$ inherits the mu UNAE security of SE without degradation in the bound. Ciphertexts in $\overline{\mathsf{SE}}$ are one block longer than those in SE. Applying HtE to $\overline{\mathsf{SE}}$ and a suitable collision-resistant PRF H, we obtain a UNAE CMT-4 scheme, leaving ciphertext overhead at one block.

UtC however does not preserve MRAE security. We give a second generic transform, RtC, that takes any MRAE scheme SE and returns a scheme $\overline{\mathsf{SE}} \leftarrow \mathsf{RtC}[\mathsf{SE}, \mathsf{F}, \mathsf{H}]$ that is MRAE and CMT-1-secure. Here F as before is a committing PRF that we set to CX, and H is a collision-resistant PRF that we instantiate via the Davies-Meyer method. Theorem 8 establishes CMT-1 security of $\overline{\mathsf{SE}}$, and also shows that $\overline{\mathsf{SE}}$ inherits the mu MRAE security of SE without degradation

in the bound. Ciphertexts in $\overline{\mathsf{SE}}$ are one block longer than those in SE. Again, applying HtE to $\overline{\mathsf{SE}}$ yields a MRAE CMT-4 scheme, leaving ciphertext overhead at one block.

Extensions and remarks. For an integer parameter $s \geq 2$, we can extend CMT-1 to a notion CMT$_s$-1 of multi-input committing security. Here the adversary returns an s-tuple $((K_1, N_1, A_1, M_1), \ldots (K_s, N_s, A_s, M_s))$ in which $K_1, \ldots, K_s$ are all distinct, and is successful if $\mathsf{SE.Enc}(K_1, N_1, A_1, M_1), \ldots, \mathsf{SE.Enc}(K_s, N_s, A_s, M_s)$ are all the same. CMT-4 is likewise extended to CMT$_s$-4. Clearly CMT-n implies CMT$_s$-n ($n \in \{1, 4\}$). Our results however consider CMT$_s$-n (not just CMT-n) and prove bounds on its being violated that degrade quickly with s. This allows us to give better guarantees for security against partitioning oracle attacks [34]. Namely, we can show that, with use of one of our CMT-1 schemes, the probability that an attacker can speed up the attack by a factor s decreases quickly as a function of s.

Related work. We start by noting a few "firsts." (1) Prior nonce-based committing schemes were only for UNAE. We are giving the first ones for MRAE. (2) We give the first schemes that commit to all encryption inputs, meaning achieve CMT-4. (3) We give the first schemes (our four CAU schemes) that have zero ciphertext overhead (4) We give analyses of multi-input committing security with bounds that degrade quickly in the number s of inputs.

FOR [25] take a broad, systematic approach, giving general methods to build key-committing primitives. Their key-committing encryption schemes however are randomized rather than nonce-based. Also, they don't show multi-user security with good bounds.

Many of the schemes of GLR [27] are randomized. Their leading nonce-based scheme, Committing Encrypt-and-PRF (CEP), has a block of ciphertext overhead, unlike our CAU schemes. CEP also seems to fare somewhat more poorly than our schemes with regard to performance and extent of software change. They don't show good multi-user security.

The DGRW scheme [22] is randomized, not nonce-based. It uses a compression function that is assumed collision resistant and RKA-PRF-secure [9]. Instantiating the latter via Davies-Meyers yields a blockcipher-based scheme, but speed with AES-NI is reduced because the blockcipher key changes with each message block. They incur ciphertext overhead, and don't show good multi-user security.

It should be noted that GLR [27] and DGRW [22] are targeting and achieving properties beyond key-commiting security, as needed for message franking. In particular, their schemes, unlike ours, produce a *compact* commitment to the message.

ADGKLS [4] consider nonce-based schemes and give a generic way to add key-committing security to a UNAE scheme. Their transform uses a pair of collision-resistant PRFs. UtC generalizes this, using instead our (new) committing PRF abstraction; instantiation with CX yields efficiency improvements over ADGKLS.

They also give a padding-based key-committing extension of GCM, but, unlike our CAU-C1, it increases ciphertext size.

LGR [34] say "our results suggest that future work should design, standardize, and add to libraries, AE schemes designed to be key-committing." Our schemes are intended as a response.

2 Preliminaries

NOTATION AND TERMINOLOGY. Let ε denote the empty string. For a string x we write $|x|$ to refer to its bit length, and $x[i:j]$ is the bits i through j (inclusive) of x, for $1 \le i \le j \le |x|$. By $\mathsf{Func}(\mathsf{Dom}, \mathsf{Rng})$ we denote the set of all functions $f : \mathsf{Dom} \to \mathsf{Rng}$ and by $\mathsf{Perm}(\mathsf{Dom})$ the set of all permutations $\pi : \mathsf{Dom} \to \mathsf{Dom}$. We use $\perp$ as a special symbol to denote rejection, and it is assumed to be outside $\{0,1\}^*$. In the context that we use a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, the *block length* of a string x, denoted as $|x|_n$, is $\max\{1, \lceil |x|/n \rceil\}$. If X is a finite set, we let $x \leftarrow\!\!\$\, X$ denote picking an element of X uniformly at random and assigning it to x.

SYMMETRIC ENCRYPTION. A (nonce-based) symmetric encryption (SE) scheme SE specifies deterministic algorithms $\mathsf{SE.Enc} : \mathcal{K} \times \mathcal{N} \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ and $\mathsf{SE.Dec} : \mathcal{K} \times \mathcal{N} \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \cup \{\perp\}$. Here $\mathcal{K}, \mathcal{N}$ are the associated key and nonce spaces. The encryption algorithm takes as input a key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$, associated data $A \in \{0,1\}^*$ and a message $M \in \mathcal{M}$, and returns a ciphertext $C \leftarrow \mathsf{SE.Enc}(K, N, A, M)$. The decryption algorithm takes as input K, N, A, C and returns either a message $M \in \{0,1\}^*$ or the special symbol $\perp$ indicating invalidity or rejection. The correctness requirement says that decryption reverses encryption, namely if $C \leftarrow \mathsf{SE.Enc}(K, N, A, M)$ then $\mathsf{SE.Dec}(K, N, A, C)$ returns M. We assume that there is a ciphertext-length function $\mathsf{SE.len} : \mathbb{N} \to \mathbb{N}$ such that the length of $\mathsf{SE.Enc}(K, N, A, M)$ is exactly $\mathsf{SE.len}(|M|)$ bits for all K, N, A, M.

We say that SE is *tidy* [38] if $M \leftarrow \mathsf{SE.Dec}(K, N, A, C)$ implies that $\mathsf{SE.Enc}(K, N, A, M)$ returns C. Combining correctness and tidiness means that functions $\mathsf{SE.Enc}(K, N, A, \cdot)$ and $\mathsf{SE.Dec}(K, N, A, \cdot)$ are the inverse of each other. The schemes we consider will be tidy.

AE SECURITY. Let SE be a symmetric encryption scheme with key space $\mathcal{K}$ and nonce space $\mathcal{N}$. We now define its security as an authenticated encryption (AE) scheme in the multi-user setting, following the formalization of [13]. The first, basic requirement, called unique-nonce AE (UNAE), asks for security assuming encryption never repeats a nonce for any given user. The second, advanced requirement, called misuse-resistant AE (MRAE) drops this condition. Consider games $\mathbf{G}^{\mathrm{real}}_{\mathsf{SE}}(\mathcal{A})$ and $\mathbf{G}^{\mathrm{rand}}_{\mathsf{SE}}(\mathcal{A})$ in Fig. 2. We define the mrae advantage of an adversary $\mathcal{A}$ as

$$\mathbf{Adv}^{\mathrm{mrae}}_{\mathsf{SE}}(A) = \Pr[\mathbf{G}^{\mathrm{real}}_{\mathsf{SE}}(\mathcal{A})] - \Pr[\mathbf{G}^{\mathrm{rand}}_{\mathsf{SE}}(\mathcal{A})] \ .$$

To avoid trivial wins, we forbid the adversary from repeating a query to either its ENC or its VF oracles. Moreover, if the adversary previously received $C \leftarrow$

Game $\mathbf{G}^{\text{real}}_{\mathsf{SE}}(\mathcal{A})$	Game $\mathbf{G}^{\text{rand}}_{\mathsf{SE}}(\mathcal{A})$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Enc},\text{Vf}}$; return b'	$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Enc},\text{Vf}}$; return b'
New()	New()
$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \mathcal{K}$	$v \leftarrow v + 1$
Enc(i, N, A, M)	Enc(i, N, A, M)
If $i \notin \{1, \ldots, v\}$ return $\bot$	If $i \notin \{1, \ldots, v\}$ return $\bot$
$C \leftarrow \mathsf{SE.Enc}(K_i, N, A, M)$	$C \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.len}(\lvert M\rvert)}$
Return C	Return C
Vf(i, N, A, C)	Vf(i, N, A, C)
If $i \notin \{1, \ldots, v\}$ return $\bot$	If $i \notin \{1, \ldots, v\}$ return $\bot$
$V \leftarrow \mathsf{SE.Dec}(K_i, N, A, C)$; return $(V \neq \bot)$	return false

Fig. 2. Games defining misuse-resistance security of a SE scheme SE.

Enc(i, N, A, M) then later it is not allowed to query Vf(i, N, A, C). We can now recover UNAE security by restricting attention to *unique-nonce* adversaries, these being ones that never repeat an (i, N) pair across their Enc queries. (That is, a nonce is never reused for a given user.) We stress that there is no such restriction on decryption queries. If $\mathcal{A}$ is a unique-nonce adversary, then we write its advantage as $\mathbf{Adv}^{\text{unae}}_{\mathsf{SE}}(\mathcal{A})$ for clarity.

Multi-collision resistance. Let $H : \mathsf{Dom} \to \mathsf{Rng}$ be a function. Let $s \geq 2$ be an integer. An s-way collision for H is a tuple $(X_1, \ldots, X_s)$ of distinct points in Dom such that $H(X_1) = \cdots = H(X_s)$. For an adversary $\mathcal{A}$, define its advantage in breaking the s-way multi-collision resistance of H as

$$\mathbf{Adv}^{\text{coll}}_{H,s}(\mathcal{A}) = \Pr[(X_1, \ldots, X_s) \text{ is an } s\text{-way collision for } H]$$

where the probability is over $(X_1, \ldots, X_s) \leftarrow\!\!\$\ \mathcal{A}$. When $s = 2$ we recover the classical notion of collision resistance.

AXU hashing. Let $G : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ be a keyed hash function. We say that G is *c-almost xor universal* if for all $(M, A) \neq (M', A')$ and all $\Delta \in \{0,1\}^n$,

$$\Pr_{K \leftarrow\!\!\$\ \{0,1\}^n}[G_K(M, A) \oplus G_K(M', A') = \Delta] \leq \frac{c \cdot \max\{|M|_n + |A|_n, |M'|_n + |A'|_n\}}{2^n} \ .$$

PRFs and PRPs. For a function $\mathsf{F} : \{0,1\}^k \times \mathsf{Dom} \to \mathsf{Rng}$ and an adversary $\mathcal{A}$, we define the advantage of $\mathcal{A}$ in breaking the (multi-user) PRF security of F [6] as

$$\mathbf{Adv}^{\text{prf}}_{\mathsf{F}}(\mathcal{A}) = 2\Pr[\mathbf{G}^{\mathsf{prf}}_{\mathsf{F}}(\mathcal{A})] - 1 \ ,$$

where game $\mathbf{G}^{\mathsf{prf}}_{\mathsf{F}}(\mathcal{A})$ is shown in Fig. 3. For a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ and an adversary $\mathcal{A}$, we define the advantage of $\mathcal{A}$ in breaking the multi-

Game $\mathbf{G}^{\mathsf{prf}}_{\mathsf{F}}(\mathcal{A})$	NEW()	EVAL(i, M)
$v \leftarrow 0;\ b \leftarrow\$ \{0,1\}$	$v \leftarrow v + 1$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$b' \leftarrow\$ \mathcal{A}^{\text{NEW},\text{EVAL}}$	$K_v \leftarrow\$ \{0,1\}^k$	$C_1 \leftarrow \mathsf{F}(K_i, M);\ C_0 \leftarrow f_i(M)$
return $(b' = b)$	$f_v \leftarrow\$ \mathsf{Func}(\mathsf{Dom}, \mathsf{Rng})$	return C_b

Game $\mathbf{G}^{\mathsf{prp}}_{E}(\mathcal{A})$	NEW()	EVAL(i, M)
$v \leftarrow 0;\ b \leftarrow\$ \{0,1\}$	$v \leftarrow v + 1$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$b' \leftarrow \mathcal{A}^{\text{NEW},\text{EVAL}}$	$K_v \leftarrow\$ \{0,1\}^k$	$C_1 \leftarrow E(K_i, M);\ C_0 \leftarrow \pi_i(M)$
return $(b' = b)$	$\pi_v \leftarrow\$ \mathsf{Perm}(\{0,1\}^n)$	return C_b

Fig. 3. Games defining PRF security of F and PRP security of E.

user PRP security of E as

$$\mathbf{Adv}^{\text{prp}}_{E}(\mathcal{A}) = 2\Pr[\mathbf{G}^{\mathsf{prp}}_{E}(\mathcal{A})] - 1\ ,$$

where game $\mathbf{Adv}^{\text{prp}}_{\mathsf{F}}(\mathcal{A})$ is defined in Fig. 3. Mouha and Luykx [37] show that if we model E as an ideal cipher then for any adversary making q evaluation queries and p ideal-cipher queries, $\mathbf{Adv}^{\text{prp}}_{E}(\mathcal{A}) \leq (q^2 + 2pq)/2^{k+1}$.

3 Committing AE Framework

Let SE be a symmetric encryption scheme with key space $\mathcal{K}$ and nonce space $\mathcal{N}$. We define a hierarchy of levels of committing security CMTD-1 $\leftarrow$ CMTD-3 $\leftrightarrow$ CMTD-4, where the "D" indicates these are decryption-based. For each $\ell \in \{1, 3, 4\}$ we also recast CMTD-ℓ as an encryption-based notion CMT-ℓ that is simpler but equivalent if SE is tidy. We give relations between the notions, and then extend all this to s-way committing security for $s \geq 2$.

Think of ℓ here as indicating that we commit to the first ℓ inputs of the encryption algorithm. Since popular schemes, and the ones in this paper in particular, are tidy, the CMT-ℓ notions become our focus moving forward. The Introduction had discussed only CMT-1 and CMT-4; here we introduce the $\ell = 3$ notions as simpler than, but equivalent to, the $\ell = 4$ ones, something our results will exploit.

This section concludes with a simple transform, called EtH, that promotes $\ell = 1$ security to $\ell = 4$ security with minimal overhead.

WHAT IS COMMITTED? In asking that a ciphertext $C \leftarrow \mathsf{SE.Enc}(K, N, A, M)$ be a committal, the question is, to what? We consider this in a fine-grained way. We define a function WiC_ℓ (What is Committed) that on input (K, N, A, M) returns the part of the input to which we want the ciphertext to be a commitment. It is defined as shown in the table in Figure 4. Thus, when $\ell = 1$, we are asking that we commit to the key; this corresponds to robustness [3], also called key-robustness [25] or key-committing [4] security. When $\ell = 3$, we commit to the key, nonce and associated data. Finally $\ell = 4$ means we commit, additionally, to the message, and thus to all the inputs of $\mathsf{SE.Enc}$.

Game $\mathbf{G}^{\text{cmtd-}\ell}_{\mathsf{SE}}(\mathcal{A})$
$\big(C, (K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2)\big) \leftarrow\!\!\$\ \mathcal{A}$
Require: $\mathsf{WiC}_\ell(K_1, N_1, A_1, M_1) \neq \mathsf{WiC}_\ell(K_2, N_2, A_2, M_2)$
Return $((M_1 = \mathsf{SE.Dec}(K_1, N_1, A_1, C)$ and $M_2 = \mathsf{SE.Dec}(K_2, N_2, A_2, C))$

Game $\mathbf{G}^{\text{cmt-}\ell}_{\mathsf{SE}}(\mathcal{A})$
$\big((K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2)\big) \leftarrow\!\!\$\ \mathcal{A}$
Require: $\mathsf{WiC}_\ell(K_1, N_1, A_1, M_1) \neq \mathsf{WiC}_\ell(K_2, N_2, A_2, M_2)$
Return $(\mathsf{SE.Enc}(K_1, N_1, A_1, M_1) = \mathsf{SE.Enc}(K_2, N_2, A_2, M_2))$

ℓ	1	3	4
$\mathsf{WiC}_\ell(K, N, A, M)$	K	(K, N, A)	(K, N, A, M)

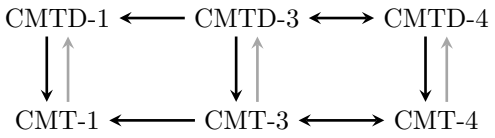

Fig. 4. Games defining committing security of a symmetric encryption scheme SE. Below them are the associated what-is-committed functions WiC_ℓ, and then the relations between the notions. The **gray** arrows hold for tidy SE.

THE D-NOTIONS. Let $\ell \in \{1, 3, 4\}$ be an integer representing the level of committing security. Consider game $\mathbf{G}^{\text{cmtd-}\ell}_{\mathsf{SE}}(\mathcal{A})$ in Fig. 4, and define the advantage of adversary $\mathcal{A}$ as $\mathbf{Adv}^{\text{cmtd-}\ell}_{\mathsf{SE}}(\mathcal{A}) = \Pr[\mathbf{G}^{\text{cmtd-}\ell}_{\mathsf{SE}}(\mathcal{A})]$. In the game, the adversary provides a ciphertext C together with a pair of tuples (K_1, N_1, A_1, M_1) and (K_2, N_2, A_2, M_2). (No entry of a tuple is allowed to be $\perp$.) The adversary wins if both decryptions of C equal the respective adversary-provided messages. The game requires that the outputs of the WiC_ℓ function on the adversary-provided tuples be different, precluding a trivial win. The only difference between the different levels indicated by ℓ is in the value of $\mathsf{WiC}_\ell(K, N, M, A)$ as given in the table. We denote the resulting notions by CMTD-ℓ for $\ell \in \{1, 3, 4\}$.

Our CMTD-1 notion is stronger than the key-committing notion in prior work [4], since we allow the adversary to specify *different* nonces N_1 and N_2. In contrast, the key-committing nonce requires the two nonces to be the same.

On the other hand, achieving CMTD-4 security requires processing the associated data under a collision-resistant hash function. To see why, note that in settings where messages are the empty string, a ciphertext is a *compact* commitment of the associated data.

THE E-NOTIONS. Let $\ell \in \{1, 3, 4\}$ be an integer representing the level of committing security. Consider game $\mathbf{G}^{\text{cmt-}\ell}_{\mathsf{SE}}(\mathcal{A})$ in Fig. 4, and define the advantage of adversary $\mathcal{A}$ as $\mathbf{Adv}^{\text{cmt-}\ell}_{\mathsf{SE}}(\mathcal{A}) = \Pr[\mathbf{G}^{\text{cmt-}\ell}_{\mathsf{SE}}(\mathcal{A})]$. In the game, the adversary

Game $\mathbf{G}^{\text{cmtd-}\ell}_{\mathsf{SE},s}(\mathcal{A})$
$(C, (K_1, N_1, A_1, M_1), \ldots, (K_s, N_s, A_s, M_s)) \leftarrow\!\!\$\ \mathcal{A}$
Require: $\mathsf{WiC}_\ell(K_1, N_1, A_1, M_1), \ldots, \mathsf{WiC}_\ell(K_s, N_s, A_s, M_s)$ are all distinct
Return $(\forall i : M_i = \mathsf{SE.Dec}(K_i, N_i, A_i, C_i))$

Game $\mathbf{G}^{\text{cmt-}\ell}_{\mathsf{SE},s}(\mathcal{A})$
$((K_1, N_1, A_1, M_1), \ldots, (K_s, N_s, A_s, M_s)) \leftarrow\!\!\$\ \mathcal{A}$
Require: $\mathsf{WiC}_\ell(K_1, N_1, A_1, M_1), \ldots, \mathsf{WiC}_\ell(K_s, N_s, A_s, M_s)$ are all distinct
Return $(\mathsf{SE.Enc}(K_1, N_1, A_1, M_1) = \cdots = \mathsf{SE.Enc}(K_s, N_s, A_s, M_s))$

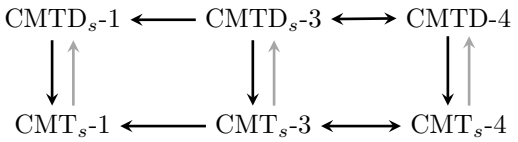

Fig. 5. Games defining s-way committing security of a symmetric encryption scheme SE for $s \geq 2$. Below them are the relations between the notions. The **gray** arrows hold for tidy SE.

provides a pair of tuples (K_1, N_1, A_1, M_1) and (K_2, N_2, A_2, M_2). (No entry of a tuple is allowed to be $\perp$.) The functions WiC_ℓ are unchanged. The game returns true (the adversary wins) if the encryptions of the two tuples are the same. We denote the resulting notions by CMT-ℓ for $\ell \in \{1, 3, 4\}$.

Relations. The bottom of Fig. 4 shows the relations between the notions of committing security. An arrow A $\rightarrow$ B, read as A implies B, means that any scheme SE that is A-secure is also B-secure. A gray arrow means the implication holds when SE is tidy. The relations in the picture are justified in [8].

Multi-input committing security. The notions above considered an adversary successful if it opened a ciphertext in two different ways (D) or provided two encryption inputs with the same output (E). We now generalize from "two" to an integer parameter $s \geq 2$, the prior notions being the special case $s = 2$. The games, in Figure 5, are parameterized, as before, with symmetric encryption scheme SE, but now also with s. Again there are "D" and "E" variants. The functions WiC_ℓ remain as in Figure 4. The advantages of an adversary $\mathcal{A}$ are defined as $\mathbf{Adv}^{\text{cmtx-}\ell}_{\mathsf{SE},s}(\mathcal{A}) = \Pr[\mathbf{G}^{\text{cmtx-}\ell}_{\mathsf{SE},s}(\mathcal{A})]$ for x $\in \{\text{d}, \varepsilon\}$ and $\ell \in \{1, 3, 4\}$. We denote the resulting notions by CMTX$_s$-ℓ for X $\in \{\text{D}, \varepsilon\}$ and $\ell \in \{1, 3, 4\}$. Their relations remain as before and for completeness are also illustrated in Figure 5.

Why generalize? It is easy to see that CMTX-ℓ implies CMTX$_s$-ℓ for all $s \geq 2$ and X $\in \{\text{D}, \varepsilon\}$, meaning if a scheme SE is CMTX-ℓ-secure then it is also CMTX$_s$-ℓ for all $s \geq 2$. So why consider $s > 2$? The reason is that we can give schemes for which the bound on adversary advantage gets better as s gets larger, indeed even decaying exponentially with s. Indeed, one can break

$\underline{\overline{\mathsf{SE}}.\mathsf{Enc}(K, N, A, M)}$	$\underline{\overline{\mathsf{SE}}.\mathsf{Dec}(K, N, A, C^* \| T')}$
$L \leftarrow H(K, (N, A))$	$L \leftarrow H(K, (N, A))$
$C \leftarrow \mathsf{SE}.\mathsf{Enc}(L, N, \varepsilon, M)$	$M \leftarrow \mathsf{SE}.\mathsf{Dec}(L, N, \varepsilon, C)$
Return C	Return M

Fig. 6. The scheme $\overline{\mathsf{SE}} = \mathsf{HtE}[\mathsf{SE}, H]$ defined via the Hash-then-Encrypt transform applied to a symmetric encryption scheme SE and a function H.

CMT-1-security of the scheme CAU-C1 in Section 5 in about 2^{64} operations. However, for any adversary $\mathcal{A}$ that spends at most 2^{80} operations, the chance that it can break CMT_3-1 security of CAU-C1 is at most 2^{-62}. This allows us to offer a much stronger guarantee for situations like the Partitioning Oracle attack [34]. Recall that here, breaking CMT_s-1 security speeds up the time to find the underlying password used for key derivation by a factor of s. Thus our results say that despite investing 2^{80} operations, $\mathcal{A}$ can at best speed up its password search by a factor of two.

Discussion. Practical schemes tend to be tidy, and all the ones we consider are, so, moving forward, we make tidiness an implicit assumption and focus on the E notions. Our primary focus is (s-way) CMT-1 because this is already non-trivial, what was targeted in many previous works, and enough for many applications. Below we give a generic way to promote CMT-1 security to CMT-4 security.

From CMT-1 to CMT-4. We give a way to turn CMT-1 security into CMT-4 security, for both unique-nonce and misuse-resistance security. (That is, if you can commit to the key, it is easy to commit to everything.) It takes the form of a transform we call HtE (Hash then Encrypt). The ingredients are a base symmetric encryption scheme SE with key space $\{0,1\}^k$, and a function $H : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^k$. The encryption and decryption algorithms of the scheme $\overline{\mathsf{SE}} = \mathsf{HtE}[\mathsf{SE}, H]$ are shown in Fig. 6. The key-space and nonce-space remain that of SE.

With regard to performance, HtE preserves ciphertext length, meaning we are promoting CMT-1 to CMT-4 without increase in ciphertext size. The computational overhead, which is the computation of $H(K, (N, A))$, is optimal, since achieving CMT-4 requires processing the associated data with a collision-resistant hash function. In practice, associated data is often short (for example, IP headers are at most 60B), and thus HtE typically incurs just a constant computational overhead over the base scheme SE.

With regard to security, intuitively, if H is collision-resistant then the subkey L is a commitment to the master key K, the nonce N and the associated data A. As a result, if the ciphertext is a commitment to the subkey L then it is also a commitment to (K, N, A). Hence the CMT-1 security of SE implies the CMT-3 security of $\overline{\mathsf{SE}}$, and thus, as per the relations in Figure 4, also its CMT-4 security. Furthermore we will show that HtE preserves both unique-nonce and misuse-resistance security assuming H is a PRF.

We note that we do not assume H is a random oracle, instead making the standard-model assumption that it is a collision-resistant PRF.We now give formal results confirming the intuition above. The following shows that HtE indeed promotes CMT-1 security to CMT-4 security. The proof is in [8].

Theorem 1. *Let SE be an SE scheme with key length k, and let $H : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^k$ be a hash function. Let $\overline{\mathsf{SE}} = \mathsf{HtE}[\mathsf{SE}, H]$. Fix an integer $s \geq 2$ and let $t = \lceil \sqrt{s} \rceil$. Then given an adversary $\mathcal{A}$, we can construct adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$ such that*

$$\mathbf{Adv}^{\text{cmt-4}}_{\overline{\mathsf{SE}},s}(\mathcal{A}) \leq \max\left\{\mathbf{Adv}^{\text{coll}}_{H,t}(\mathcal{B}_0), \mathbf{Adv}^{\text{cmt-1}}_{\overline{\mathsf{SE}},t}(\mathcal{B}_1)\right\} \ .$$

Each $\mathcal{B}_i$ runs $\mathcal{A}$ and then runs H on s pairs (nonce, associated data) of $\mathcal{A}$.

The next result shows that HtE preserves both unique-nonce and misuse-resistance security, provided that H is a good PRF. The proof is in [8].

Theorem 2. *Let SE be an SE scheme with key length k, and let $H : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^k$ be a hash function. Let $\overline{\mathsf{SE}} = \mathsf{HtE}[\mathsf{SE}, H]$. Then given an adversary $\mathcal{A}$ that makes at most q queries of totally σ_a bits for (nonce, AD) pairs and at most B queries per (user, nonce, AD) triples, we can construct adversaries $\mathcal{B}$ and $\mathcal{D}$ such that*

$$\mathbf{Adv}^{\text{mrae}}_{\overline{\mathsf{SE}}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{prf}}_{H}(\mathcal{B}) + \mathbf{Adv}^{\text{mrae}}_{\mathsf{SE}}(\mathcal{D}) \ .$$

If $\mathcal{A}$ is unique-nonce then so is $\mathcal{D}$, and we can rewrite the bound as

$$\mathbf{Adv}^{\text{unae}}_{\overline{\mathsf{SE}}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{prf}}_{H}(\mathcal{B}) + \mathbf{Adv}^{\text{unae}}_{\mathsf{SE}}(\mathcal{D}) \ .$$

Adversary $\mathcal{B}$ makes at most q queries on at most σ_a bits. Its running time is about that of $\mathcal{A}$ plus the time to encrypt/decrypt $\mathcal{A}$'s queries. Adversary $\mathcal{D}$ makes q queries of the total length as $\mathcal{A}$, but it makes only B queries per user. Its running time is about that of $\mathcal{A}$ plus $O(\sigma_a \log(B))$.

We now discuss the choice of H. If nonce length is fixed, one can instantiate $H(K,(N,A))$ via $\mathsf{HMAC\text{-}SHA256}(K\|N\|A)[1:k]$ or $\mathsf{SHA3}(K\|N\|A)[1:k]$. We stress that if one considers using $\mathsf{SHA256}(K\|N\|A)[1:k]$, one must beware of the extension attack, to avoid which one should only use this if $k = 128$ [21].

4 Some Building Blocks

We give building blocks, technical results and information that we will use later. Some of the results are interesting in their own right, and may have applications beyond the context of committing AE.

Multi-user PRP/PRF Switching. Lemma 1 below generalizes the classical PRP/PRF Switching Lemma [12] to the multi-user setting; see [8] for a proof. If one uses a hybrid argument on the standard single-user PRP/PRF Switching Lemma, one will obtain a weak bound $uB^2/2^n$, where u is the number of users. If there are $\Theta(q)$ users and some user makes $\Theta(q)$ queries then this bound is in the order of $q^3/2^n$, whereas our bound is just $q^2/2^n$ in this case.

Alternatively, if one parameterizes on q only, as in [35], one will end up with another weak bound $q^2/2^n$. In the setting where each user makes approximately B queries, this bound is even weaker than the trivial bound $uB^2/2^n$. Lemma 1 instead uses a different parameterization to obtain a sharp bound $qB/2^n$. The idea of using both B and q as parameters in multi-user analysis is first introduced in [20].

Lemma 1 (Multi-user PRP/PRF Switching Lemma). *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. For any adversary $\mathcal{A}$, if it makes at most q evaluation queries in total, with at most B queries per user, then*

$$\mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{A}) + \frac{Bq}{2^n} \enspace .$$

Simplifying UNAE/MRAE proofs. In UNAE/MRAE proofs, an adversary can adaptively interleave encryption and verification queries. Proofs will be simpler if the adversary is *orderly*, meaning that (i) its verification queries are made at the very end, and (ii) each verification query does not depend on the answers of prior verification queries, but may still depend on the answers of prior encryption queries. Proposition 1 shows that one can consider only orderly adversaries in UNAE/MRAE notions with just a small loss in the advantage; see [8] for a proof. The idea of restricting to orderly adversaries has been used in prior works [20, 11]. They show that one can factor an UNAE/MRAE adversary $\mathcal{A}$ into two adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$ attacking privacy and authenticity respectively, where $\mathcal{B}_1$ is orderly. Here we instead transform $\mathcal{A}$ to another orderly UNAE/MRAE adversary $\mathcal{B}$.

Proposition 1. *Let SE be a symmetric encryption scheme such that its ciphertext is at least τ-bit longer than the corresponding plaintext. For any adversary $\mathcal{A}$ that makes q_v verification queries, we can construct another orderly adversary $\mathcal{B}$ of about the same running time such that*

$$\mathbf{Adv}^{\mathrm{mrae}}_{\mathsf{SE}}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{mrae}}_{\mathsf{SE}}(\mathcal{B}) + \frac{2q_v}{2^\tau} \enspace .$$

Adversary $\mathcal{B}$ has the same query statistics as $\mathcal{A}$. Moreover, if $\mathcal{A}$ is unique-nonce then so its $\mathcal{B}$, and thus in that case we can rewrite the bound as

$$\mathbf{Adv}^{\mathrm{unae}}_{\mathsf{SE}}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{unae}}_{\mathsf{SE}}(\mathcal{B}) + \frac{2q_v}{2^\tau} \enspace .$$

For both notions, if every ciphertext of SE is exactly τ-bit longer than its plaintext then the term $2q_v/2^\tau$ can be improved to $q_v/2^\tau$.

Committing AE via collision-resistant hash. Intuitively, from the definition of committing AE, to achieve this goal, one needs to include the image of the key under some (multi)collision-resistant hash function in the ciphertext. This connection has been recognized and explored in prior works. For example, (i) the OPAQUE protocol [30] recommends the use of the Encrypt-then-HMAC construction, (ii) Albertini et al. [4] suggest using a hash-based key-derivation

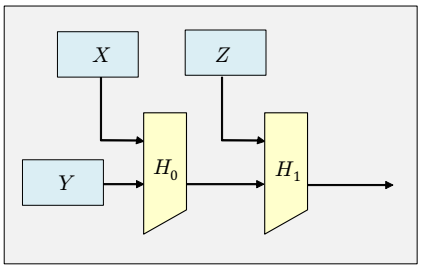

Fig. 7. Illustration of the cascade of the two hash functions H_0 and H_1.

function to add key-committing security into legacy AE schemes; and (iii) Dodis et al. [22] propose a hash-based AE design for Facebook's message franking. The definition was recalled in Section 2. We now give some new fundamental results.

The Truncated Davies-Meyer construction. A common way to build a collision-resistant compression function from a blockcipher is the Davies-Meyer construction. Our paper makes extensive use of this construction to have a cheap commitment of the key for obtaining committing security. It appears in both the AE schemes of Sections 5 and 6. While the collision resistance of the Davies-Meyer construction is well-known [18], its multi-collision resistance has not been studied before. Moreover, in our use of Davies-Meyer, we usually have to truncate the output, and even ordinary collision resistance of truncated Davies-Meyer has not been investigated.

In particular, let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Let $m \leq n$ be an integer, and define $\mathsf{DM}[E,m] : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^m$ via

$$\mathsf{DM}[E,m](X,Y) = \big(E_X(Y) \oplus Y\big)[1:m] \ .$$

We write $\mathsf{DM}[E]$ for the special case $m = n$ (meaning there is no truncation). Proposition 2 below analyzes the multi-collision resistance of $\mathsf{DM}[E,m]$; see [8] for a proof. The result is in the ideal-cipher model, that is, the adversary is given oracle access to both E and its inverse, and the number of ideal-cipher queries refers to the total queries to these two oracles.

Proposition 2. *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Let $s \geq 2$ and $m \leq n$ be integers. For an adversary $\mathcal{A}$ that makes at most $p \leq 2^{n-1} - s$ ideal-cipher queries,*

$$\mathbf{Adv}^{\mathrm{coll}}_{\mathsf{DM}[E,m],s}(\mathcal{A}) \leq 2^{1-m} + \binom{p}{s} \cdot 2^{(1-m)(s-1)} \ .$$

For the case $s = 2$ and $m = n$, our bound is $2^{1-n} + p(p-1)/2^n$, which slightly improves the classical bound $p(p+1)/2^n$ of Black, Rogaway, and Shrimpton [18]. For a general s, in [8], we show that for an ideal hash function on range $\{0,1\}^m$,

there is an attack on the s-way multi-collision resistance of advantage

$$\frac{1}{4} \cdot \binom{p}{s} \cdot 2^{-m(s-1)} \ .$$

Thus the Truncated Davies-Meyer construction achieves essentially the best possible multi-collision resistance that we can hope for the output length m.

The Iterative Truncated-Permutation construction. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Let $r < n$ be a positive integer, and let $m \leq 2n$ be a positive even integer. Let $\mathsf{pad} : \{0,1\}^r \times \{1,2\} \to \{0,1\}^n$ be a one-to-one mapping. Define $\mathsf{ITP}[E,r,m] : \{0,1\}^k \times \{0,1\}^r \to \{0,1\}^{2m}$ via

$$\mathsf{ITP}[E,r,m](K,X) = E_K(\mathsf{pad}(X,1))[1:m/2] \| E_K(\mathsf{pad}(X,2))[1:m/2] \ .$$

The ITP construction is used in the key-derivation function of AES-GCM-SIV, where $r = 96$ and $m = n = 128$, and $\mathsf{pad}(X,i)$ is the concatenation of X and an $(n-r)$-bit encoding of i. For proving the committing security of the variants of AES-GCM-SIV in Section 6, we need to show that in using ITP to derive subkeys, one is also committing the master key and the nonce to one of the subkeys. Proposition 3 below analyzes the multi-collision resistance of ITP; see [8] for a proof. The analysis is difficult because ITP was *not* designed for collision resistance. This result is in the ideal-cipher model, meaning that the adversary is given oracle access to both E and E^{-1}, and the number of ideal-cipher queries refers to the total queries to both oracles. Note that for $r \leq 3n/4$ and $m = n$ (which holds for the situation of AES-GCM-SIV), ITP has birthday-bound security or better.

Proposition 3. *Let m, r, n be positive integers such that $r < n$, and $m \leq 2n$ is even. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Let $s \geq 2$ be an integer. For an adversary $\mathcal{A}$ that makes at most $p \leq 2^{n-3} - s$ ideal-cipher queries,*

$$\mathbf{Adv}^{\mathrm{coll}}_{\mathsf{ITP}[E,r,m],s}(\mathcal{A}) \leq 2^{1-m} + \frac{4p^s}{s! \cdot 2^{(m-2)(s-1)}} + \frac{2^{m/2+1} \cdot p^s}{s! \cdot 2^{(m/2+n-r-2)s}} \ .$$

Compared to the lower bound $\binom{p}{s} \cdot 2^{-m(s-1)}$, the ITP construction has some security degradation due to the last term in the bound of Proposition 3. In [8], we give an attack that matches this term, implying that the bound of Proposition 3 is tight.

Multi-collision resistance on a cascade. Let $c \geq 2$ be an integer. For each $i \in \{0, \ldots, c-1\}$, let $H_i : \mathcal{L}_i \times \mathcal{R}_i \to \mathsf{Rng}_i$ be a hash function such that $\mathsf{Rng}_i \subseteq \mathcal{R}_{i+1}$. Define the *cascade* $H_0 \circ H_1$ of H_0 and H_1 as the hash function H such that $H(X,Y,Z) = H_1\big(Z, H_0(X,Y)\big)$; see Fig. 7 for an illustration. The cascade $H_0 \circ \cdots \circ H_i$ of $H_0, \ldots, H_i$ is defined recursively as $(H_0 \circ \cdots \circ H_{i-1}) \circ H_i$. Cascading appears in AE schemes of Section 6 where one first commits the master key into a subkey, and then includes a commitment of the subkey into the ciphertext. The following result shows how to bound the multi-collision resistance of $H_0 \circ \cdots \circ H_{c-1}$; see [8] for a proof.

Proposition 4. *Let H be the cascade of hash functions $H_0, H_1, \ldots, H_{c-1}$ as above. Let $s \geq 2$ be an integer, and let $t = \lceil \sqrt[c]{s} \rceil$. Then for any adversary $\mathcal{A}$, we can construct adversaries $\mathcal{B}_0, \ldots, \mathcal{B}_{c-1}$ such that*

$$\mathbf{Adv}^{\mathrm{coll}}_{H,s}(\mathcal{A}) \leq \max\left\{\mathbf{Adv}^{\mathrm{coll}}_{H_0,t}(\mathcal{B}_0), \ldots, \mathbf{Adv}^{\mathrm{coll}}_{H_{c-1},t}(\mathcal{B}_{c-1})\right\} \ .$$

Each adversary B_i runs $\mathcal{A}$, and then runs the cascade of $H_0, \ldots, H_{\min\{c-2,i\}}$ on the s inputs of $\mathcal{A}$.

5 A Committing Variant of GCM

In this section, we describe a close variant CAU-SIV-C1 of AES-GCM-SIV that achieves both CMT-1 and unique-nonce security with the same speed and bandwidth costs as GCM. In this entire section, let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Following Bellare and Tackmann [14], we consider a generalization CAU of GCM. This scheme loosely follows the encrypt-then-MAC paradigm, where the encryption scheme is the CTR mode, and the MAC is the Carter-Wegman construction via an almost-xor-universal (AXU) hash function. (The name CAU is a mnemonic for the use of the CTR mode and an AXU hash function.) In GCM, the function G is instantiated by a 1.5-AXU hash GHASH.

The scheme CAU. We now describe the scheme CAU. Let $G : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ be an AXU hash function. Let $\mathcal{N} = \{0,1\}^r$ be the nonce space, where $r < n$ is an integer. In GCM, $n = 128$ and $r = 96$. For a string $N \in \mathcal{N}$, we write $\mathsf{pad}(N)$ to refer to $N \| 0^{n-r-1} \| 1$. Let $\tau \leq n$ be the tag length. The scheme $\mathsf{CAU}[E, G, \tau]$ is specified in Fig. 8; it only accepts messages of at most $2^{n-r} - 2$ blocks. See also Fig. 9 for an illustration.

Specification of CAU-C1. The code of $\mathsf{CAU\text{-}C1}[E, G, \tau]$ is shown in Fig. 8. Like CAU, it only accepts messages of at most $2^{n-r} - 2$ blocks. Compared to CAU, the change occurs in how we derive the tag, as illustrated in Fig. 9. In particular, in CAU, one obtains the tag by using the Carter-Wegman paradigm, applying a one-time pad $E_K(\mathsf{pad}(N))$ to the output R of the AXU hash. In contrast, in CAU-C1, we use a different Carter-Wegman flavor, enciphering $V \leftarrow R \oplus \mathsf{pad}(N)$. However, to ensure committing security, instead of using $T \leftarrow E_K(V)[1:\tau]$, we employ the Truncated Davies-Meyer method, outputting $T \leftarrow \mathsf{DM}[E, \tau](K, V)$.

We note that if one instead computes $T \leftarrow \mathsf{DM}[E, \tau](K, R)$ then the resulting scheme will not have unique-nonce security. In particular, once we obtain a valid ciphertext C under nonce N and associated data A, the pair (A, C) remains valid for any nonce N', and thus breaking authenticity is trivial. Xor'ing $\mathsf{pad}(N)$ to R ensures that the tag T depends on all of N, A, C.

Farshim, Orlandi, and Roşie [25] also point out that in Encrypt-and-MAC, if the encryption scheme and the PRF can use the same key, and the PRF is committing, then the composition has key-committing security. Their result is however for probabilistic AE, so it does not imply the key-committing security of CAU-C1.

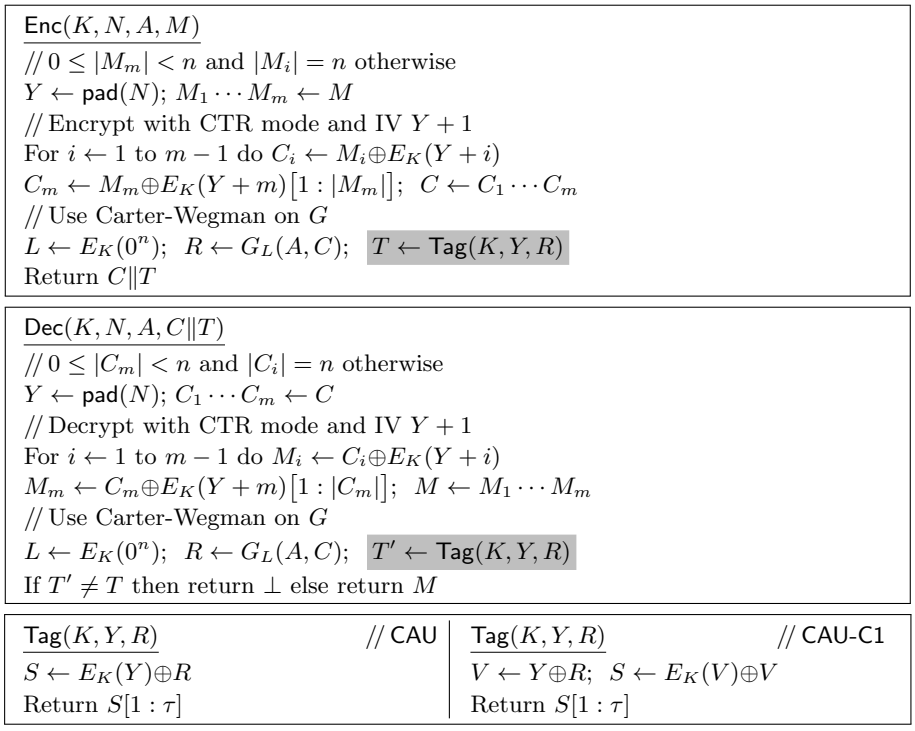

Fig. 8. The common blueprint for encryption (top) and decryption (middle) of $\mathsf{CAU}[E, G, \tau]$ and $\mathsf{CAU\text{-}C1}[E, G, \tau]$. The two schemes only differ on how they implement the internal procedure Tag, as shown in the bottom panels.

Discussion. Our CAU-C1 scheme has several merits. (1) The change to CAU is small, making it easy to modify existing CAU code to get CAU-C1 code. (2) The speed of CAU-C1 is about the same as CAU for moderate and large messages. Moreover, the absence of any ciphertext overhead over CAU means there is no additional bandwidth cost. In contrast, prior proposed solutions [30, 4, 25, 22, 27] have to sacrifice either speed or bandwidth. (3) As we will show later, for short tag length, CAU-C1 has much better UNAE security than CAU.

It however does have some limitations. (1) Since it requires modifying CAU's code, one may not be able to use CAU-C1 in some legacy systems. (2) In the encryption algorithm of CAU-C1, the blockcipher call for the tag must be computed strictly after all other blockcipher calls are completed. In contrast, in CAU, all blockcipher calls can be done in parallel. This slowdown can be significant for tiny messages.

CMT-1 security of CAU-C1. The following Theorem 3 analyzes CMT-1 security of CAU-C1; the proof is in [8]. The result is in the standard model, although

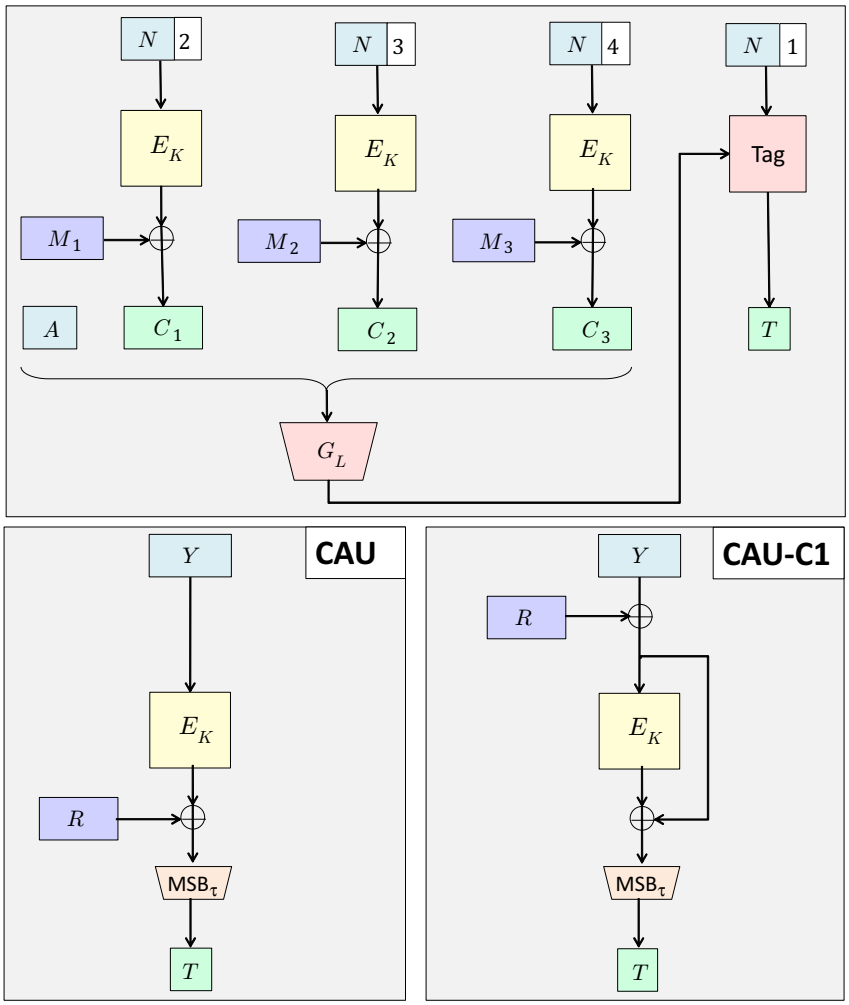

Fig. 9. A pictorial comparison of the encryption schemes of CAU and CAU-C1. The two scheme have the same blueprint on the top panel. They however have different implementations for the internal procedure Tag, illustrated in the bottom panels. Here the trapezoid MSB_τ outputs the τ-bit prefix of the input.

---

it relies on the multi-collision of the truncated Davies-Meyer that is justified in the ideal-cipher model via Proposition 2.

Theorem 3. *Let* CAU-C1$[E, G, \tau]$ *be as above. Let* $s \geq 2$ *be an integer. Then for any adversary* $\mathcal{A}$*, we can construct an adversary* $\mathcal{B}$ *such that*

$$\mathbf{Adv}^{\text{cmt-1}}_{\text{CAU-C1}[E,G,\tau],s}(\mathcal{A}) \leq \mathbf{Adv}^{\text{coll}}_{\text{DM}[E,\tau],s}(\mathcal{B}) \ .$$

Adversary $\mathcal{B}$ *runs* $\mathcal{A}$ *and makes* s *other calls on* E*.*

Discussion. Note that an adversary can break the two-way CMT-1 security of $\mathsf{CAU\text{-}C1}[E, G, \tau]$ by using about $2^{\tau/2}$ operations. If one aims for at least birthday-bound security and one's application requires two-way CMT-1 security, we must not truncate the tag, namely τ must be 128. However, if we only need to resist the Partitioning-Oracle attack and can tolerate a small speedup in adversarial password search, we can use, say $\tau = 96$. From Proposition 2, with $\tau = 96$, for any adversary $\mathcal{B}$ that spends at most 2^{64} operations, it can find a 5-way multi-collision on $\mathsf{DM}[E, \tau]$ with probability at most 2^{-60}, and thus $\mathcal{B}$ can at best speed up its password searching by a factor of four.

Unique-nonce security of CAU-C1. For the scheme $\mathsf{CAU\text{-}C1}[E, G, \tau]$ to have unique-nonce security, in addition for the hash G to be AXU, we also need it to be *weakly regular*, a notion that we define below.

Let $G : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ be a keyed hash function. We say that G is *weakly c-regular* if $G_K(\varepsilon, \varepsilon) = 0^n$ for every $K \in \{0,1\}^n$, and for all $Y \in \{0,1\}^n$ and $(A, M) \in \{0,1\}^* \times \{0,1\}^* \backslash (\varepsilon, \varepsilon)$,

$$\Pr_{K \xleftarrow{\$} \{0,1\}^n}[G_K(A, M) = Y] \le \frac{c \cdot (|M|_n + |A|_n)}{2^n} \quad .$$

Why does CAU-C1 need a weakly regular hash function? In CAU-C1, in each encryption, we encrypt the i-th block of the message by running the blockcipher on $\mathsf{pad}(N) + i$, and obtain the tag by calling the blockcipher on $V \leftarrow \mathsf{pad}(N) \oplus R$, where R is the output of the hash G. The weak regularity of G ensures that these inputs are different. In contrast, CAU obtains the tag by running the blockcipher on $\mathsf{pad}(N)$, and thus does not need a weakly regular hash.

In [8] we show that the hash function GHASH of GCM is weakly 1.5-regular. The following result confirms that CAU-C1 has good unique-nonce security. The proof is in [8].

Theorem 4. *Let* $\mathsf{CAU\text{-}C1}[E, G, \tau]$ *be as above, building on top of a c-AXU, weakly c-regular hash function* G *and a blockcipher* $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. *Then for an adversary* $\mathcal{A}$ *that makes at most* q *queries of* σ *blocks and* q_v *verification queries in total, with at most* B *blocks per user, we can construct another* $\mathcal{B}$ *of at most* $\sigma + q$ *queries such that*

$$\mathbf{Adv}^{\mathrm{unae}}_{\mathsf{CAU\text{-}C1}[E,G,\tau]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{B}) + \frac{(4c+2)B\sigma + (2c+2)Bq}{2^n} + \frac{2q_v}{2^\tau} \quad .$$

The running time of $\mathcal{B}$ *is about that of* $\mathcal{A}$ *plus the time to use* G *on* $\mathcal{A}$*'s messages and associated data.*

On short tags. When the tag length τ is short, CAU-C1 has much better unique-nonce security than CAU. In particular, Ferguson [26] gives a (single-user) attack of q_v decryption queries, each of ℓ blocks, to break the security of CAU with advantage $q_v \ell / 2^\tau$. In contrast, CAU-C1 enjoys a smaller term $q_v / 2^\tau$.

CAU-C4 for CMT-4-security. Applying the HtE transform of Section 3, with a suitable choice of H, to CAU-C1, yields a CMT-4 and UNAE scheme that we call CAU-C4. There is no increase in ciphertext size. The computational overhead,

running H on the key, nonce and associated data, is independent of the message length.

6 A Committing Variant of AES-GCM-SIV

In this section, we describe a close variant CAU-SIV-C1 of AES-GCM-SIV that achieves both CMT-1 and misuse-resistance security with the same speed and bandwidth costs as AES-GCM-SIV. In this entire section, let n be an even integer, and let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher, with $k \in \{n, 2n\}$. We will consider a generalization CAU-SIV of AES-GCM-SIV that we describe below. The name CAU-SIV is a mnemonic for the use of (i) the classic SIV paradigm [41] in achieving misuse-resistance security, (ii) (a variant of) the CTR mode and (iii) an AXU hash function.

THE PRF GMAC$^+$. Like CAU, the scheme CAU-SIV is based on a c-AXU hash. As shown in [20], the hash function POLYVAL of AES-GCM-SIV is 1.5-AXU. In CAU-SIV, the AXU hash function is used to build a PRF that Bose, Hoang, and Tessaro [20] call GMAC$^+$. We begin with the description of this PRF.

For strings X and Y such that $|X| < |Y| = n$, let $X \boxplus Y$ denote the string obtained by setting the first bit of $(0^{n-|X|}\|X) \oplus Y$ to 0. Let $r < n$ be an integer, and let $\mathcal{N} = \{0,1\}^r$. Define $\mathsf{GMAC}^+[E,G] : \{0,1\}^{k+n} \times \mathcal{N} \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ via

$$\mathsf{GMAC}^+[E,G](K_{\mathsf{in}}\|K_{\mathsf{out}}, N, A, M) = E\big(K_{\mathsf{out}}, X\big) \ ,$$

where $X \leftarrow N \boxplus G(K_{\mathsf{in}}, M, A)$. See Fig. 11 for an illustration of GMAC$^+$.

THE KEY-DERIVATION FUNCTION KD1. In each encryption, CAU-SIV derives subkeys by applying a key-derivation function (which we call KD1) on the given nonce. Specifically, KD1 is exactly the ITP hash function in Section 4 with padding $\mathsf{pad}(N,i) = N\|[i]_{n-r}$, where $[i]_{n-r}$ denote an $(n-r)$-bit encoding of an integer i. The code of KD1 is given in the second-top panel of Fig. 10 for completeness.

CTR MODE. CAU-SIV is based on the following variant of the CTR mode. Let $r < n$ be an integer. (For AES-GCM-SIV, $r = 96$ and $n = 128$.) Let add be an operation on $\{0,1\}^n \times \{0,1,\ldots,2^{n-r}-1\}$ such that

$$\mathsf{add}(X,i) = 1\|X[2:r]\|(X[r+1:n] + i \bmod 2^{n-r}) \ .$$

The encryption and decryption schemes of $\mathsf{CTR}[E,\mathsf{add}]$ are defined in the bottom panels of Fig. 10. They are essentially the same as the standard CTR mode, except that they use the add operation instead of the modular addition in mod 2^n.

THE SCHEME CAU-SIV. The scheme $\mathsf{CAU\text{-}SIV}[E,G,\mathsf{add}]$ is described in Fig. 10. Informally, one first uses KD1 on the given nonce to derive subkeys $K_{\mathsf{in}} \in \{0,1\}^n$ and $K_{\mathsf{out}} \in \{0,1\}^k$. One then follows the classic SIV paradigm [41] in building a misuse-resistant AE scheme: first use the PRF GMAC$^+$ on the triple (N,A,M) to derive an initialization vector IV, and then run CTR with that particular IV to encrypt M. However, unlike the standard SIV with key separation, here both

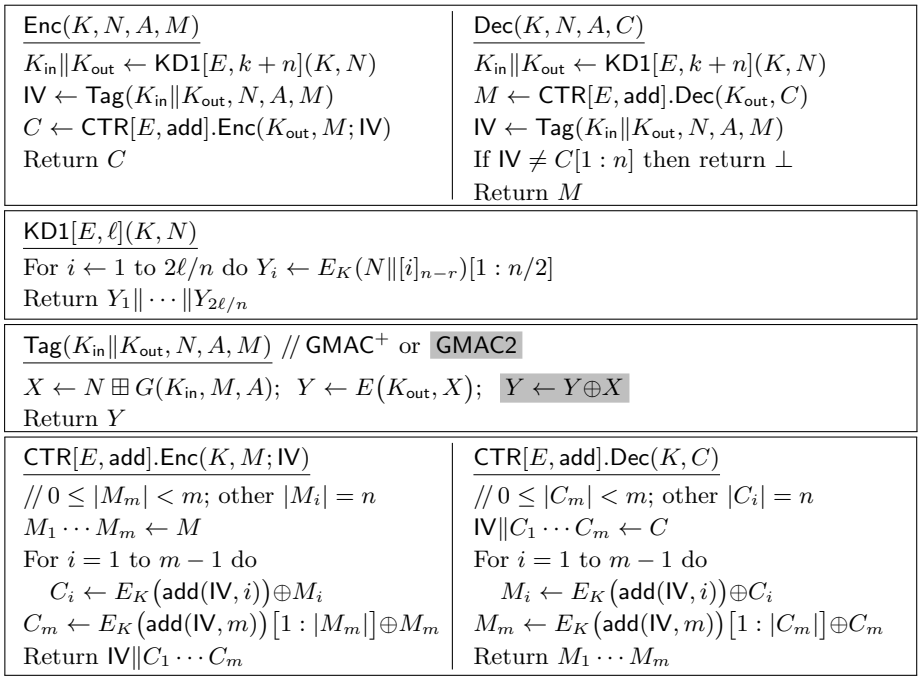

Fig. 10. The schemes CAU-SIV and CAU-SIV-C1 whose encryption and decryption schemes are given in the top-left and top-right panels, respectively. Procedure Tag implements GMAC$^+$ (for CAU-SIV) or GMAC2 (for CAU-SIV-C1); the latter contains the highlighted code, but the former does not.

GMAC$^+$ and CTR use E on the same key K_{out}. There is, however, a domain separation in the use of the blockcipher: GMAC$^+$ will only run E on an input whose most significant bit is 0, whereas CTR runs E on inputs of most significant bit 1.

THE CAU-SIV-C1 SCHEME. We now show how to add CMT-1 security to CAU-SIV. Recall that CAU-SIV internally uses a PRF GMAC$^+$ that is based on an AXU, weakly regular hash function G. The scheme CAU-SIV-C1 introduces an extra xor in GMAC$^+$, resulting in a new PRF construction that we call GMAC2, and that is the only difference between the two AE schemes. In particular,

$$\mathsf{GMAC}^+[E,G](K_{\mathsf{in}}\|K_{\mathsf{out}},N,A,M) = E\big(K_{\mathsf{out}},X\big)\ ,$$

where $X \leftarrow N \boxplus G(K_{\mathsf{in}},M,A)$. In contrast, GMAC2 employs the Davies-Meyer construction to break the invertibility of E, namely,

$$\mathsf{GMAC2}[E,G](K_{\mathsf{in}}\|K_{\mathsf{out}},N,A,M) = E\big(K_{\mathsf{out}},X\big)\oplus X\ .$$

See Fig. 11 for a side-by-side pictorial comparison of GMAC$^+$ and GMAC2. The code of CAU-SIV-C1 is given in Fig. 10.

The difference of CAU-SIV-C1 and CAU-SIV is tiny, just a single xor. As a result, the speed and bandwidth costs of CAU-SIV-C1 are about the same as

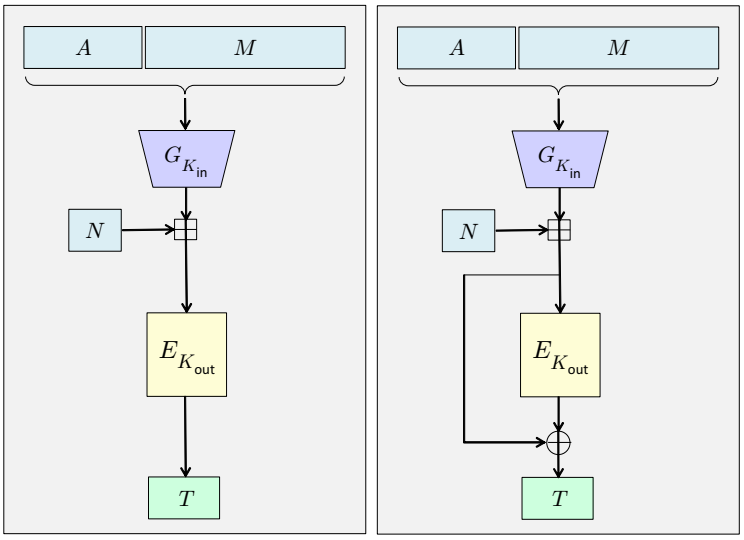

Fig. 11. The GMAC$^+$ construction (left) and its variant GMAC2 (right).

CAU-SIV for all message sizes. While one must intrusively modify CAU-SIV's code to obtain CAU-SIV-C1, since CAU-SIV is new, we anticipate that there will be very few legacy situations that one cannot adopt CAU-SIV-C1.

Committing Security of CAU-SIV-C1. Theorem 5 below confirms that the extra xor indeed hardens CAU-SIV-C1, ensuring CMT-1 security. The proof is in [8]. Intuitively, the synthetic IV of CAU-SIV-C1 is obtained by a two-step chain of hashing: (i) first use the Iterative Truncated Permutation construction $\mathsf{ITP}[E, r, n]$ to commit the master key K and the nonce N to the n-bit prefix of the blockcipher subkey K_{out}, and then (ii) use the Davies-Meyer construction $\mathsf{DM}[E]$ to commit K_{out}. We show in [8] how this allows the CMT-1 security of CAU-SIV-C1 to reduce to the multi-collision resistance of $\mathsf{ITP}[E, r, n]$ and $\mathsf{DM}[E]$, both of which we justify with good bounds.

Theorem 5. *Let* $\mathsf{SE} = \mathsf{CAU\text{-}SIV\text{-}C1}[E, G, \mathsf{add}]$ *be as described above, building on top of a blockcipher* $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^k$. *Let* $r < n$ *be the nonce length. Let* $s \geq 2$ *be an integer, and let* $t = \lceil \sqrt{s} \rceil$. *Then for any adversary* $\mathcal{A}$, *we can construct adversaries* $\mathcal{D}_0$ *and* $\mathcal{D}_1$ *such that*

$$\mathbf{Adv}^{\text{cmt-1}}_{\mathsf{SE},s}(\mathcal{A}) \leq \max\{\mathbf{Adv}^{\text{coll}}_{\mathsf{ITP}[E,r,n],t}(\mathcal{D}_0), \mathbf{Adv}^{\text{coll}}_{\mathsf{DM}[E],t}(\mathcal{D}_1)\} \ .$$

Each of $\mathcal{D}_0$ *and* $\mathcal{D}_1$ *runs* $\mathcal{A}$ *and then makes at most* $6s$ *other blockcipher calls.*

Misuse-Resistance Security of CAU-SIV-C1. The following result shows that CAU-SIV-C1 also has good misuse-resistance security; the proof is in [8].

Game $\mathbf{G}^{\mathrm{bind}}_{\mathsf{F},s}(\mathcal{A})$
$(K_1, M_1, \ldots, K_s, M_s) \leftarrow\!\!\$\ \mathcal{A}$ // $(K_1, M_1), \ldots, (K_s, M_s)$ must be distinct
For $i \leftarrow 1$ to s do $(P_i, L_i) \leftarrow \mathsf{F}(K_i, M_i)$
Return $(P_1 = \cdots = P_s)$

Fig. 12. Game defining the binding security of a committing PRF F.

Theorem 6. *Let* $\mathsf{SE} = \mathsf{CAU\text{-}SIV\text{-}C1}[E, G, \mathsf{add}]$ *be as described above, building on top of a c-AXU hash function* G *and a blockcipher* $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$*. Then for any adversary* $\mathcal{A}$ *that makes at most* q *queries of totally* σ *blocks with at most* B *blocks per (user, nonce) pair and* D *queries per user, we can construct an adversary* $\mathcal{B}$ *of* $\max\{6q, \sigma + q\}$ *queries such that*

$$\mathbf{Adv}^{\mathrm{mrae}}_{\mathsf{SE}}(\mathcal{A}) \le 2 \cdot \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{6\sqrt{nDq}}{2^{3n/4}} + \frac{7\sigma B + (2c+7)qB}{2^n} \quad .$$

The running time of $\mathcal{B}$ *is at most that of* $\mathcal{A}$ *plus the time to encrypt/decrypt the latter's queries.*

CAU-SIV-C4 FOR CMT-4-SECURITY. Applying the HtE transform of Section 3, with a suitable choice of H, to $\mathsf{CAU\text{-}SIV\text{-}C1}$, yields a CMT-4 and MRAE scheme that we call $\mathsf{CAU\text{-}SIV\text{-}C4}$. There is no increase in ciphertext size. The computational overhead is independent of the message length.

7 Adding Key-Committing Security To Legacy AE

In this section, we describe two generic methods UNAE-then-Commit (UtC) and MRAE-then-Commit (RtC) that transform an AE scheme SE into a CMT-1-secure one. The former preserves unique-nonce security, whereas the latter preserves misuse-resistance security. As a stepping stone, we define a new primitive that we call *committing PRF*, which we will describe below.

COMMITTING PRFS. A *committing PRF* F is a deterministic algorithm, and associated with a message space $\mathcal{M}$ and key space $\{0,1\}^k$. It takes as input a key $K \in \{0,1\}^k$ and a message $M \in \mathcal{M}$, and then produces $(P, L) \in \{0,1\}^\ell \times \{0,1\}^\lambda$. We refer to ℓ as the *commitment length* of F, and λ as the *mask length* of F.

We require that F be a good PRF, meaning that its outputs (P, L) are indistinguishable from $(P^*, L^*) \leftarrow\!\!\$\ \{0,1\}^\ell \times \{0,1\}^\lambda$. In addition, for an adversary $\mathcal{A}$ and an integer $s \ge 2$, we define the advantage of $\mathcal{A}$ breaking the s-way binding security of F as $\mathbf{Adv}^{\mathrm{bind}}_{\mathsf{F},s}(\mathcal{A}) = \Pr[\mathbf{G}^{\mathrm{bind}}_{\mathsf{F},s}(\mathcal{A})]$, where game $\mathbf{G}^{\mathrm{bind}}_{\mathsf{F},s}(\mathcal{A})$ is defined in Fig. 12. Informally, a committing PRF is a combination of a PRF and a commitment scheme, where the string P is a commitment of the key K and the message M.

For $s = 2$, our notion of committing PRF can be viewed as a PRF counterpart of the notion of *right collision-resistant PRG* in [25]. We however will

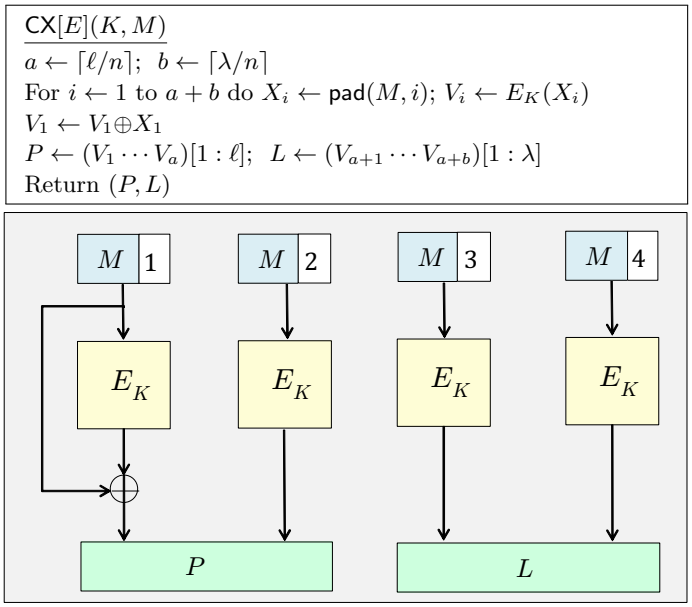

Fig. 13. The committing PRF scheme $\mathsf{CX}[E, \mathsf{pad}]$, illustrated for the case $\ell = \lambda = 2n$ and $\mathsf{pad}(M, i)$ is the concatenation of M and an $(n - m)$-bit encoding of i.

give practical instantiations via a blockcipher whereas the construction in [25] is theoretical, using hardcore predicates.

An efficient committing PRF. We now describe an efficient committing PRF Counter-then-Xor (CX) that is built on top of a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. Here the message space $\mathcal{M} = \{0,1\}^m$ and the key space is $\{0,1\}^k$, with $m < n$. Let pad denote a one-to-one encoding that turns a pair $(M, i) \in \{0,1\}^m \times \{1, \ldots, 2^{n-m}\}$ into an n-bit string. The commitment length $\ell \geq n$ and the mask length λ satisfy $\lceil \ell/n \rceil + \lceil \lambda/n \rceil \leq 2^{n-m}$. The construction $\mathsf{CX}[E, \mathsf{pad}]$ is shown in Fig. 13.

The following result shows that CX is a good committing PRF scheme. Part (a) is a straightforward application of the (multi-user) PRP/PRF Switching Lemma, with an observation that for each query that $\mathcal{A}_0$ makes to CX, it translates to $d = \lceil \ell/n \rceil + \lceil \lambda/n \rceil$ PRP queries on the blockcipher. For applications in this paper, $d \leq 5$. Part (b) is a direct corollary of Proposition 2, since the first block of P is obtained from the Davies-Meyer construction $\mathsf{DM}[E]$.

Proposition 5. *Let* $\mathsf{CX}[E, \mathsf{pad}]$ *be as above, and let* $s \geq 2$ *be an integer. Let* $d = \lceil \ell/n \rceil + \lceil \lambda/n \rceil$.

a) For any adversary $\mathcal{A}_0$ *making* q *queries in total with at most* B *queries per user, we can construct an adversary* $\mathcal{B}$ *of about the same running time that makes*

$\underline{\mathsf{UtC}[\mathsf{F},\mathsf{SE}].\mathsf{Enc}(K,N,A,M)}$	$\underline{\mathsf{UtC}[\mathsf{F},\mathsf{SE}].\mathsf{Dec}(K,N,A,P^*\|\|C)}$
$(P,L) \leftarrow \mathsf{F}(K,N)$	$(P,L) \leftarrow \mathsf{F}(K,N)$
$C \leftarrow \mathsf{SE}.\mathsf{Enc}(L,N,A,M)$	If $P^* \neq P$ then return $\perp$
Return $P\|\|C$	Else return $\mathsf{SE}.\mathsf{Dec}(L,N,A,C)$

Fig. 14. The encryption (left) and decryption (right) schemes of the resulting AE scheme under the UtC transform.

at most dq queries such that

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CX}[E,\mathsf{pad}]}(\mathcal{A}_0) \le \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{d^2 \cdot Bq}{2^n} \ .$$

b) For any adversary $\mathcal{A}_1$, we can construct another adversary $\mathcal{B}$ of about the same running time and resources such that

$$\mathbf{Adv}^{\mathrm{bind}}_{\mathsf{CX}[E,\mathsf{pad}],s}(\mathcal{A}_1) \le \mathbf{Adv}^{\mathrm{coll}}_{\mathsf{DM}[E],s}(\mathcal{B}) \ .$$

The UNAE-then-Commit (UtC) transform. Let SE be an AE scheme with key space $\{0,1\}^k$ and nonce space $\mathcal{N}$. Let F be a committing PRF scheme of message space $\mathcal{N}$ and mask length k. The scheme $\mathsf{UtC}[\mathsf{F},\mathsf{SE}]$ is shown in Fig. 14. Informally, under UtC, a ciphertext contains a commitment P of the master key K, ensuring CMT-1 security. The security of $\mathsf{UtC}[\mathsf{F},\mathsf{SE}]$ is analyzed below; the proof is in [8].

Theorem 7. *Let SE and F be as above. Let $s \ge 2$ be an integer.*
a) For any adversary $\mathcal{A}_0$, we can construct an adversary $\mathcal{B}_0$ of about the same running time and using the same resources as $\mathcal{A}_0$ such that

$$\mathbf{Adv}^{\text{cmt-1}}_{\mathsf{UtC}[\mathsf{F},\mathsf{SE}],s}(\mathcal{A}_0) \le \mathbf{Adv}^{\mathrm{bind}}_{\mathsf{F},s}(\mathcal{B}_0) \ .$$

b) For any adversary $\mathcal{A}_1$ of at most B queries per (user, nonce) pair, we can construct an adversary $\mathcal{B}_1$ and $\mathcal{B}_2$ such that

$$\mathbf{Adv}^{\mathrm{unae}}_{\mathsf{UtC}[\mathsf{F},\mathsf{SE}]}(\mathcal{A}_1) \le \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{F}}(\mathcal{B}_1) + \mathbf{Adv}^{\mathrm{unae}}_{\mathsf{SE}}(\mathcal{B}_2) \ .$$

The running time of $\mathcal{B}_1$ is about that of $\mathcal{A}_1$ plus the time to encrypt/decrypt the queries of $\mathcal{A}_1$ via SE, and its queries statistics is the same as $\mathcal{A}_1$. Adversary $\mathcal{B}_2$ has the same number of queries and the total query length as $\mathcal{A}_1$, but it makes at most B queries per user. It has about the same running time as $\mathcal{A}_1$.

Discussion. Albertini et al. [4] also give a generic transform. (An instantiation of this transform is now deployed in the latest version of the AWS Encryption SDK, an open-source client-side encryption library [1].) It can be viewed as a specific instantiation of UtC, in which the committing PRF F is built on top of two collision-resistant PRFs. One of these two collision-resistant PRFs however may have to provide up to 256-bit output (since this output is used as a key of the legacy SE), obstructing an obvious instantiation via Davies-Meyer on

$\mathsf{RtC}[\mathsf{F},\mathsf{SE},H].\mathsf{Enc}(K,N,A,M)$	$\mathsf{RtC}[\mathsf{F},\mathsf{SE},H].\mathsf{Dec}(K,N,A,T\|\|C)$
$(P,L) \leftarrow \mathsf{F}(K,N)$	$(P,L) \leftarrow \mathsf{F}(K,N)$
$C \leftarrow \mathsf{SE}.\mathsf{Enc}(L,N,A,M)$	$T^* \leftarrow H(P,C[1:n])$
$T \leftarrow H(P,C[1:n])$	If $T \neq T^*$ then return $\perp$
Return $T\|\|C$	Return $\mathsf{SE}.\mathsf{Dec}(L,N,A,C)$

Fig. 15. The encryption (left) and decryption (right) algorithms of the scheme given by the RtC transform.

AES. As a result, Albertini et al. instantiate them via SHA-256. Not only is this instantiation slower than our Count-then-Xor construction, but using it in UtC also requires an additional primitive in addition to AES. In addition, we realize that UtC achieves CMT-1 security, whereas Albertini et al. only claim key-committing security.

The MRAE-then-Commit (RtC) transform. Let SE be an AE scheme with key space $\{0,1\}^\lambda$ and nonce space $\mathcal{N}$. Let F be a committing PRF scheme of message space $\mathcal{N}$, key space $\{0,1\}^k$, commitment length ℓ, and mask length λ (that is also the key length of SE). Assume that each ciphertext in SE is at least n-bit long. Let $H : \{0,1\}^\ell \times \{0,1\}^n \to \{0,1\}^n$ be a collision-resistant PRF. We can instantiate F via CX, and H via the Davies-Meyer construction. (The PRF security of this particular choice of H can be trivially obtained from Lemma 1.) The scheme $\mathsf{RtC}[\mathsf{F},\mathsf{SE},H]$ is shown in Fig. 15. Intuitively, RtC creates a two-step chain of commitments $K \to P \to T$, where K is the master key, P is the commitment generated by F, and T is the hash output, which is a part of the ciphertext. This leads to an underlying cascade of two hash functions whose collision resistance an adversary has to break in order to break the CMT-1 security of $\mathsf{RtC}[\mathsf{F},\mathsf{SE},H]$. Thus from Proposition 4, the CMT-1 security of $\mathsf{RtC}[\mathsf{F},\mathsf{SE},H]$ is reduced to the committing security of F and the collision resistance of H. The proof of the following is in [8].

Theorem 8. *Let* SE *and* F *be as above.*
a) Let $s \geq 2$ be an integer, and let $t = \lceil \sqrt{s} \rceil$. For any adversary $\mathcal{A}_0$, we can construct adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$ such that

$$\mathbf{Adv}^{\text{cmt-1}}_{\mathsf{RtC}[\mathsf{F},\mathsf{SE},H],s}(\mathcal{A}_0) \leq \max\left\{\mathbf{Adv}^{\text{bind}}_{\mathsf{F},t}(\mathcal{B}_0), \mathbf{Adv}^{\text{coll}}_{H,t}(\mathcal{B}_1)\right\} \ .$$

Each of $\mathcal{B}_0$ and $\mathcal{B}_1$ runs $\mathcal{A}_0$, and then runs $\mathsf{RtC}[\mathsf{F},\mathsf{SE},H]$ to encrypt one out of the s messages that $\mathcal{A}_0$ outputs, and then evaluates F *on s inputs.*

b) For any adversary $\mathcal{A}_1$ of at most B queries per (user, nonce) pair and at most q queries, we can construct adversaries $\mathcal{B}_2$, $\mathcal{B}_3$, and $\mathcal{B}_4$ such that

$$\mathbf{Adv}^{\text{mrae}}_{\mathsf{RtC}[\mathsf{F},\mathsf{SE},H]}(\mathcal{A}_1) \leq \mathbf{Adv}^{\text{prf}}_{\mathsf{F}}(\mathcal{B}_2) + \mathbf{Adv}^{\text{mrae}}_{\mathsf{SE}}(\mathcal{B}_3) + \mathbf{Adv}^{\text{prf}}_{H}(\mathcal{B}_4) + \frac{Bq}{2^n} \ .$$

Adversary $\mathcal{B}_2$ has the same query statistics as $\mathcal{A}_1$, and its running time is at most that of $\mathcal{A}_1$ plus the time to use RtC *to encrypt/decrypt the latter's queries.*

Adversaries $\mathcal{B}_3$ and $\mathcal{B}_4$ have the same number of queries and the total query length as $\mathcal{A}_1$, but they make only B queries per user. The running time of $\mathcal{B}_3$ is about that of $\mathcal{A}_1$ plus the time to run H on q inputs, and $\mathcal{B}_4$ has about the same running time as $\mathcal{A}_1$.

Connection to libsodium's approach. The libsodium library [2] suggests the following transformation to add key-committing security to an AE scheme SE. Assume that a ciphertext of SE can be parsed as a concatenation of a tag T and a ciphertext core C^*. Let $H : \{0,1\}^* \to \{0,1\}^m$ be a cryptographic hash function. To encrypt (N, A, M) under key K, let $T\|C^* \leftarrow \mathsf{SE.Enc}(K, N, A, M)$, let $T^* \leftarrow H(K\|N\|T)$, and output $T^*\|T\|C^*$. To decrypt $(N, A, T^*\|T\|C^*)$ with key K, first check if $T^* = H(K\|N\|T)$. If they agree then return $\mathsf{SE.Dec}(K, N, A, T\|C^*)$, else return $\perp$.

The transform above works for the AE schemes in the libsodium libraries (namely GCM and ChaChaPoly1305) if we model (i) the hash function H as a random oracle, (ii) AES as an ideal cipher, and (iii) ChaCha20 permutation as an ideal permutation. The RtC transform can be viewed as a way to refine libsodium's approach to (i) work with a generic AE scheme and (ii) instantiate the hash function via the Davies-Meyer construction instead of SHA-256. While the libsodium's transform is suggested for unique-nonce security, we points out that RtC also works for misuse-resistance security.

Acknowledgments

We thank the EUROCRYPT 2022 reviewers for their careful reading and valuable comments. Mihir Bellare was supported in part by NSF grant CNS-1717640 and a gift from Microsoft. Viet Tung Hoang was supported in part by NSF grants CNS-2046540 (CAREER), CICI-1738912, and CRII-1755539.

References

1. AWS Encryption SDK 2.0. `https://docs.aws.amazon.com/encryption-sdk/latest/developer-guide/introduction.html`, 2020.
2. The Sodium cryptography library (Libsodium). `https://libsodium.gitbook.io/doc`, 2021.
3. M. Abdalla, M. Bellare, and G. Neven. Robust encryption. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, Springer 2010.
4. A. Albertini, T. Duong, S. Gueron, S. Kölbl, A. Luykx, and S. Schmieg. How to abuse and fix authenticated encryption without key commitment. In *31st USENIX Security Symposium*, 2022.
5. M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In N. Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, Springer 1996.
6. M. Bellare, R. Canetti, and H. Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th FOCS*. IEEE 1996.
7. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *38th FOCS*. IEEE 1997.

8. M. Bellare and V. T. Hoang. Efficient schemes for committing authenticated encryption. Cryptology ePrint Archive, 2022. https://ia.cr/2022/
9. M. Bellare and T. Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, Springer 2003.
10. M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, Springer 2000.
11. M. Bellare, R. Ng, and B. Tackmann. Nonces are noticed: AEAD revisited. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, Springer 2019.
12. M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, Springer 2006.
13. M. Bellare and B. Tackmann. The multi-user security of authenticated encryption: AES-GCM in TLS 1.3. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, Springer 2016.
14. M. Bellare and B. Tackmann. Nonce-based cryptography: Retaining security when randomness fails. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, Springer 2016.
15. D. Bernstein. Chacha, a variant of salsa20. In *Workshop record of SASC*, volume 8, pages 3–5, 2008.
16. D. Bernstein. The salsa20 family of stream ciphers. In *New stream cipher designs: The eSTREAM finalists, Lecture Notes in Computer Science*, volume 4986. Springer, 2008.
17. D. J. Bernstein. The poly1305-AES message-authentication code. In H. Gilbert and H. Handschuh, editors, *FSE 2005*, volume 3557 of *LNCS*, Springer 2005.
18. J. Black, P. Rogaway, and T. Shrimpton. Black-box analysis of the block-cipher-based hash-function constructions from PGV. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, Springer 2002.
19. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, Springer 2004.
20. P. Bose, V. T. Hoang, and S. Tessaro. Revisiting AES-GCM-SIV: Multi-user security, faster key derivation, and better bounds. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, Springer 2018.
21. J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-Damgård revisited: How to construct a hash function. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, Springer 2005.
22. Y. Dodis, P. Grubbs, T. Ristenpart, and J. Woodage. Fast message franking: From invisible salamanders to encryptment. In H. Shacham and A. Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, Springer 2018.
23. M. Dworkin. Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D, November 2007.
24. P. Farshim, B. Libert, K. G. Paterson, and E. A. Quaglia. Robust encryption, revisited. In K. Kurosawa and G. Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, Springer 2013.
25. P. Farshim, C. Orlandi, and R. Roşie. Security of symmetric primitives under incorrect usage of keys. *IACR Trans. Symm. Cryptol.*, 2017(1):449–473, 2017.

26. N. Ferguson. Authentication weaknesses in GCM. Manuscript, available in NIST webpage, 2005.
27. P. Grubbs, J. Lu, and T. Ristenpart. Message franking via committing authenticated encryption. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, Springer 2017.
28. S. Gueron and Y. Lindell. Better bounds for block cipher modes of operation via nonce-based key derivation. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 2017*, 2017.
29. V. T. Hoang, S. Tessaro, and A. Thiruvengadam. The multi-user security of GCM, revisited: Tight bounds for nonce randomization. In D. Lie, M. Mannan, M. Backes, and X. Wang, editors, *ACM CCS 2018*, 2018.
30. S. Jarecki, H. Krawczyk, and J. Xu. OPAQUE: An asymmetric PAKE protocol secure against pre-computation attacks. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, Springer 2018.
31. B. Kaliski. PKCS #5: Password-Based Cryptography Specification Version 2.0. RFC 2898, Sep. 2000. `https://datatracker.ietf.org/doc/html/rfc2898`.
32. J. Katz and M. Yung. Unforgeable encryption and chosen ciphertext secure modes of operation. In B. Schneier, editor, *FSE 2000*, volume 1978 of *LNCS*, Springer 2001.
33. M. Lambæk. Breaking and fixing private set intersection protocols. Cryptology ePrint Archive, Report 2016/665, 2016. `https://eprint.iacr.org/2016/665`.
34. J. Len, P. Grubbs, and T. Ristenpart. Partitioning oracle attacks. In M. Bailey and R. Greenstadt, editors, *30th USENIX Security Symposium*. USENIX Association, 2021.
35. A. Luykx, B. Mennink, and K. G. Paterson. Analyzing multi-key security degradation. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, Springer 2017.
36. D. A. McGrew and J. Viega. The security and performance of the Galois/counter mode (GCM) of operation. In A. Canteaut and K. Viswanathan, editors, *INDOCRYPT 2004*, volume 3348 of *LNCS*, Springer 2004.
37. N. Mouha and A. Luykx. Multi-key security: The Even-Mansour construction revisited. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, Springer 2015.
38. C. Namprempre, P. Rogaway, and T. Shrimpton. Reconsidering generic composition. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, Springer 2014.
39. P. Rogaway. Authenticated-encryption with associated-data. In V. Atluri, editor, *ACM CCS 2002*, 2002.
40. P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In M. K. Reiter and P. Samarati, editors, *ACM CCS 2001*, 2001.
41. P. Rogaway and T. Shrimpton. A provable-security treatment of the key-wrap problem. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, Springer 2006.
42. J. Salowey, A. Choudhury, and D. McGrew. AES Galois Counter Mode (GCM) Cipher Suites for TLS. RFC 5288, Aug. 2008. `https://datatracker.ietf.org/doc/html/rfc5288`.
43. J. Salowey, A. Choudury, and D. A. McGrew. AES Galois Counter Mode (GCM) cipher suites for TLS. RFC 5288, August 2008.