A preliminary version of this paper appears in the proceedings of EUROCRYPT 2022. This is the full version.

Efficient Schemes for Committing Authenticated Encryption

Mihir Bellare[1] Viet Tung Hoang[2]

March 2, 2022

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] Department of Computer Science & Engineering, University of California San Diego, USA. Email: mihir@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~mihir/. Supported in part by NSF grant CNS-1717640 and a gift from Microsoft.

[2] Department of Computer Science, Florida State University, USA. Email: tvhoang@cs.fsu.edu. URL: cs.fsu.edu/~tvhoang/. Supported in part by NSF grants CNS-2046540 (CAREER), CICI-1738912, and CRII-1755539.

Contents

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 [9, 37]. The underlying syntax, meanwhile, has gone from randomized or counter-based [7] to nonce-based [47, 46].

Recent attacks and applications [39, 29, 3, 26, 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 $M \leftarrow \mathsf{SE.Dec}(K, N, A, C)$ [47, 46]. 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 [47, 46]. MRAE (Misuse-resistant AE security) is stronger, asking in addition for best-possible security under any reuse of an encryption nonce [48].

A central scheme is GCM [41]. It is a government standard [24] and is used in TLS [49]. Other standardized and widely-used schemes are XSalsa20/Poly1305 and ChaCha20/Poly1305 [16, 14, 15]. All these are UNAE-secure. AES-GCM-SIV [50, 31] 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 [12] —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 [12, 40, 33], and likewise for the MRAE security of AES-GCM-SIV [19]. 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 [43]; 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 [39, 4, 29] show that GCM, XSalsa20/Poly1305, ChaCha20/Poly1305 and OCB [47] 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 [25]. Now calling it key-robustness, Farshi, Orlandi and Rǫsie (FOR) [26] bring it to randomized symmetric encryption. Albertini, Duong, Gueron, Kölbl, Luykx and Schmieg (ADGKLS) [4] and Len, Grubbs and Ristenpart (LGR) [39] consider it for nonce-based symmetric encryption, giving definitions slightly weaker than CMT-1.

Grubbs, Lu and Ristenpart (GLR) [29] 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 [29] call compact commitment.

Why commit to the key? The canonical method for password-based encryption (PKCS#5 [36]) uses a symmetric encryption scheme SE, such as GCM, as a tool. In a surprising new attack, LGR [39] 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 [18]; they also note that when encryption strives to be anonymous, key-committing security is necessary for unambiguous decryption. FOR [26] 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 [38]. 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 [29] 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 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 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,

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

Figure 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.

---

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 3.1 shows that SE-4 is CMT-4 assuming SE-1 is CMT-1 and H is collision resistant. Theorem 3.2 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 [41] is a UNAE scheme that, due to its standardization [24] and use in TLS [49], is already widely implemented. Attacks [39, 4, 29] 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 5.1 establishes CMT-1 security of CAU-C1, and Theorem 5.2 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 [50, 31], 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 6.5 establishes CMT-1 security of CAU-SIV-C1, and Theorem 6.6 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 [26]. We build a cheap committing PRF, that we call CX, from (only) a blockcipher. Proposition 7.1 proves its security with good bounds. Theorem 7.2 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 7.3 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 [39]. 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.

Our CMT-1 notion is strong, allowing the adversary to pick both keys, but in many applications, one key is randomly and honestly chosen and the adversary only gets to pick the other. This means that, for AES-based instantiations of our schemes in which tags are 128 bits, there are no known 2^{64} time attacks that violate security of these applications.

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 [26] 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 [29] 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 [23] is randomized, not nonce-based. It uses a compression function that is assumed collision resistant and RKA-PRF-secure [8]. 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 [29] and DGRW [23] 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 $\mathsf{CAU\text{-}C1}$, it increases ciphertext size.

LGR [39] 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* [43] 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.

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

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

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 [12]. 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 $C \leftarrow \textsc{Enc}(i, N, A, M)$ then later it is not allowed to query $\textsc{Vf}(i, N, A, C)$. 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 $\mathbf{Adv}^{\mathrm{unae}}_{\mathsf{SE}}(\mathcal{A})$ for clarity.

Systems and Transcripts. Following the notation from [32], it is convenient to consider interactions of a distinguisher $\mathcal{A}$ with an abstract system $\mathbf{S}$ which answers $\mathcal{A}$'s queries. The resulting interaction then generates a transcript $\theta = \big((X_1, Y_1), \ldots, (X_q, Y_q)\big)$ of query-answer pairs. It is known that $\mathbf{S}$ is entirely described by the probabilities $\mathsf{p}_{\mathbf{S}}(\theta)$ that correspond to the system $\mathbf{S}$ responding with answers as indicated by θ when the queries in θ are made.

We will generally describe systems informally, or more formally in terms of a set of oracles they provide, and only use the fact that they define corresponding probabilities $\mathsf{p}_{\mathbf{S}}(\theta)$ without explicitly giving these probabilities.

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

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

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

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

Figure 3: Games defining PRF security of F and PRP security of E.

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-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 [42] 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

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)

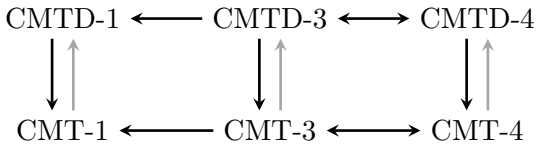

Figure 4: Games defining committing security of a symmetric encryption scheme SE. Below them are the associated what-is-committed functions WiC_ℓ, and then the relations between the notions. The **gray** arrows hold for tidy SE.

---

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 [26] 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 $\mathsf{SE.Enc}$.

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 $\mathbf{Adv}^{\text{cmt-}\ell}_{\mathsf{SE}}(\mathcal{A}) =$

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 $(\mathsf{SE.Enc}(K_1, N_1, A_1, M_1) = \cdots = \mathsf{SE.Enc}(K_s, N_s, A_s, M_s))$

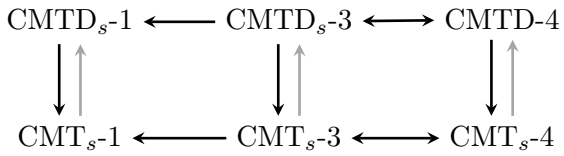

Figure 5: Games defining s-way committing security of a symmetric encryption scheme SE for $s \geq 2$. Below them are the relations between the notions. The **gray** arrows hold for tidy SE.

$\Pr[\mathbf{G}^{\text{cmt-}\ell}_{\mathsf{SE}}(\mathcal{A})]$. In the game, the adversary 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 Appendix A.

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 CMT-1-security of the scheme $\mathsf{CAU\text{-}C1}$ 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 $\mathsf{CAU\text{-}C1}$ is at most 2^{-62}. This allows us to offer a much stronger guarantee for situations like the Partitioning Oracle attack [39]. 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.

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

Figure 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.

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 Appendix L.

Theorem 3.1 *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*

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

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 Appendix M.

Theorem 3.2 *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

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

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

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

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 4.1 below generalizes the classical PRP/PRF Switching Lemma [11] to the multi-user setting; see Appendix D 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 [40], 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 4.1 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 [19].

Lemma 4.1 (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*

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

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 4.2 shows that one can consider only orderly adversaries in UNAE/MRAE notions with just a small loss in the advantage; see Appendix E for a proof. The idea of restricting to orderly adversaries has been used in prior works [19, 10]. 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 $\mathcal{B}$.

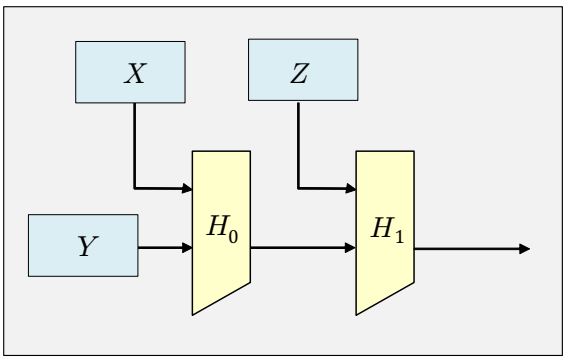

Figure 7: Illustration of the cascade of the two hash functions H_0 and H_1.

Proposition 4.2 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

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

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

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

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 [35] recommends the use of the Encrypt-then-HMAC construction, (ii) Albertini et al. [4] suggest using a hash-based key-derivation function to add key-committing security into legacy AE schemes; and (iii) Dodis et al. [23] 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 [17], 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 $m \le n$ 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 4.3 below analyzes the multi-collision resistance of $\mathsf{DM}[E,m]$; see Appendix F 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 4.3 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 [17]. For a general s, in Appendix C, 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 4.4 below analyzes the multi-collision resistance of ITP; see Appendix G 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 4.4 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 4.4. In Appendix H, we give an attack that matches this term, implying that the bound of Proposition 4.4 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 Appendix I for a proof.

Proposition 4.5 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 [13], 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 [26] 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.

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 [35, 4, 26, 23, 29] 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.

$\underline{\mathsf{Enc}(K, N, A, M)}$
// $0 \le |M_m| < n$ and $|M_i| = n$ otherwise
$Y \leftarrow \mathsf{pad}(N)$; $M_1 \cdots M_m \leftarrow M$
// Encrypt with CTR mode and IV $Y + 1$
For $i \leftarrow 1$ to $m - 1$ do $C_i \leftarrow M_i \oplus E_K(Y + i)$
$C_m \leftarrow M_m \oplus E_K(Y + m)[1 : |M_m|]$; $C \leftarrow C_1 \cdots C_m$
// Use Carter-Wegman on G
$L \leftarrow E_K(0^n)$; $R \leftarrow G_L(A, C)$; $T \leftarrow \mathsf{Tag}(K, Y, R)$
Return $C \| T$

$\underline{\mathsf{Dec}(K, N, A, C \| T)}$
// $0 \le |C_m| < n$ and $|C_i| = n$ otherwise
$Y \leftarrow \mathsf{pad}(N)$; $C_1 \cdots C_m \leftarrow C$
// Decrypt with CTR mode and IV $Y + 1$
For $i \leftarrow 1$ to $m - 1$ do $M_i \leftarrow C_i \oplus E_K(Y + i)$
$M_m \leftarrow C_m \oplus E_K(Y + m)[1 : |C_m|]$; $M \leftarrow M_1 \cdots M_m$
// Use Carter-Wegman on G
$L \leftarrow E_K(0^n)$; $R \leftarrow G_L(A, C)$; $T' \leftarrow \mathsf{Tag}(K, Y, R)$
If $T' \neq T$ then return $\perp$ else return M

$\underline{\mathsf{Tag}(K, Y, R)}$ // CAU	$\underline{\mathsf{Tag}(K, Y, R)}$ // CAU-C1
$S \leftarrow E_K(Y) \oplus R$	$V \leftarrow Y \oplus R$; $S \leftarrow E_K(V) \oplus V$
Return $S[1 : \tau]$	Return $S[1 : \tau]$

Figure 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.

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 5.1 analyzes CMT-1 security of CAU-C1; the proof is in Appendix J. The result is in the standard model, although it relies on the multi-collision of the truncated Davies-Meyer that is justified in the ideal-cipher model via Proposition 4.3.

Theorem 5.1 *Let* $\mathsf{CAU\text{-}C1}[E, G, \tau]$ *be as above. Let* $s \ge 2$ *be an integer. Then for any adversary* $\mathcal{A}$*, we can construct an adversary* $\mathcal{B}$ *such that*

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

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 4.3, with $\tau = 96$, for any adversary $\mathcal{B}$

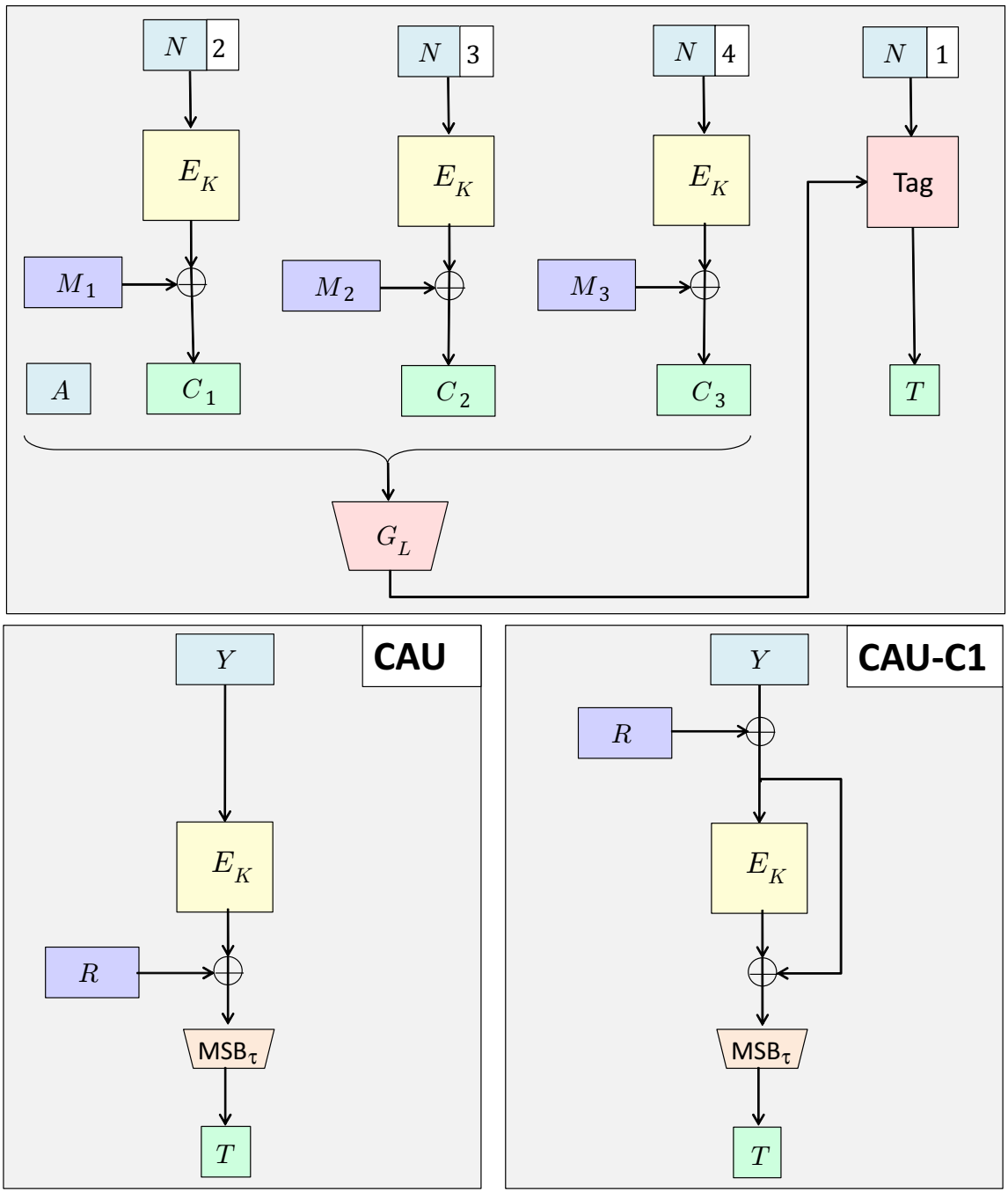

Figure 9: A pictorial comparison of the encryption schemes of CAU and CAU-C1. The two scheme have the same blueprint on the top panel. They however have different implementations for the internal procedure Tag, illustrated in the bottom panels. Here the trapezoid MSB_τ outputs the τ-bit prefix of the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

---

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 se-

curity, 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 $(A,M) \in \{0,1\}^* \times \{0,1\}^* \backslash (\varepsilon,\varepsilon)$,

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

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 Appendix B 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. See Appendix K for a proof.

Theorem 5.2 *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*

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

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 [27] 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 [48] in achieving misuse-resistance security, (ii) (a variant of) the CTR mode and (iii) an AXU hash function. We first recall the syntax and (multi-user) CPA security notion for IV-based encryption, which is needed to analyze the CTR variant.

IV-based Encryption. An IV-based symmetric encryption scheme SE consists of two algorithms, the randomized encryption algorithm SE.Enc and the deterministic decryption algorithm SE.Dec, and is associated with a key space $\mathcal{K}$ and an initialization-vector (IV) length n.

Game $\mathbf{G}^{\text{ind}}_{\mathsf{SE}}(\mathcal{A})$	NEW()	ENC(i, M)
$v \leftarrow 0$; $b \leftarrow\!\!\$\ \{0,1\}$	$v \leftarrow v + 1$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{NEW},\text{ENC}}$	$K_v \leftarrow\!\!\$\ \mathcal{K}$	$C_1 \leftarrow\!\!\$\ \mathsf{SE.Enc}(K_i, M)$; $C_0 \leftarrow\!\!\$\ \{0,1\}^{\lvert C_1\rvert}$
return b'		return C_b

Figure 10: Game defining the multi-user chosen-plaintext security of an IV-based encryption scheme SE.

The encryption algorithm $\mathsf{SE.Enc} : \mathcal{K} \times \{0,1\}^* \to \{0,1\}^*$ takes as input a secret key $K \in \mathcal{K}$ and a message M. It then samples $\mathsf{IV} \leftarrow\!\!\$\ \{0,1\}^n$, deterministically computes a ciphertext core C' from (K, M, IV), and then outputs $C \leftarrow \mathsf{IV} \| C'$. If we want to enforce SE.Enc to use a specific initialization vector IV, we will write $\mathsf{SE.Enc}(K, M; \mathsf{IV})$.

The decryption algorithm $\mathsf{SE.Dec} : \mathcal{K} \times \{0,1\}^* \to \{0,1\}^* \cup \{\perp\}$ takes as input a secret key $K \in \mathcal{K}$ and a ciphertext C, and returns either a message M or an error symbol $\perp$. For correctness, we require that if $C \leftarrow\!\!\$\ \mathsf{SE.Enc}(K, M)$ then $M \leftarrow \mathsf{SE.Dec}(K, C)$.

(MULTI-USER) CPA SECURITY FOR IV-BASED ENCRYPTION. Let SE be an IV-based symmetric encryption scheme with keyspace $\mathcal{K}$. For an adversary $\mathcal{A}$, define its advantage in breaking the multi-user chosen-plaintext security of SE as

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

where game $\mathbf{G}^{\text{ind}}_{\mathsf{SE}}(\mathcal{A})$ is defined in Fig. 10.

THE PRF GMAC$^+$. Like CAU, the scheme CAU-SIV is based on a c-AXU hash. As shown in [19], 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 [19] 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. 12 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. 11 for completeness.

The core of the scheme KD1 is the Truncated-Permutation construction $\mathsf{TP}[E, m] : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^m$ that $\mathsf{TP}[E, m](K, x) = E_K(x)[1 : m]$, where $m < n$ is a positive integer. Proposition 6.1 below gives a sharper bound on the PRF security of the Truncated-Permutation construction than what can be obtained via the Multi-user PRP/PRF Switching Lemma. The proof, which is in Appendix N, is based on the Chi-Squared technique of Dai, Hoang, and Tessaro [22]. If we ignore the PRP term then our bound is still meaningful even if $q > 2^n$.

Proposition 6.1 Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. For any adversary $\mathcal{A}$ of q queries in total and at most $B \leq 2^{n-1}$ queries per user, we can construct $\mathcal{B}$ of q queries and about

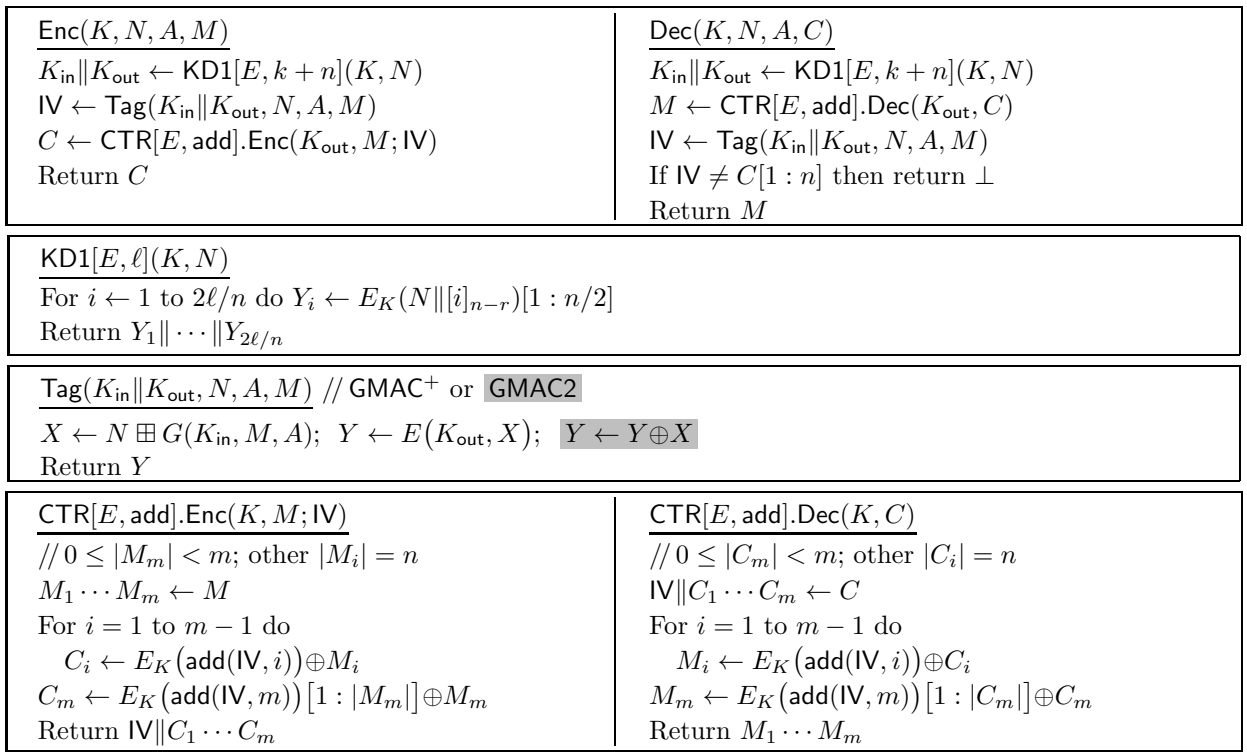

Figure 11: The schemes CAU-SIV and CAU-SIV-C1 whose encryption and decryption schemes are given in the top-left and top-right panels, respectively. Procedure Tag implements GMAC$^+$ (for CAU-SIV) or GMAC2 (for CAU-SIV-C1); the latter contains the highlighted code, but the former does not.

the same running time such that

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{TP}[E,m]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \min\{2\sqrt{nBq}\cdot 2^{m/2-n}, Bq/2^n\} \ .$$

In the single-user setting —namely when $B = q$— if we ignore the $\sqrt{n}$ factor and the PRP term then our bound degenerates to $\min\{B\cdot 2^{m/2-n}, B^2/2^n\}$, which matches a classical result of Stam [51]. This single-user bound is recently shown to be asymptotically tight by Gilboa and Gueron [28].

In the multi-user case, there is a matching attack, assuming that $B \ge 2^{m/2}$. In particular, the adversary $\mathcal{A}$ will attack $u = q/B$ users, each of B queries. For each user i, the adversary will run the single-user attack in [28] to obtain a guess bit b_i. Since this is a matching attack and $B \ge 2^{m/2}$, its advantage is $\Theta(B)\cdot 2^{m/2-n}$. From the definition of the PRF notion, if b is the challenge bit of game $\mathbf{G}^{\mathrm{prf}}_{\mathsf{TP}[E,m]}(\mathcal{A})$ then

$$\Pr[b_i = b] = \frac{1}{2} + \Theta(B)\cdot 2^{m/2-n} \ .$$

Finally, $\mathcal{A}$ will output the majority of the bits $b_1,\ldots,b_u$ as its guess. As $b_1,\ldots,b_u$ are independent and identically distributed, by using Chernoff's bound, one can show that the majority decision will amplify the bias $\Theta(B)\cdot 2^{m/2-n}$ by a factor of $\Theta(\sqrt{u}) = \Theta(\sqrt{Bq})$. In other words,

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{TP}[E,m]}(\mathcal{A}) = \Theta(\sqrt{Bq})\cdot\Theta(B)\cdot 2^{m/2-n} = \Theta(\sqrt{Bq})\cdot 2^{m/2-n} \ .$$

Security of KD1. From Proposition 6.1, it is straightforward that KD1 is also a good (multi-user) PRF; the formal result is stated below for completeness.

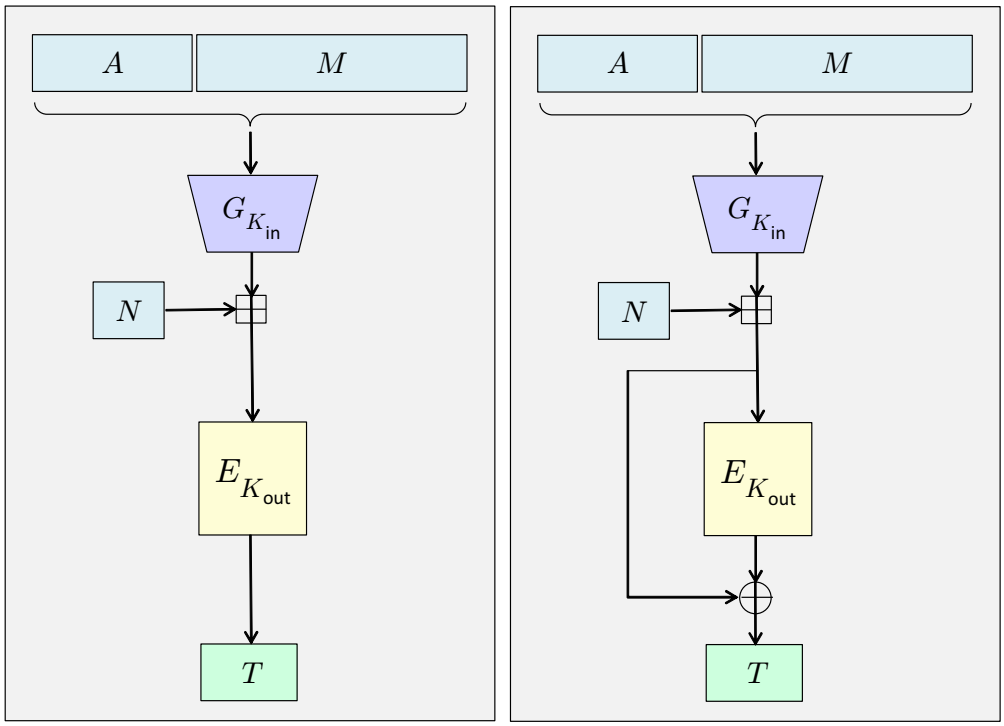

Figure 12: The GMAC^+ construction (left) and its variant GMAC2 (right).

Lemma 6.2 *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^m$ be a blockcipher, and let ℓ be a multiple of n. Define $\mathsf{KD1}[E,\ell]$ as above, and let $t = \ell/n$. Then for any adversary $\mathcal{A}$ of q queries in total and at most B queries per user, we can construct an adversary $\mathcal{B}$ of about the same running time and $2qt$ queries such that*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{KD1}[E,\ell]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \min\{2t \cdot \sqrt{nBq} \cdot 2^{-3n/4}, 4Bqt^2/2^n\} \ .$$

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. 11. 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 (multi-user) chosen-plaintext security of this CTR variant is already analyzed in the ideal-cipher model by Bose, Hoang, and Tessaro [19]. The following result shows the standard-model counterpart; the proof is in Appendix O.

Proposition 6.3 Let CTR be as above, and let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be the underlying blockcipher. For an adversary $\mathcal{A}$ whose queries consist of totally σ blocks with at most B blocks per user, we can construct an adversary $\mathcal{B}$ of about the same running time that makes at most σ queries such that

$$\mathbf{Adv}^{\mathrm{ind}}_{\mathsf{CTR}[E,\mathsf{add}]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{3\sigma B}{2^n} \ .$$

THE SCHEME CAU-SIV. The scheme CAU-SIV$[E, G, \mathsf{add}]$ is described in Fig. 11. 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 [48] 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 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. 12 for a side-by-side pictorial comparison of GMAC$^+$ and GMAC2. The code of CAU-SIV-C1 is given in Fig. 11.

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 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.

SECURITY OF GMAC2. The following result shows that GMAC2 is a good (multi-user) PRF; see Appendix P for a proof.

Proposition 6.4 Let GMAC2$[E, G]$ be as above, building on top of a c-AXU hash function G and a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. For an adversary $\mathcal{A}$ that makes at most q queries in total with at most B blocks per user, we can construct an adversary $\mathcal{B}$ of about the same running time that makes at most q queries such that

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{GMAC2}[E,G]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{(2c+1)qB}{2^n} \ .$$

COMMITTING SECURITY OF CAU-SIV-C1. Theorem 6.5 below confirms that the extra xor indeed hardens CAU-SIV-C1, ensuring CMT-1 security. The proof is in Appendix Q. 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 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 DM$[E]$ to commit K_{out}. Thus from Proposition 4.5, the CMT-1 security of CAU-SIV-C1 is reduced to the multi-collision resistance of ITP$[E, r, n]$ and DM$[E]$ that are justified in Propositions 4.3 and 4.4.

Theorem 6.5 *Let* SE = CAU-SIV-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*

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

Figure 13: Game defining the binding security of a committing PRF F.

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*

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

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 Appendix R.

Theorem 6.6 *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*

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

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 CAU-SIV-C1, yields a CMT-4 and MRAE scheme that we call CAU-SIV-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

$$\mathbf{Adv}^{\text{bind}}_{\mathsf{F},s}(\mathcal{A}) = \Pr[\mathbf{G}^{\text{bind}}_{\mathsf{F},s}(\mathcal{A})] \ ,$$

where game $\mathbf{G}^{\text{bind}}_{\mathsf{F},s}(\mathcal{A})$ is defined in Fig. 13. 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.

$\underline{\mathsf{CX}[E](K, M)}$
$a \leftarrow \lceil \ell/n \rceil$; $b \leftarrow \lceil \lambda/n \rceil$
For $i \leftarrow 1$ to $a + b$ do $X_i \leftarrow \mathsf{pad}(M, i)$; $V_i \leftarrow E_K(X_i)$
$V_1 \leftarrow V_1 \oplus X_1$
$P \leftarrow (V_1 \cdots V_a)[1 : \ell]$; $L \leftarrow (V_{a+1} \cdots V_{a+b})[1 : \lambda]$
Return (P, L)

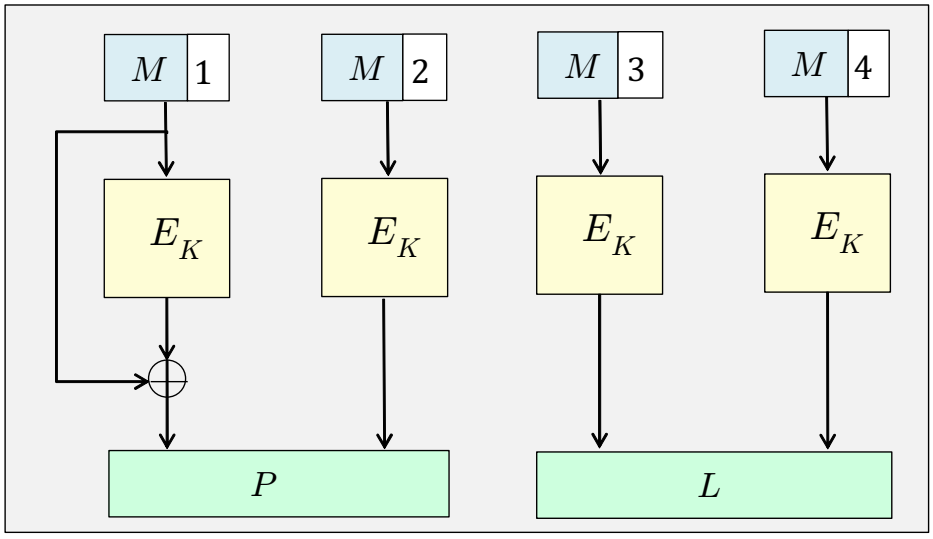

Figure 14: The committing PRF scheme $\mathsf{CX}[E, \mathsf{pad}]$, illustrated for the case $\ell = \lambda = 2n$ and $\mathsf{pad}(M, i)$ is the concatenation of M and an $(n - m)$-bit encoding of i.

For $s = 2$, our notion of committing PRF can be viewed as a PRF counterpart of the notion of *right collision-resistant PRG* in [26]. We however will give practical instantiations via a blockcipher whereas the construction in [26] 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. 14.

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 4.3, since the first block of P is obtained from the Davies-Meyer construction $\mathsf{DM}[E]$.

Proposition 7.1 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 at most dq queries such that

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

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

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

Figure 15: The encryption (left) and decryption (right) schemes of the resulting AE scheme under the UtC transform.

and resources such that

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

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. 15. 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 Appendix S.

Theorem 7.2 *Let* SE *and* F *be as above. Let* $s \geq 2$ *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*

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

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*

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

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 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.

For unique-nonce security, using UtC actually *improves* the concrete security of SE, because UtC uses an independent subkey for SE per (user, nonce). The same mechanism was used in AES-GCM-SIV [31] to improve the concrete security of GCM-SIV [30]. The UtC transform can add CMT-1 security to an existing AE scheme SE without being intrusive. This is an important benefit if SE is a widely deployed scheme like GCM. However, a ciphertext in UtC is always, say 128-bit

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

Figure 16: The encryption (left) and decryption (right) algorithms of the scheme given by the RtC transform.

longer than the corresponding ciphertext in SE; this bandwidth cost can be significant for some applications.

We note that UtC does not preserve misuse-resistance security. In particular, the commitment P depends only on the nonce N and the master key K. Thus even in the single-user setting, an adversary can trivially break the misuse-resistance security of $\mathsf{UtC}[\mathsf{F},\mathsf{SE}]$ by making two encryption queries of the same nonce but different messages, and checking if the two ciphertexts have the same ℓ-bit prefix.

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 4.1.) The scheme $\mathsf{RtC}[\mathsf{F},\mathsf{SE},H]$ is shown in Fig. 16. 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 4.5, 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 Appendix T.

Theorem 7.3 *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 $\mathcal{A}_1$.*

DISCUSSION. The transform RtC also preserves unique-nonce security, but in this setting, it is inferior to the UtC transform. For misuse-resistance security, using RtC also *improves* the concrete security of SE, because RtC uses an independent subkey for SE per (user, nonce). The same mechanism was used in AES-GCM-SIV [31] to improve the concrete security of GCM-SIV [30].

While the RtC transform can add key-committing security to an existing AE scheme SE without being intrusive, its ciphertext is always, say 128-bit longer than the corresponding ciphertext in SE. This bandwidth cost can be significant for some applications.

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.

References

[1] AWS Encryption SDK 2.0. `https://docs.aws.amazon.com/encryption-sdk/latest/developer-guide/in` 2020. 27

[2] The Sodium cryptography library (Libsodium). `https://libsodium.gitbook.io/doc`, 2021. 29

[3] M. Abdalla, M. Bellare, and G. Neven. Robust encryption. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 480–497. Springer, Heidelberg, Feb. 2010. 4, 5, 11

[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. 4, 5, 6, 8, 11, 15, 17, 27

[5] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In N. Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 1–15. Springer, Heidelberg, Aug. 1996. 6

[6] M. Bellare, R. Canetti, and H. Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th FOCS*, pages 514–523. IEEE Computer Society Press, Oct. 1996. 10

[7] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *38th FOCS*, pages 394–403. IEEE Computer Society Press, Oct. 1997. 4

[8] 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*, pages 491–506. Springer, Heidelberg, May 2003. 8

[9] 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*, pages 531–545. Springer, Heidelberg, Dec. 2000. 4

[10] 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*, pages 235–265. Springer, Heidelberg, Aug. 2019. 14

[11] 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*, pages 409–426. Springer, Heidelberg, May / June 2006. 14, 37, 54, 55, 65

[12] 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*, pages 247–276. Springer, Heidelberg, Aug. 2016. 4, 9

[13] 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*, pages 729–757. Springer, Heidelberg, May 2016. 17

[14] D. Bernstein. Chacha, a variant of salsa20. In *Workshop record of SASC*, volume 8, pages 3–5, 2008. 4

[15] 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. 4

[16] D. J. Bernstein. The poly1305-AES message-authentication code. In H. Gilbert and H. Handschuh, editors, *FSE 2005*, volume 3557 of *LNCS*, pages 32–49. Springer, Heidelberg, Feb. 2005. 4

[17] 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*, pages 320–335. Springer, Heidelberg, Aug. 2002. 15, 16

[18] 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*, pages 506–522. Springer, Heidelberg, May 2004. 5

[19] 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*, pages 468–499. Springer, Heidelberg, Apr. / May 2018. 4, 14, 21, 23

[20] S. Chen and J. P. Steinberger. Tight security bounds for key-alternating ciphers. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 327–350. Springer, Heidelberg, May 2014. 44

[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*, pages 430–448. Springer, Heidelberg, Aug. 2005. 14

[22] W. Dai, V. T. Hoang, and S. Tessaro. Information-theoretic indistinguishability via the chi-squared method. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 497–523. Springer, Heidelberg, Aug. 2017. 21, 50

[23] 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*, pages 155–186. Springer, Heidelberg, Aug. 2018. 8, 15, 17

[24] M. Dworkin. Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D, November 2007. 4, 6

[25] 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*, pages 352–368. Springer, Heidelberg, Feb. / Mar. 2013. 5

[26] 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. 4, 5, 7, 8, 11, 17, 26

[27] N. Ferguson. Authentication weaknesses in GCM. Manuscript, available in NIST webpage, 2005. 20

[28] S. Gilboa and S. Gueron. The advantage of truncated permutations. *Discrete Applied Mathematics*, 294:214–223, 2021. 22

[29] 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*, pages 66–97. Springer, Heidelberg, Aug. 2017. 4, 5, 6, 8, 17

[30] S. Gueron and Y. Lindell. GCM-SIV: Full nonce misuse-resistant authenticated encryption at under one cycle per byte. In I. Ray, N. Li, and C. Kruegel, editors, *ACM CCS 2015*, pages 109–119. ACM Press, Oct. 2015. 27, 29

[31] 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*, pages 1019–1036. ACM Press, Oct. / Nov. 2017. 4, 7, 27, 29

[32] V. T. Hoang and S. Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 3–32. Springer, Heidelberg, Aug. 2016. 9

[33] 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*, pages 1429–1440. ACM Press, Oct. 2018. 4

[34] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. 52

[35] 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*, pages 456–486. Springer, Heidelberg, Apr. / May 2018. 15, 17

[36] B. Kaliski. PKCS #5: Password-Based Cryptography Specification Version 2.0. RFC 2898, Sep. 2000. `https://datatracker.ietf.org/doc/html/rfc2898`. 5

[37] J. Katz and M. Yung. Unforgeable encryption and chosen ciphertext secure modes of operation. In B. Schneier, editor, *FSE 2000*, volume 1978 of *LNCS*, pages 284–299. Springer, Heidelberg, Apr. 2001. 4

[38] M. Lambæk. Breaking and fixing private set intersection protocols. Cryptology ePrint Archive, Report 2016/665, 2016. `https://eprint.iacr.org/2016/665`. 5

[39] J. Len, P. Grubbs, and T. Ristenpart. Partitioning oracle attacks. In M. Bailey and R. Greenstadt, editors, *30th USENIX Security Symposium*. USENIX Association, 2021. 4, 5, 6, 7, 8, 12

[40] 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*, pages 575–605. Springer, Heidelberg, Dec. 2017. 4, 14

[41] 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*, pages 343–355. Springer, Heidelberg, Dec. 2004. 4, 6

[42] 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*, pages 209–223. Springer, Heidelberg, Aug. 2015. 10

[43] C. Namprempre, P. Rogaway, and T. Shrimpton. Reconsidering generic composition. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 257–274. Springer, Heidelberg, May 2014. 4, 8

[44] R. Pagh and F. F. Rodler. Cuckoo hashing. *Journal of Algorithms*, 51(2):122–144, 2004. 35

[45] J. Patarin. The "coefficients H" technique (invited talk). In R. M. Avanzi, L. Keliher, and F. Sica, editors, *SAC 2008*, volume 5381 of *LNCS*, pages 328–345. Springer, Heidelberg, Aug. 2009. 44

[46] P. Rogaway. Authenticated-encryption with associated-data. In V. Atluri, editor, *ACM CCS 2002*, pages 98–107. ACM Press, Nov. 2002. 4

[47] 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*, pages 196–205. ACM Press, Nov. 2001. 4, 5

[48] P. Rogaway and T. Shrimpton. A provable-security treatment of the key-wrap problem. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 373–390. Springer, Heidelberg, May / June 2006. 4, 20, 24

[49] 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`. 4, 6

[50] J. Salowey, A. Choudury, and D. A. McGrew. AES Galois Counter Mode (GCM) cipher suites for TLS. RFC 5288, August 2008. 4, 7

[51] A. J. Stam. Distance between sampling with and without replacement. *Statistica Neerlandica*, 32(2):81–91, 1978. 22

A Relations Among Committing Notions

The implications indicated by the leftward arrows are trivial. Below, we will justify the remaining ones.

CMTD-3 $\rightarrow$ CMTD-4. Suppose $\mathcal{A}$ is an adversary that violates CMTD-4, outputting $(C, (K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2))$ such that for each $i \in \{1, 2\}$, we have $M_i = \mathsf{SE.Dec}(K_i, N_i, A_i, C)$, but $(K_1, N_1, A_1, M_1) \neq (K_2, N_2, A_2, M_2)$. If $(K_1, N_1, A_1) \neq (K_2, N_2, A_2)$ then $\mathcal{A}$ violates CMTD-3 and we are done, so suppose $(K_1, N_1, A_1) = (K_2, N_2, A_2)$. But since $\mathsf{SE.Dec}$ is deterministic it must be that $\mathsf{SE.Dec}(K_1, N_1, A_1, C) = \mathsf{SE.Dec}(K_2, N_2, A_2, C)$, meaning $M_1 = M_2$ and thus $(K_1, N_1, A_1, M_1) = (K_2, N_2, A_2, M_2)$, which contradicts our assumption that these two tuples are distinct.

CMT-3 $\rightarrow$ CMT-4. Suppose $\mathcal{A}$ is an adversary that violates CMT-4, outputting $((K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2))$ such that $C_1 = C_2$ but $(K_1, N_1, A_1, M_1) \neq (K_2, N_2, A_2, M_2)$, where $C_i \leftarrow \mathsf{SE.Enc}(K_i, N_i, A_i, M_i)$ for each $i \in \{1, 2\}$. If $(K_1, N_1, A_1) \neq (K_2, N_2, A_2)$ then $\mathcal{A}$ violates CMT-3 and we are done, so suppose $(K_1, N_1, A_1) = (K_2, N_2, A_2)$. From the correctness requirement, $M_i = \mathsf{SE.Dec}(K_i, N_i, A_i, C_i)$ for each $i \in \{1, 2\}$. Since $C_1 = C_2$ and $(K_1, N_1, A_1) = (K_2, N_2, A_2)$, the determinism of $\mathsf{SE.Dec}$ implies that $M_1 = M_2$, and thence $(K_1, N_1, A_1, M_1) = (K_2, N_2, A_2, M_2)$, which contradicts our assumption that these two tuples are distinct.

CMTD-$\ell \rightarrow$ CMT-ℓ. Fix $\ell \in \{1, 3, 4\}$. Let $\mathcal{A}_{\mathrm{e}}$ be an adversary attacking the CMT-ℓ security of SE. We build an adversary $\mathcal{A}_{\mathrm{d}}$ attacking the CMTD-ℓ security of SE as follows. Adversary $\mathcal{A}_{\mathrm{d}}$ runs $((K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2)) \leftarrow\!\!\$\ \mathcal{A}_{\mathrm{e}}$, and then sets $C_1 \leftarrow \mathsf{SE.Enc}(K_1, N_1, A_1, M_1)$ and returns $(C_1, (K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2))$.

To analyze the advantage of the adversaries, without loss of generality, assume that $\mathsf{WiC}_\ell(K_1, N_1, A_1, M_1) \neq \mathsf{WiC}_\ell(K_2, N_2, A_2, M_2)$. Let $C_2 \leftarrow \mathsf{SE.Enc}(K_2, N_2, A_2, M_2)$. The correctness requirement implies that for each $i \in \{1, 2\}$, we have $M_i = \mathsf{SE.Dec}(K_i, N_i, A_i, C_i)$. If $\mathcal{A}_{\mathrm{e}}$ wins then $C_1 = C_2$, meaning that $M_2 = \mathsf{SE.Dec}(K_2, N_2, A_2, C_1)$, and thus $\mathcal{A}_{\mathrm{d}}$ also wins. Hence

$$\mathbf{Adv}^{\text{cmt-}\ell}_{\mathsf{SE}}(\mathcal{A}_{\mathrm{e}}) \leq \mathbf{Adv}^{\text{cmtd-}\ell}_{\mathsf{SE}}(\mathcal{A}_{\mathrm{d}}) \ .$$

CMT-$\ell \rightarrow$ CMTD-ℓ FOR TIDY SCHEMES. Fix $\ell \in \{1, 3, 4\}$. Suppose that SE is tidy. Let $\mathcal{A}_{\mathrm{d}}$ be an adversary attacking the CMTD-ℓ security of SE. We build an adversary $\mathcal{A}_{\mathrm{e}}$ attacking the CMT-ℓ security of SE as follows. Adversary $\mathcal{A}_{\mathrm{e}}$ runs $(C, (K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2)) \leftarrow\!\!\$\ \mathcal{A}_{\mathrm{d}}$, and returns $((K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2))$.

To analyze the advantage of the adversaries, without loss of generality, assume that $\mathsf{WiC}_\ell(K_1, N_1, A_1, M_1) \neq \mathsf{WiC}_\ell(K_2, N_2, A_2, M_2)$. If $\mathcal{A}_{\mathrm{d}}$ wins then $M_i = \mathsf{SE.Dec}(K_i, N_i, A_i, C)$ for each $i \in \{1, 2\}$. From the tidiness of SE, this means that $C = \mathsf{SE.Enc}(K_i, N_i, A_i, M_i)$, and thus $\mathcal{A}_{\mathrm{e}}$ also wins. Hence Hence

$$\mathbf{Adv}^{\text{cmtd-}\ell}_{\mathsf{SE}}(\mathcal{A}_{\mathrm{d}}) \leq \mathbf{Adv}^{\text{cmt-}\ell}_{\mathsf{SE}}(\mathcal{A}_{\mathrm{e}}) \ .$$

$\underline{\mathsf{GHASH}[\mathbb{F}](K, A, M)}$
$t \leftarrow n/2;\ X \leftarrow (A0^*)\|(M0^*)\|\big[|A|\big]_t\|\big[|M|\big]_t$
$X_1 \cdots X_m \leftarrow X;\ Y \leftarrow 0^n$ // $|X_i| = n$
For $i \leftarrow 1$ to m do $Y \leftarrow (Y \oplus X_i) \bullet K$
Return Y

Figure 17: The GHASH function.

B GHASH As a Weakly Regular Hash

Description of GHASH. Let $n \geq 2$ be an even integer. Let $\mathbb{F}$ be be a finite field of 2^n elements, meaning that we can interpret a string in $\{0,1\}^n$ as an element of $\mathbb{F}$ and vice versa. Assume that 0^n is the zero element in $\mathbb{F}$, and the addition in $\mathbb{F}$ is the same as $\oplus$ in $\{0,1\}^n$. Let $\bullet$ be the finite-field multiplication in $\mathbb{F}$. For a string x, let $x0^*$ denote $x0^p$, where $p \geq 0$ is the smallest integer such that $|x| + p$ is a multiple of n. For a number $i \geq 0$, we write $[i]_t$ to denote a t-bit representation of i. The hash function $\mathsf{GHASH}[\mathbb{F}] : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ is defined in Fig. 17. Note that $\mathsf{GHASH}[\mathbb{F}](K, \varepsilon, \varepsilon) = 0^n$ for every K.

Weak regularity of GHASH. We now show that GHASH is weakly 1.5-regular. Fix $(A, M) \neq (\varepsilon, \varepsilon)$ and $Y \in \{0,1\}^n$. Let

$$X \leftarrow (A0^*)\|(M0^*)\|\big[|A|\big]_t\|\big[|M|\big]_t \ ,$$

where $t = n/2$. Parse X as $X_1 \cdots X_m$, where each $|X_i| = n$. Note that

$$m = |X|/n \leq |A|_n + |M|_n + 1 \leq 1.5(|A|_n + |M|_n) \ ,$$

since $|A|_n, |M|_n \geq 1$. Let

$$f(x) = (X_1 \bullet x^m) \oplus (X_2 \bullet x^{m-1}) \oplus \cdots \oplus (X_m \bullet x) \oplus Y \ .$$

Note that f is a polynomial of degree at most m, and since $(A, M) \neq (\varepsilon, \varepsilon)$, f is non-zero. Hence f has at most m roots. If we pick $K \xleftarrow{\$} \{0,1\}^n$ then the chance that K is one of those m roots is at most $m/2^n \leq 1.5(|M|_n + |A|_n)/2^n$. Hence

$$\begin{aligned} \Pr_{K \xleftarrow{\$} \{0,1\}^n}[\mathsf{GHASH}[\mathbb{F}](K, A, M) = Y] &= \Pr_{K \xleftarrow{\$} \{0,1\}^n}[f(K) = 0^n] \\ &\leq \frac{1.5(|M|_n + |A|_n)}{2^n} \end{aligned}$$

and thus GHASH is weakly 1.5-regular.

C A Lower Bound on Multi-collision Resistance

To understand what we can expect for the multi-collision resistance of a hash function, in Proposition C.1 below, we give an attack on a hash function $H : \mathsf{Dom} \to \mathsf{Rng}$ that we will model as a random oracle.

Proposition C.1 Let $H : \mathsf{Dom} \to \mathsf{Rng}$ be a hash function that we model as a random oracle. Let $N = |\mathsf{Rng}|$. Let p be an integer such that $2s \leq p \leq N/6$ and $\binom{p}{s} \leq N^{s-1}$. Then we can construct an adversary $\mathcal{A}$ of $O(p \log(p))$ time such that

$$\mathbf{Adv}^{\mathrm{coll}}_{H,s}(\mathcal{A}) \geq \binom{p}{s} \cdot \frac{N^{-(s-1)}}{4} \ .$$

Proof: The adversary $\mathcal{A}$ is as follows. It picks arbitrary, distinct $M_1, \ldots, M_p \in \mathsf{Dom}$ and then searches for $X_1, \ldots, X_s \in \{M_1, \ldots, M_p\}$ such that $(X_1, \ldots, X_s)$ is an s-way multi-collision for H. To implement this in $O(p \log(p))$ time, $\mathcal{A}$ computes $V_i \leftarrow H(M_i)$ for every $i \leq p$, sorts the list $(V_1, \ldots, V_p)$, and then looks for a sequence of s consecutive points in the sorted list that are the same. (The running time of the adversary can be improved to $O(p)$ at the cost of a failure rate $1/p$ by using cuckoo hashing [44].)

To analyze the adversary's advantage, we need the following inequality.

Lemma C.2 (Bonferroni's inequality) *For any events $A_1, \ldots, A_n$,*

$$\Pr[A_1 \cup \cdots \cup A_n] \geq \Big(\sum_{i=1}^{n} \Pr[A_i]\Big) - \Big(\sum_{1 \leq i < j \leq n} \Pr[A_i \cap A_j]\Big) \ .$$

Back to the analysis of the adversary's advantage, let

$$\Delta = \binom{p}{s} \cdot N^{-s} \leq \frac{1}{N} \ .$$

We claim that

$$\mathbf{Adv}^{\mathrm{coll}}_{H,s}(\mathcal{A}) \geq \frac{N \cdot \Delta}{4} = \binom{p}{s} \cdot \frac{N^{-(s-1)}}{4} \ .$$

To justify this claim, we view each $H(M_i)$ as throwing a ball into N possible bins in a uniformly random way, and the p throws are independent. For each $i \in \{1, \ldots, N\}$, let Bad_i be the event that there are at least s balls in the i-th bin. Note that

$$\begin{aligned} \Pr[\mathrm{Bad}_i] &\geq \binom{p}{s} \frac{1}{N^s} \Big(1 - \frac{1}{N}\Big)^{p-s} \\ &= \Delta \cdot \Big(1 - \frac{1}{N}\Big)^{p-s} \geq \Delta \cdot \Big(1 - \frac{1}{N}\Big)^{N/6} \geq \frac{\Delta}{4^{1/6}} \geq \frac{3\Delta}{4} \end{aligned}$$

for every $i \in \{1, \ldots, N\}$, where the third last inequality is due to the hypothesis that $p \leq N/6$, and the second last inequality is due to the fact that $(1 - 1/x)^x \geq 1/4$ for every integer $x \geq 2$. Moreover, for every $1 \leq i < j \leq N$,

$$\Pr[\mathrm{Bad}_i \cap \mathrm{Bad}_j] \leq \binom{p}{s} \binom{p-s}{s} N^{-2s} \leq \Delta^2 \ .$$

By Bonferroni inequality,

$$\begin{aligned} \mathbf{Adv}^{\mathrm{coll}}_{H,s}(\mathcal{A}) &= \Pr\Big[\bigcup_{i=1}^{N} \mathrm{Bad}_i\Big] \\ &\geq \Big(\sum_{i=1}^{N} \Pr[\mathrm{Bad}_i]\Big) - \Big(\sum_{1 \leq i < j \leq N} \Pr[\mathrm{Bad}_i \cap \mathrm{Bad}_j]\Big) \\ &\geq \frac{3\Delta \cdot N}{4} - \frac{N(N-1)}{2}\Delta^2 \geq \Delta \cdot N \cdot \Big(\frac{3}{4} - \frac{N \cdot \Delta}{2}\Big) \geq \frac{\Delta \cdot N}{4} \ . \end{aligned}$$

This concludes the proof. ∎

Game $\mathbf{G}_1(\mathcal{A})$	$\underline{\text{EVAL}(i, M)}$
$v \leftarrow 0$; $b' \leftarrow \mathcal{A}^{\text{NEW},\text{EVAL}}$; return $(b' = 1)$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$\underline{\text{NEW}()}$	$C \leftarrow E(K_i, M)$
$v \leftarrow v + 1$; $K_v \xleftarrow{\$} \{0,1\}^k$	return C

Game $\mathbf{G}_2(\mathcal{A})$	$\underline{\text{EVAL}(i, M)}$
$v \leftarrow 0$; $b' \leftarrow \mathcal{A}^{\text{NEW},\text{EVAL}}$; return $(b' = 1)$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$\underline{\text{NEW}()}$	$C \leftarrow \pi_i(M)$
$v \leftarrow v + 1$; $\pi_v \xleftarrow{\$} \mathsf{Perm}(\{0,1\}^n)$	return C

Games $\mathbf{G}_3(\mathcal{A})$, $\mathbf{G}_4(\mathcal{A})$	$\underline{\text{EVAL}(i, M)}$
$v \leftarrow 0$; $b' \leftarrow \mathcal{A}^{\text{NEW},\text{EVAL}}$; return $(b' = 1)$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$\underline{\text{NEW}()}$	$C \xleftarrow{\$} \{0,1\}^n$
$v \leftarrow v + 1$; $S_v \xleftarrow{\$} \emptyset$	If $C \in S_i$ then
	$\quad \mathsf{bad} \leftarrow \mathsf{true}$; $C \xleftarrow{\$} \{0,1\}^n \backslash \{C\}$
	$S_i \leftarrow S_i \cup \{C\}$; return C

Game $\mathbf{G}_5(\mathcal{A})$	$\underline{\text{EVAL}(i, M)}$
$v \leftarrow 0$; $b' \leftarrow \mathcal{A}^{\text{NEW},\text{EVAL}}$; return $(b' = 1)$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$\underline{\text{NEW}()}$	$C \xleftarrow{\$} \{0,1\}^n$
$v \leftarrow v + 1$	return C

Game $\mathbf{G}_6(\mathcal{A})$	$\underline{\text{EVAL}(i, M)}$
$v \leftarrow 0$; $b' \leftarrow \mathcal{A}^{\text{NEW},\text{EVAL}}$; return $(b' = 1)$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$\underline{\text{NEW}()}$	$C \leftarrow f_i(M)$
$v \leftarrow v + 1$; $f_v \xleftarrow{\$} \mathsf{Func}(\{0,1\}^n, \{0,1\}^n)$	return C

Figure 18: Games in the proof of Lemma 4.1. Game $\mathbf{G}_3$ contains the corresponding highlighted code but game $\mathbf{G}_4$ does not.

D Proof of Lemma 4.1

Consider games $\mathbf{G}_1$–$\mathbf{G}_6$ in Fig. 18. By definition,

$$\mathbf{Adv}^{\text{prf}}_E(\mathcal{A}) = \Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_6(\mathcal{A})] \ ,$$

whereas

$$\mathbf{Adv}^{\text{prp}}_E(\mathcal{A}) = \Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_2(\mathcal{A})] \quad .$$

To justify the claim of this lemma, it suffices to show that

$$\Pr[\mathbf{G}_2(\mathcal{A})] - \Pr[\mathbf{G}_6(\mathcal{A})] \leq \frac{Bq}{2^n} \ .$$

In other words, the advantage of $\mathcal{A}$ in distinguishing the games $\mathbf{G}_2$ and $\mathbf{G}_6$ is at most $Bq/2^n$. We shall prove this for any (even computationally unbounded) adversary that makes totally at most q queries to EVAL, with at most B queries per user. As we consider even computationally unbounded adversaries, without loss of generality, assume that $\mathcal{A}$ is deterministic. Assume further that the adversary will not repeat prior queries to EVAL.

We now explain the game chain from $\mathbf{G}_2$ to $\mathbf{G}_6$. Game $\mathbf{G}_3$ is the same as $\mathbf{G}_2$, but instead of eagerly sampling a permutation $\pi_i \leftarrow\!\!\$\ \mathsf{Perm}(\{0,1\}^n)$ for each user i, we lazily implement π_i by maintaining a set S_i of the current defined outputs of π_i, and each call $\pi_i(M)$ is implemented via a uniform sampling from $\{0,1\}^n \backslash S_i$. Hence

$$\Pr[\mathbf{G}_2(\mathcal{A})] = \Pr[\mathbf{G}_3(\mathcal{A})] \ .$$

In game $\mathbf{G}_4$, in each call $\pi_i(M)$, we instead sample the answer $C \leftarrow\!\!\$\ \{0,1\}^n$. If $C \in S_i$ then this sets the flag bad to be true, but unlike game $\mathbf{G}_3$, we will *not* resample C from $\{0,1\}^n \backslash S_i$. From the Fundamental Lemma of Game Playing [11], the two games $\mathbf{G}_3$ and $\mathbf{G}_4$ are identical-until-bad, and thus

$$\Pr[\mathbf{G}_3(\mathcal{A})] - \Pr[\mathbf{G}_4(\mathcal{A})] \leq \Pr[\mathbf{G}_4(\mathcal{A}) \text{ sets } \mathsf{bad}] \ .$$

We now bound the chance that game $\mathbf{G}_4$ sets bad to be true. For each query $\textsc{Eval}(i, M)$, the current size of S_i is at most B, since there are at most B prior queries to user i. Thus if we pick $C \leftarrow\!\!\$\ \{0,1\}^n$ independent of S_i, the chance that $C \in S_i$, which is also the chance that this query triggers bad to be true, is at most $B/2^n$. Summing this over at most q queries, by the Union Bound,

$$\Pr[\mathbf{G}_4(\mathcal{A}) \text{ sets } \mathsf{bad}] \leq \frac{Bq}{2^n} \ ,$$

and thus

$$\Pr[\mathbf{G}_3(\mathcal{A})] - \Pr[\mathbf{G}_4(\mathcal{A})] \leq \frac{Bq}{2^n} \ .$$

Back to the game chain, game $\mathbf{G}_5$ is exactly the same as $\mathbf{G}_4$; we only simplify the code. Hence

$$\Pr[\mathbf{G}_4(\mathcal{A})] = \Pr[\mathbf{G}_5(\mathcal{A})] \ .$$

Finally, game $\mathbf{G}_6$ is the same as $\mathbf{G}_5$, but we now eagerly sample a function $f_i \leftarrow\!\!\$\ \mathsf{Func}(\{0,1\}^n, \{0,1\}^n)$ for each user i instead of lazily implementing it. Therefore,

$$\Pr[\mathbf{G}_5(\mathcal{A})] = \Pr[\mathbf{G}_6(\mathcal{A})] \ .$$

Summing up,

$$\Pr[\mathbf{G}_2(\mathcal{A})] - \Pr[\mathbf{G}_6(\mathcal{A})] = \sum_{i=2}^{5} \Pr[\mathbf{G}_i(\mathcal{A})] - \Pr[\mathbf{G}_{i+1}(\mathcal{A})] \leq \frac{Bq}{2^n}$$

as claimed.

E Proof of Proposition 4.2

The adversary $\mathcal{B}$ runs $\mathcal{A}$, and acts as a proxy for $\textsc{New}$ or $\textsc{Enc}$ queries of $\mathcal{A}$. For each verification query of $\mathcal{A}$, however, $\mathcal{B}$ stores that in a list L, and simply returns false to $\mathcal{A}$. When $\mathcal{A}$ terminates with its guess b', for each query (i, N, A, C) in L, if there is no prior encryption query $C \leftarrow \textsc{Enc}(i, N, A, M)$ then $\mathcal{B}$ will query $\textsc{Vf}(i, N, A, C)$, otherwise it will terminate and return 1. If one of those verification queries results in a true answer then $\mathcal{B}$ will return 1, otherwise it returns b'. Note that $\mathcal{B}$ is orderly. Moreover, if $\mathcal{A}$ is nonce-respecting then so is $\mathcal{B}$.

In the real world, if some verification query could result in a true-answer then $\mathcal{B}$ will answer 1 anyway (even if it can't make this query due to the restriction). If all verification queries are destined to give false answers then $\mathcal{B}$ correctly simulates $\mathbf{G}^{\text{real}}_{\mathsf{SE}}(\mathcal{A})$, and it either gives the same answer as $\mathcal{A}$, or returns 1. Hence

$$\Pr[\mathbf{G}^{\text{real}}_{\mathsf{SE}}(\mathcal{B})] \geq \Pr[\mathbf{G}^{\text{real}}_{\mathsf{SE}}(\mathcal{A})] \ .$$

Let Bad be the event that in the ideal world, there is a verification query $\textsc{Vf}(i, N, A, C)$ and then

later there is an encryption query $C \leftarrow \textsc{Enc}(i, N, A, M)$. If Bad doesn't happen then $\mathcal{B}$ correctly simulates $\mathbf{G}^{\text{rand}}_{\mathsf{SE}}(\mathcal{A})$ and has the same answer as $\mathcal{A}$, and thus

$$\Pr[\mathbf{G}^{\text{rand}}_{\mathsf{SE}}(\mathcal{B})] \le \Pr[\mathbf{G}^{\text{rand}}_{\mathsf{SE}}(\mathcal{A})] + \Pr[\text{Bad}] \ .$$

To bound Pr[Bad], note that for each verification query $\textsc{Vf}(i, N, A, C)$, it can be targeted by at most

$$\sum_{j=0}^{s} 2^j \le 2^{s+1}$$

encryption queries where $s = |C| - \tau$ is the maximum length of the corresponding message. However, the chance that one those those 2^{s+1} encryption queries can result in C is at most

$$2^s \cdot \frac{1}{2^{|C|}} = \frac{2}{2^\tau} \ .$$

Summing this over q_v verification queries,

$$\Pr[\text{Bad}] \le \frac{2q_v}{2^\tau} \ .$$

Hence

$$\Pr[\mathbf{G}^{\text{rand}}_{\mathsf{SE}}(\mathcal{B})] \le \Pr[\mathbf{G}^{\text{rand}}_{\mathsf{SE}}(\mathcal{A})] + \frac{2q_v}{2^\tau} \ ,$$

and thus

$$\mathbf{Adv}^{\text{mrae}}_{\mathsf{SE}}(\mathcal{B}) \ge \mathbf{Adv}^{\text{mrae}}_{\mathsf{SE}}(\mathcal{A}) - \frac{2q_v}{2^\tau} \ .$$

Moreover, recall that if $\mathcal{A}$ is nonce-respecting then so is $\mathcal{B}$, and in that case the bound above can be rewritten as

$$\mathbf{Adv}^{\text{unae}}_{\mathsf{SE}}(\mathcal{B}) \ge \mathbf{Adv}^{\text{unae}}_{\mathsf{SE}}(\mathcal{A}) - \frac{2q_v}{2^\tau} \ .$$

On the other hand, if every ciphertext in SE is always τ-bit longer than its plaintext then each verification query $\textsc{Vf}(i, N, A, C)$ can be targeted by at most 2^s (instead of 2^{s+1}) encryption queries, where $s = |C| - \tau$ is the length of the corresponding message. As a result, the term $2q_v/2^\tau$ can be improved to $q_v/2^\tau$.

F Proof of Proposition 4.3

Without loss of generality, assume that the adversary won't make redundant queries. That is, (i) it will not repeat a prior query, (ii) if it queries $C \leftarrow E(K, M)$ then later it will not query $E^{-1}(K, C)$, and (iii) if it queries $M \leftarrow E^{-1}(K, C)$ then later it will not query $E(K, M)$. We will store the ideal-cipher queries of the adversary in a transcript, in the order of the queries are made. For the i-th query of the adversary, if it is a forward query $C \leftarrow E(K, M)$ then we store an entry $(i, K, M, C, +)$. Otherwise, if it is a backward query $M \leftarrow E^{-1}(K, C)$ then we store a corresponding entry $(i, K, M, C, -)$.

Let S be the collection of ordered subsets $\mathcal{I} = (r_1, \ldots, r_s)$ in $\{1, \ldots, p\}$, with $r_1 < \cdots < r_s$. For $\mathcal{I} = (r_1, \ldots, r_s)$ with $(r_i, K_i, M_i, C_i, *)$ as the corresponding entry of the r_i-th queries in the transcript, let $\text{Bad}(\mathcal{I})$ be the event that

$$(C_1 \oplus M_1)[1 : m] = \cdots = (C_s \oplus M_s)[1 : m] \ .$$

Let

$$\text{Bad} = \bigcup_{\mathcal{I} \in S} \text{Bad}(\mathcal{I}) \ .$$

IF Bad DOES NOT HAPPEN. We now show that if Bad doesn't happen then the chance that the adversary can produce an s-way multi-collision is at most 2^{1-n}. Suppose that Bad indeed does not happen. Let $\big((K_1, X_1), \ldots, (K_s, X_s)\big)$ be the output of the adversary. We consider the following cases.

Case 1: For each (K_i, X_i), there is a corresponding entry $(r_i, K_i, X_i, C_i, *)$ in the transcript. Without loss of generality, assume that $r_1 < \cdots < r_s$, and let $\mathcal{I} = (r_1, \ldots, r_s)$. Then $(K_1, X_1, \ldots, K_s, X_s)$ is an s-way multi-collision if and only if $\mathrm{Bad}(\mathcal{I})$ happens. As Bad does not happen, the adversary does not create an s-way multi-collision.

Case 2: There is some (K_i, X_i) such that there is no entry $(*, K_i, X_i, *, *)$ in the transcript. Recall that the output $\big((K_1, X_1), \ldots, (K_s, X_s)\big)$ forms an s-way multi-collision if

$$(E(K_1, X_1) \oplus X_1)[1:m] = \cdots = (E(K_s, X_s) \oplus X_s)[1:m] \ .$$

Now, as there is no entry $(*, K_i, X_i, *, *)$ in the transcript, given $E(K_1, X_1), \ldots, E(K_{i-1}, X_{i-1}), E(K_{i+1}, X_{i+1}), \ldots, E(K_s, X_s)$, the random variable $E_K(X_i)$ is uniformly distributed over a set of at least $2^n - p - s \geq 2^{n-1}$ values. Thus the chance that $\mathcal{A}$ creates an s-way multi-collision is at most $2^{n-m}/2^{n-1} = 2^{1-m}$.

BOUNDING THE CHANCE THAT Bad HAPPENS. We claim that for each $\mathcal{I} \in S$,

$$\Pr[\mathrm{Bad}(\mathcal{I})] \leq \frac{2^{s-1}}{2^{(s-1)m}} \ . \tag{1}$$

This claim will be justified later. By the union bound,

$$\begin{aligned} \Pr[\mathrm{Bad}] &= \Pr\Big[\bigcup_{\mathcal{I} \in S} \mathrm{Bad}(\mathcal{I})\Big] \\ &\leq \sum_{\mathcal{I} \in S} \Pr[\mathrm{Bad}(\mathcal{I})] \leq \binom{p}{s} \cdot \frac{2^{s-1}}{2^{(s-1)m}} = \binom{p}{s} \cdot 2^{(1-m)(s-1)} \ . \end{aligned}$$

We now prove the claim in Equation (1). Fix $\mathcal{I} = (r_1, \ldots, r_s) \in S$, and let $(r_i, K_i, M_i, C_i, *)$ be the corresponding entry of the r_i-th queries in the transcript. Fix $i \in \{2, \ldots, s\}$. We consider the following cases.

Case 1: The r_i-th entry is $(r_i, K_i, M_i, C_i, +)$. Then given M_i and all prior queries/answers before the r_i-th query, C_i is uniformly distributed over a set of at least $2^n - p \geq 2^{n-1}$ values, and thus the conditional probability that $(C_i \oplus M_i)[1:m] = (C_1 \oplus M_1)[1:m]$ is at most $2^{n-m}/2^{n-1} = 2^{1-m}$.

Case 2: The r_i-th entry is $(r_i, K_i, M_i, C_i, -)$. Then given C_i and all prior queries/answers before the r_i-th query, M_i is uniformly distributed over a set of at least $2^n - p \geq 2^{n-1}$ values, and thus the conditional probability that $(C_i \oplus M_i)[1:m] = (C_1 \oplus M_1)[1:m]$ is at most $2^{n-m}/2^{n-1} = 2^{1-m}$.

Multiplying these conditional probabilities for all $i \in \{2, \ldots, s\}$, we obtain

$$\Pr[\mathrm{Bad}(\mathcal{I})] \leq 2^{(s-1)(1-m)} \ ,$$

justifying the claim in Equation (1).

WRAPPING THINGS UP. Summing up,

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

G Proof of Proposition 4.4

Let $\ell = m/2$. For $b \in \{1, 2\}$, let $\overline{b}$ be the other element in $\{1, 2\}$. We say that a query $C \leftarrow E(K, M)$ or $M \leftarrow E^{-1}(K, C)$ is *well-formed* if M is of the form $\mathsf{pad}(X, b)$ with $b \in \{1, 2\}$. For a well-formed

query $E(K, \mathsf{pad}(X, b))$, we will immediately grant the adversary the free query $E(K, \mathsf{pad}(X, \overline{b}))$. Likewise, for a well-formed query $\mathsf{pad}(X, b) \leftarrow E^{-1}(K, C)$, we will immediately grant the adversary the free query $E(K, \mathsf{pad}(X, \overline{b}))$. Totally there are at most $2p$ queries.

Without loss of generality, assume that the adversary won't make redundant queries. That is, (i) it will not repeat a prior query, (ii) if it queries $C \leftarrow E(K, M)$ then later it will not query $E^{-1}(K, C)$, and (iii) if it queries $M \leftarrow E^{-1}(K, C)$ then later it will not query $E(K, M)$. We will store all ideal-cipher queries (including the granted ones) in a transcript, in the order of the queries are made. For a forward query $C \leftarrow E(K, M)$, we store an entry $(K, M, C, +)$. Likewise, for a backward query $M \leftarrow E^{-1}(K, C)$, we store a corresponding entry $(K, M, C, -)$.

We call a pair $P = \big[(K, \mathsf{pad}(X, b), C, *), (K, \mathsf{pad}(X, \overline{b}), C', *)\big]$—ordered by their appearance in the transcript—a *couple*. A couple is *positive* if both its queries are forward ones, and is *negative* otherwise. The *child* $\mathsf{Child}(P)$ of this couple is $\mathsf{ITP}[E, r, m](K, X)$. In other words, $\mathsf{Child}(P)$ is $C[1:\ell] \| C'[1:\ell]$ if $b = 1$, and $\mathsf{Child}(P) = C'[1:\ell] \| C[1:\ell]$ otherwise. Let Bad be the event that there are s couples $P_1, \ldots, P_s$ of the same child.

If Bad does not happen. We now show that if Bad doesn't happen then the chance that the adversary can produce an s-way multi-collision is at most 2^{1-m}. Suppose that Bad indeed does not happen. Let $\big((K_1, X_1), \ldots, (K_s, X_s)\big)$ be the output of the adversary. We consider the following cases.

Case 1: For each (K_i, X_i), there is a corresponding unordered couple

$$\{(K_i, \mathsf{pad}(X_i, 1), C_i, *), (K_i, \mathsf{pad}(X_i, 2), C_i', *)\}$$

in the transcript. Then $\big((K_1, X_1), \ldots, (K_s, X_s)\big)$ is not an s-way multi-collision because Bad does not happen.

Case 2: There is some (K_i, X_i) such that there is no couple $\{(K_i, \mathsf{pad}(X_i, 1), C_i, *), (K_i, \mathsf{pad}(X_i, 2), C_i', *)\}$ in the transcript. Thus given the queries/answers of the adversary and its output,

$$E(K_i, \mathsf{pad}(X_i, 1)) \| E(K_i, \mathsf{pad}(X_i, 2))$$

is (conditionally) uniformly distributed over a set of at least $(2^n - 2p - s)^2 \geq 2^{2n-1}$ values. Hence $\mathsf{ITP}[E, r, m](K_i, X_i) = E(K_i, \mathsf{pad}(X_i, 1))[1:\ell] \| E(K_i, \mathsf{pad}(X_i, 2))[1:\ell]$ is (conditionally) uniformly distributed over a set of at least $2^{2n-1}/2^{2n-m} = 2^{m-1}$ values. As a result, the chance that the adversary creates an s-way multi-collision is at most 2^{1-m}.

Bounding the chance that Bad happens. Let $\mathrm{Bad}_{x,i}$ be the event that (i) Bad happens, and (ii) among its couples $P_1, \ldots, P_s$, there are exactly i positive couples, and $s - i$ negative ones, and (iii) $\mathsf{Child}(P_1) = x$. Then

$$\Pr[\mathrm{Bad}] \leq \Pr\Big[\bigcup_{x,i} \mathrm{Bad}_{x,i}\Big] \leq \sum_{x \in \{0,1\}^m} \sum_{i=0}^{s} \Pr[\mathrm{Bad}_{x,i}] \ .$$

Let U be the random variable for the number of positive couples. For each $x \in \{0,1\}^m$, let V_x be the random variable for the number of queries $E^{-1}(K, C)$ such that $C[1:\ell] \in \{x[1:\ell], x[\ell+1:m]\}$. Now, recall that the output of each query, given the prior queries and answers, is (conditionally) uniformly distributed over a set of at least $2^n - 2p \geq 2^{n-0.5}$ elements. Thus for each positive couple, its child is (conditionally) uniformly distributed among a set of at least $2^{2(n-0.5)}/2^{2n-m} = 2^{m-1}$ elements. Moreover, for each $b \in \{1, 2\}$, there are at most 2^r strings in $\{0,1\}^n$ that are of form $\mathsf{pad}(X, b)$. Consider the event that when we query $E^{-1}(K, C)$, it results in an answer of the form $\mathsf{pad}(X, b)$, and the granted query ends up with an answer C' such that $C'[1:\ell] = x[\ell+1:m]$ if $b = 1$, and $C'[1:\ell] = x[1:\ell]$ if $b = 2$. This happens with probability at most $2^{r+1} \cdot 2^{n-\ell}/2^{2(n-0.5)} =$

$2^{r+1-\ell-n}$. Then

$$\Pr[\mathrm{Bad}_{x,i}] \le \mathbf{E}\Big[\binom{U}{i}2^{(1-m)i} \cdot \binom{V_x}{s-i}2^{(r+1-\ell-n)(s-i)}\Big] \ .$$

By using the fact that $\binom{a}{b} \le a^b/b!$ for every integer $a, b \ge 0$,

$$\begin{aligned}\sum_{i=0}^{s}\Pr[\mathrm{Bad}_{x,i}] &\le \sum_{i=0}^{s}\mathbf{E}\Big[\frac{U^i}{i!}2^{(1-m)i} \cdot \frac{(V_x)^{s-i}}{(s-i)!}2^{(r+1-\ell-n)(s-i)}\Big] \\ &= \frac{1}{s!} \cdot \mathbf{E}\Big[\sum_{i=0}^{s}\binom{s}{i}U^i 2^{(1-m)i} \cdot (V_x)^{s-i}2^{(r+1-\ell-n)(s-i)}\Big] \\ &= \frac{1}{s!} \cdot \mathbf{E}\Big[\Big(\frac{U}{2^{m-1}} + \frac{V_x}{2^{\ell+n-r-1}}\Big)^s\Big] \ .\end{aligned}$$

By using the fact that $(a+b)^s \le (2 \cdot \max\{a,b\})^s = 2^s \cdot \max\{a^s, b^s\}$ for every $a, b \ge 0$, we can bound the last expectation by

$$2^s \cdot \mathbf{E}\Big[\Big(\frac{U}{2^{m-1}}\Big)^s + \Big(\frac{V_x}{2^{\ell+n-r-1}}\Big)^s\Big] = \frac{\mathbf{E}[U^s]}{2^{(m-2)s}} + \frac{\mathbf{E}[(V_x)^s]}{2^{(\ell+n-r-2)s}} \ .$$

Summing up,

$$\Pr[\mathrm{Bad}] \le \frac{1}{s!}\sum_x \frac{\mathbf{E}[U^s]}{2^{(m-2)s}} + \frac{\mathbf{E}[(V_x)^s]}{2^{(\ell+n-r-2)s}} \ .$$

On the one hand,

$$\sum_x \mathbf{E}[U^s] = 2^n \cdot \mathbf{E}[U^s] \le 2^n \cdot p^s \ .$$

On the other hand, as $V_x \le p$,

$$\sum_x \mathbf{E}[(V_x)^s] \le \sum_x \mathbf{E}[V_x] \cdot p^{s-1} = \mathbf{E}\Big[\sum_x V_x\Big] \cdot p^{s-1} \ .$$

In the sum $\sum_x V_x$, each query can be counted up to $2^{\ell+1}$ times, and thus

$$\sum_x V_x \le 2^{\ell+1} \cdot p \ .$$

Taking expectation of both sides gives us

$$\sum_x \mathbf{E}[(V_x)^s] \le 2^{\ell+1} \cdot p^s \ .$$

Summing up,

$$\Pr[\mathrm{Bad}] \le \frac{4p^s}{s! \cdot 2^{(m-2)(s-1)}} + \frac{2^{\ell+1} \cdot p^s}{s! \cdot 2^{(\ell+n-r-2)s}} \ .$$

Wrapping up. Summing up,

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

H An Attack on the ITP Construction

Let $\ell = m/2$. Let $Q = 2^{n+\ell-r}$ and $a = 2^{(r-n)} \cdot 2^{n-\ell}/(2^n - 1)$. Let p be an integer such that $2s \le p \le Q/6$ and

$$\Delta = \binom{p}{s} \cdot a^s \le 2^{-\ell} \ .$$

In this section, we give a multi-collision attack on $\mathsf{ITP}[E, r, m]$ for $2p$ ideal-cipher queries.

THE ATTACK. The adversary $\mathcal{A}$ picks arbitrary distinct keys $K_1, \ldots, K_p \in \{0,1\}^k$. For each $i \le p$, it computes $V_i \leftarrow E^{-1}(K_i, 0^n)$. If V_i is of the form $\mathsf{pad}(X_i, 1)$ for some $X_i \in \{0,1\}^r$ then it computes $Z_i \leftarrow E(K_i, \mathsf{pad}(X_i, 2))$. Note that

$$\mathsf{ITP}[E, r, m](K_i, X_i) = 0^\ell \| Z_i[1:\ell] \ .$$

The adversary then searches for s indices $i_1, \ldots, i_s$ such that $Z_{i_1}, \ldots, Z_{i_s}$ have the same ℓ-bit prefix. This can be done in $O(p \log(p))$ by sorting the pairs (i, Z_i) by the ℓ-bit prefix of Z_i. It then outputs $(K_{i_1}, X_{i_1}), \ldots, (K_{i_s}, X_{i_s})$.

ANALYSIS. Let $M = 2^\ell - 1$. For each $i \le p$, view $\mathsf{ITP}[E, r, m](K_i, X_i)$ as throwing a ball into 2^ℓ bins, where the value of the bin corresponds to the number that the string $Z_i[1:\ell]$ encodes. The p throws are independent, but for each i, there is a non-zero chance that ball i does *not* land into any bin. This corresponds to the case that $E^{-1}(K_i, Y_i)$ doesn't have a form $\mathsf{pad}(X_i, 1)$. For each i and for any $t \in \{1, \ldots, M\}$, the chance that ball i lands to bin t is a.

Let Bad_i be the event that there are s balls that land into bin i. There is an s-way multi-collision if there is some bin that contains at least s balls, which happens with probability at least $\Pr\left[\bigcup_{i=1}^{M} \mathrm{Bad}_i\right]$. From the Bonferroni's inequality as stated in Lemma C.2,

$$\Pr\left[\bigcup_{i=1}^{M} \mathrm{Bad}_i\right] \ge \sum_{i=1}^{M} \Pr[\mathrm{Bad}_i] - \sum_{1 \le i < j \le M} \Pr[\mathrm{Bad}_i \cup \mathrm{Bad}_j] \ .$$

Note that for every $i \in \{1, \ldots, M\}$,

$$\begin{aligned} \Pr[\mathrm{Bad}_i] &\ge \binom{p}{s} a^s (1-a)^{p-s} = \Delta (1-a)^{p-s} \\ &\ge \Delta \cdot \left(1 - \frac{1}{Q}\right)^{p-s} \ge \Delta \cdot \left(1 - \frac{1}{Q}\right)^{Q/6} \ge \frac{\Delta}{4^{1/6}} \ge \frac{3\Delta}{4} \ , \end{aligned}$$

where the third last inequality is from the hypothesis that $p \le Q/6$, and the second last inequality is from the fact that $(1 - 1/x)^x \ge 1/4$ for every $x \ge 2$. In contrast,

$$\Pr[\mathrm{Bad}_i \cap \mathrm{Bad}_j] \le \binom{p}{s}\binom{p-s}{s} a^{2s} \le \binom{p}{s}\binom{p}{s} a^{2s} = \Delta^2$$

for every $1 \le i < j \le M$. Hence

$$\begin{aligned} \mathbf{Adv}^{\mathrm{coll}}_{\mathsf{ITP}[E,r,m],s}(\mathcal{A}, s) &\ge \frac{3\Delta \cdot M}{4} - \frac{M(M-1) \cdot \Delta^2}{2} \\ &\ge \frac{\Delta \cdot M}{4} \ge \binom{p}{s} \cdot 2^{\ell - 3} \cdot 2^{-(n+\ell-r)s} \ , \end{aligned}$$

where the second inequality is due to the hypothesis that $\Delta \le 2^{-\ell}$ and thus $\Delta(M-1) \le 1$.

I Proof of Proposition 4.5

As a stepping stone, we first consider the multi-collision resistance on the cascade of two hash functions H_0 and H_1. However, we now consider a more general, parameterized bound, via the t_0-way multi-collision of H_0 and t_1-way multi-collision of H_1, with $t_1 = \lceil s/(t_0 - 1) \rceil$, for any $t_0 \ge 2$.

Lemma I.1 *Let H be the cascade of two hash functions H_0 and H_1 as illustrated in Fig. 7. Let $s \ge t_0 \ge 2$ be integers such that $t_1 = \lceil s/(t_0 - 1) \rceil \ge 2$. Then for any adversary $\mathcal{A}$, we can construct*

adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$ such that

$$\mathbf{Adv}^{\mathrm{coll}}_{H,s}(\mathcal{A}) \leq \max\left\{\mathbf{Adv}^{\mathrm{coll}}_{H_0,t_0}(\mathcal{B}_0), \mathbf{Adv}^{\mathrm{coll}}_{H_1,t_1}(\mathcal{B}_1)\right\} \ .$$

Each adversary B_i runs $\mathcal{A}$, and then runs H_0 on s inputs of $\mathcal{A}$.

Proof: We first describe the adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$. Adversary $\mathcal{B}_0$ runs

$$\big((X_1, Y_1, Z_1), \ldots, (X_s, Y_s, Z_s)\big) \leftarrow\!\!\$\ \mathcal{A} \ .$$

It then partitions $(X_1, Y_1), \ldots, (X_s, Y_s)$ based on the outputs under H_0. We note that if $(X_i, Y_i) = (X_j, Y_j)$ then they are still treated as two pairs (of the same value). Let $P_1, \ldots, P_a$ be the resulting partitions. If there is a partition of at least t_0 pairs of distinct values then $\mathcal{B}_0$ will output those pairs.

Adversary $\mathcal{B}_1$ also runs

$$\big((X_1, Y_1, Z_1), \ldots, (X_s, Y_s, Z_s)\big) \leftarrow\!\!\$\ \mathcal{A} \ .$$

Let $V_i \leftarrow H_0(X_i, Y_i)$ for every $i \leq s$. If there are t_1 pairs of distinct values in $(V_1, Z_1), \ldots, (V_s, Z_s)$ then $\mathcal{B}_1$ will output those pairs.

Assume that $\mathcal{A}$ succeeds in creating an s-way multi-collision. We will show that either $\mathcal{B}_0$ creates a t_0-way multi-collision on H_0 or $\mathcal{B}_1$ creates a t_1-way multi-collision on H_1. We say that a value (A, B) has *degree* d if there are exactly d pairs in $(X_1, Y_1), \ldots, (X_s, Y_s)$ of this value. For each i, let (A_i, B_i) be the value of maximum degree in P_i, and let d_i be the degree of (A_i, B_i). Assume that $\mathcal{B}_0$ does not create a t_0-way multi-collision on H_0, meaning that every partition P_i contains at most $t_0 - 1$ pairs of distinct values. In other words,

$$d_i \geq \frac{|P_i|}{t_0 - 1} \ .$$

Let S_i be the set of indices j such that (X_j, Y_j) has value (A_i, B_i). Note that the sets $S_1, \ldots, S_a$ are disjoint, and each $|S_i| = d_i$. Then the pairs

$$\{(V_j, Z_j) \mid j \in S_1 \cup \cdots \cup S_a\}$$

have distinct values. Indeed, fix i and j in $S_1 \cup \cdots \cup S_a$. If (X_i, Y_i) and (X_j, Y_j) are in different partitions then $V_i \neq V_j$. Otherwise, if (X_i, Y_i) and (X_j, Y_j) are in the same partition, since $\mathcal{A}$ must output distinct inputs for H, we must have $(X_i, Y_i, Z_i) \neq (X_j, Y_j, Z_i)$, and thus $Z_i \neq Z_j$. Hence there are at least

$$\sum_{i=1}^{a} |S_i| = \sum_{i=1}^{a} d_i \geq \sum_{i=1}^{a} \frac{|P_i|}{t_0 - 1} = \frac{s}{t_0 - 1} > t_1 - 1$$

pairs of distinct values in $(V_1, Z_1), \ldots, (V_s, Z_s)$, and thus $\mathcal{B}_1$ will succeed. ∎

Back to the proof of Proposition 4.5, let $t' = \lceil s/(t-1) \rceil \geq \lceil (\sqrt[c]{s})^{c-1} \rceil \geq 2$. Let H' be the cascade of $H_1, \ldots, H_{c-1}$. Note that H is the cascade of H_0 and H'. Using Lemma I.1, we can construct adversaries $\mathcal{B}_0$ and $\mathcal{B}'$ such that

$$\mathbf{Adv}^{\mathrm{coll}}_{H,s}(\mathcal{A}) \leq \max\left\{\mathbf{Adv}^{\mathrm{coll}}_{H_0,t}(\mathcal{B}_0), \mathbf{Adv}^{\mathrm{coll}}_{H',t'}(\mathcal{B}')\right\} \ .$$

Repeating the argument above to bound $\mathbf{Adv}^{\mathrm{coll}}_{H',t'}(\mathcal{B}')$ eventually leads to the claimed bound on $\mathbf{Adv}^{\mathrm{coll}}_{H,s}(\mathcal{A})$. Each constructed adversary $\mathcal{B}_i$ runs the cascade of $H_0, \ldots, H_{\min\{c-2,i\}}$ on s inputs of $\mathcal{A}$.

J Proof of Theorem 5.1

We construct an adversary $\mathcal{B}$ attacking the multi-collision resistance of $\mathsf{DM}[E,\tau]$ as follows. It runs $\big((N_1,K_1,M_1,A_1),\ldots,(N_s,K_s,M_s,A_s)\big) \leftarrow\!\!\$\ \mathcal{A}$. Let $C_i\|T_i$ be the ciphertext of (N_i,A_i,M_i) under $\mathsf{CAU\text{-}C1}$ with key K_i, and suppose that $T_i \leftarrow E(K_i,V_i)\oplus V_i$. Adversary $\mathcal{B}$ then outputs $\big((K_1,V_1),\ldots,(K_s,V_s)\big)$. Note that this output is legitimate since $K_1,\ldots,K_s$ are distinct. If $\mathcal{A}$ succeeds then $T_1=\cdots=T_s$, and thus $\mathcal{B}$ also can create an s-way multi-collision. Then

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

Note that $\mathcal{B}$ needs to get the hash keys to obtain V_i. Thus it will need to run $E(K_i,0^n)$ for every $i \le s$.

K Proof of Theorem 5.1

Our proof is based on the H-coefficient technique of Patarin [45, 20], which we will recall below.

The H-coefficient technique. The H-coefficient technique considers a deterministic distinguisher $\mathcal{A}$ that tries to distinguish a "real" system $\mathbf{S}_1$ from an "ideal" system $\mathbf{S}_0$. The adversary's interactions with those systems define transcripts $\mathcal{T}_1$ and $\mathcal{T}_0$, respectively. The following result bounds the distinguishing advantage of $\mathcal{A}$.

Lemma K.1 [45, 20] *Suppose we can partition the set of valid transcripts for the ideal system into good and bad ones. Further, suppose that there exists a constant $\epsilon \ge 0$ such that $1-\frac{\mathsf{ps}_1(\theta)}{\mathsf{ps}_0(\theta)} \le \epsilon$ for every good transcript θ. Then, the advantage of $\mathcal{A}$ in distinguishing $\mathbf{S}_1$ and $\mathbf{S}_0$ is at most $\epsilon + \Pr[\mathcal{T}_0 \text{ is bad}]$.*

The proof. Let $\mathsf{CAU\text{-}C1}[H,\tau]$ be the idealized version of $\mathsf{CAU\text{-}C1}[E,H,\tau]$ in which each call to $E(K_i,\cdot)$ is replaced by a corresponding call to a truly random function $f_i \leftarrow\!\!\$\ \mathsf{Func}(\{0,1\}^n,\{0,1\}^n)$. Note that game $\mathbf{G}^{\text{rand}}_{\mathsf{CAU\text{-}C1}[H,\tau]}(\mathcal{A})$ coincides with game $\mathbf{G}^{\text{rand}}_{\mathsf{CAU\text{-}C1}[E,H,\tau]}(\mathcal{A})$. To bound the gap between the real games, we construct the following adversary $\mathcal{B}$ attacking the PRF security of E. It runs $\mathcal{A}$ and simulates game $\mathbf{G}^{\text{real}}_{\mathsf{CAU\text{-}C1}[E,H,\tau]}(\mathcal{A})$, but each call to $E(K_i,\cdot)$ is replaced by a corresponding call to $\textsc{Eval}(i,\cdot)$. Then

$$\begin{aligned}\mathbf{Adv}^{\text{prf}}_{E}(\mathcal{B}) &= \Pr[\mathbf{G}^{\text{real}}_{\mathsf{CAU\text{-}C1}[E,H,\tau]}(\mathcal{A})] - \Pr[\mathbf{G}^{\text{real}}_{\mathsf{CAU\text{-}C1}[H,\tau]}(\mathcal{A})] \\ &= \mathbf{Adv}^{\text{unae}}_{\mathsf{CAU\text{-}C1}[E,H,\tau]}(\mathcal{A}) - \mathbf{Adv}^{\text{unae}}_{\mathsf{CAU\text{-}C1}[H,\tau]}(\mathcal{A}) \ .\end{aligned}$$

Adversary $\mathcal{B}$ makes at most $\sigma+q$ queries, with at most $2B$ queries per user. From the Multi-user PRP/PRF Switching Lemma,

$$\mathbf{Adv}^{\text{prf}}_{E}(\mathcal{B}) \le \mathbf{Adv}^{\text{prp}}_{E}(\mathcal{B}) + \frac{2B(\sigma+q)}{2^n} \ .$$

We will use the H-coefficient technique to bound $\mathbf{Adv}^{\text{unae}}_{\mathsf{CAU\text{-}C1}[H,\tau]}(\mathcal{A})$, even for a computationally unbounded $\mathcal{A}$. The real system corresponds to game $\mathbf{G}^{\text{real}}_{\mathsf{CAU\text{-}C1}[H,\tau]}$ whereas the ideal system corresponds to game $\mathbf{G}^{\text{rand}}_{\mathsf{CAU\text{-}C1}[H,\tau]}$. Since we consider computationally unbounded adversaries, without loss of generality, assume that $\mathcal{A}$ is deterministic. From Proposition 4.2, without loss of generality, we can assume that $\mathcal{A}$ is orderly. This can cause a difference of at most $q_v/2^\tau$ in the advantage; we will account for this difference in the final bound. When the adversary finishes querying, in the real world, we will grant it the hash keys of all users, and in the ideal world, we will give it fresh uniformly random n-bit strings. This key revelation can only help the adversary.

DEFINING BAD TRANSCRIPTS. A transcript consists of the revealed hash keys L_i and the following information:

- For each query $C\|T \leftarrow \textsc{Enc}(i, N, A, M)$, let $M = M_1 \cdots M_m$ and $C = C_1 \cdots C_m$, with $|M_m| = |C_m| < n$ and $|M_j| = |C_j| = n$ otherwise. Let $V = H(L_i, A, C) \oplus \mathsf{pad}(N)$ and for each $j < m$, let $P_j = M_j \oplus C_j$. If $|M_m| = 0$ then let $P \leftarrow P_1 \cdots P_{m-1}$, otherwise $P \leftarrow P_1 \cdots P_m$, where $P_m \leftarrow f_i(\mathsf{pad}(N) + m)$ in the real world, and $P_m \leftarrow (C_m \oplus M_m)\|Z$ with $Z \leftarrow\!\!\$\, \{0,1\}^{n-|M_m|}$ in the ideal world. We will store a corresponding entry $(\mathsf{enc}, i, N, A, M, C\|T, V, P)$.
- For each query $\textsc{Vf}(i, N, A, C\|T)$, we will store an entry $(\mathsf{vf}, i, N, A, C\|T, V)$, where $V \leftarrow H(L_i, A, C) \oplus \mathsf{pad}(N)$. Note that we do not need to keep track of the answers of the verification queries, since for any valid transcript in the ideal world, the answers of all verification queries must be false.

Entries are stored in the transcript in the order that the queries are made. A verification queries $(\mathsf{vf}, i, N, A, C\|T, V)$ is *vacuous* if there is another entry $(\mathsf{enc}, i, N, A, C\|T^*, V)$. Without loss of generality, assume that $\mathcal{A}$ makes no vacuous verification queries, since it will get false answers in either world.

A transcript is *bad* if one of the following happens:

(1) There are two (possibly the same) entries $(\mathsf{enc}, i, N, A, M, C\|T, V, P_1 \cdots P_m)$ and $(\mathsf{enc}, i, N^*, A^*, M^*, C^*\|T^*, V^*, P_1^* \cdots P_\ell^*)$ and an index $t \in \{1, \ldots, \ell\}$ such that $V = \mathsf{pad}(N^*) + t$. In the real world, $T = (P_t^* \oplus V)[1 : \tau]$, but this might not happen in the ideal world.
(2) There are two different entries $(\mathsf{enc}, i, N, A, M, C\|T, V, P)$ and $(\mathsf{enc}, i, N^*, A^*, M^*, C^*\|T^*, V^*, P^*)$ such that $V = V^*$. In the real world, $T = T^*$, but this might not happen in the ideal world.
(3) There is an entry $(\mathsf{enc}, i, N, A, M, C\|T, V, P)$ such that $V = 0^n$. In the real world, $T = L_i[1 : \tau]$ but this might not happen in the ideal world.
(4) There is an entry $(\mathsf{vf}, i, N, A, C\|T, V)$ such that $V = 0^n$. This forces the correct tag for the verification query to be $L_i[1 : \tau]$ in the real world, but there is no constraint in the ideal world.
(5) There are two entries $(\mathsf{enc}, i, N, A, M, C\|T, V, P_1 \cdots P_m)$ and $(\mathsf{vf}, i, N^*, A^*, C^*\|T^*, V^*)$ and an index $t \in \{1, \ldots, m\}$ such that $V^* = \mathsf{pad}(N) + t$. This forces the correct tag for the verification query to be $(P_t \oplus V^*)[1 : \tau]$ in the real world, but there is no constraint in the ideal world.
(6) There are entries $(\mathsf{enc}, i, N, A, M, C\|T, V, P)$ and $(\mathsf{vf}, i, N^*, A^*, C^*\|T^*, V^*)$ such that $V = V^*$. This forces the correct tag for the verification query to be T in the real world, but there is no constraint in the ideal world.

If a transcript is not bad and is valid for the ideal system then we say that it is *good*.

PROBABILITY OF BAD TRANSCRIPT. Let $\mathcal{T}_0$ be the random variable for the transcript in the ideal world. We now bound the probability that $\mathcal{T}_0$ is bad. We will fix the queries and answers of the adversary, but still treat the revealed keys as random. That is, we are dealing with the conditional probability that $\mathcal{T}_0$ is bad, given a fixed choice of the queries and answers of the adversary. Our bound holds for any such choice, and thus it also holds for the unconditional probability that $\mathcal{T}_0$ is bad.

For each $j \in \{1, \ldots, 6\}$, let Bad_j be the set of transcripts that violate the j-th constraint of badness. Then from the union bound,

$$\Pr[\mathcal{T}_0 \text{ is bad}] = \Pr[\mathcal{T}_0 \in \mathrm{Bad}_1 \cup \cdots \cup \mathrm{Bad}_6] \le \sum_{j=1}^{6} \Pr[\mathcal{T}_0 \in \mathrm{Bad}_j] \ .$$

We first bound the probability that $\mathcal{T}_0 \in \mathrm{Bad}_1$. Consider two (possibly the same) entries $(\mathsf{enc}, i, N, A, M, C\|T, V, P_1 \cdots P_m)$ and $(\mathsf{enc}, i, N^*, A^*, M^*, C^*\|T^*, V^*, P_1^* \cdots P_\ell^*)$. If $A = M = \varepsilon$ then $V =$

$\mathsf{pad}(N)$ that is different from $\mathsf{pad}(N^*)+t$, for any $t \in \{1,\ldots,2^{n-r}-2\}$. If $(A,M) \neq (\varepsilon,\varepsilon)$, since H is weakly c-regular, the chance that there is $t \in \{1,\ldots,\ell\}$ such that $V = H(L_i,A,C)\oplus\mathsf{pad}(N)$ equals to $\mathsf{pad}(N^*)+t$ is at most

$$\frac{c\ell\cdot(|A|_n+|C|_n)}{2^n} \leq \frac{c(|A|_n+|M|_n)(|A^*|_n+|M^*|_n)}{2^n} \ .$$

Summing over all pairs of enc entries of the same user,

$$\Pr[\mathcal{T}_0 \in \mathrm{Bad}_1] \leq \frac{c\sigma B}{2^n} \ .$$

We now bound the probability that $\mathcal{T}_0 \in \mathrm{Bad}_2$. Consider two different entries $(\mathsf{enc},i,N,A,M,C\|T,V,P)$ and $(\mathsf{enc},i,N^*,A^*,M^*,C^*\|T^*,V^*,P^*)$. If $(A,C)=(A^*,C^*)$ then we must have $N \neq N^*$, otherwise that will lead to $M=M^*$, violating the definitional restrictions. Consequently, $V = H(L_i,A,C)\oplus\mathsf{pad}(N)$ and $V^* = H(L_i,A^*,C^*)\oplus\mathsf{pad}(N^*)$ must be different. If $(A,C)\neq(A^*,C^*)$ then since H is c-AXU, the chance that $V = H(L_i,A,C)\oplus\mathsf{pad}(N)$ and $V^* = H(L_i,A^*,C^*)\oplus\mathsf{pad}(N^*)$ are the same is at most

$$\frac{c\cdot\max\{|A|_n+|C|_n,|A^*|_n+|C^*|_n\}}{2^n} \leq \frac{c(|A|_n+|M|_n+|A^*|_n+|M^*|_n)}{2^n} \ .$$

Summing over all pairs of enc entries of the same user,

$$\Pr[\mathcal{T}_0 \in \mathrm{Bad}_2] \leq \frac{cqB}{2^n} \ .$$

We now bound the probability that $\mathcal{T}_0 \in \mathrm{Bad}_3$. Consider an entry $(\mathsf{enc},i,N,A,M,C\|T,V,P)$. If $A=M=\varepsilon$ then $V=\mathsf{pad}(N)\neq 0^n$. If $(A,M)\neq(\varepsilon,\varepsilon)$, since H is weakly c-regular, the chance $V = H(L_i,A,C)\oplus\mathsf{pad}(N)$ equals to 0^n is at most

$$\frac{c\cdot(|A|_n+|C|_n)}{2^n} = \frac{c(|A|_n+|M|_n)}{2^n} \ .$$

Summing over all enc entries,

$$\Pr[\mathcal{T}_0 \in \mathrm{Bad}_3] \leq \frac{c\sigma}{2^n} \leq \frac{c\sigma B}{2^n} \ .$$

We now bound the probability that $\mathcal{T}_0 \in \mathrm{Bad}_4$. Consider an entry $(\mathsf{vf},i,N,A,C\|T,V)$. If $A=C=\varepsilon$ then $V=\mathsf{pad}(N)\neq 0^n$. If $(A,C)\neq(\varepsilon,\varepsilon)$, since H is weakly c-regular, the chance $V = H(L_i,A,C)\oplus\mathsf{pad}(N)$ equals to 0^n is at most

$$\frac{c\cdot(|A|_n+|C|_n)}{2^n} \ .$$

Summing over all vf entries,

$$\Pr[\mathcal{T}_0 \in \mathrm{Bad}_4] \leq \frac{c\sigma}{2^n} \leq \frac{c\sigma B}{2^n} \ .$$

We now bound the chance that $\mathcal{T}_0 \in \mathrm{Bad}_5$. Consider entries $(\mathsf{enc},i,N,A,M,C\|T,V,P_1\cdots P_m)$ and $(\mathsf{vf},i,N^*,A^*,C^*\|T^*,V^*)$. If $A^*=C^*=\varepsilon$ then $V^*=\mathsf{pad}(N^*)$ that is different from $\mathsf{pad}(N)+t$ for any $t\in\{1,\ldots,2^{n-r}-2\}$. If $(A^*,C^*)\neq(\varepsilon,\varepsilon)$, since H is weakly c-regular, the chance that there is $t\in\{1,\ldots,m\}$ such that $V^* = H(L_i,A^*,C^*)\oplus\mathsf{pad}(N^*)$ equals to $\mathsf{pad}(N)+t$ is at most

$$\frac{cm\cdot(|A^*|_n+|C^*|_n)}{2^n} \leq \frac{c(|A|_n+|M|_n)(|A^*|_n+|M^*|_n)}{2^n} \ .$$

Summing over all pairs of enc and vf entries of the same user,

$$\Pr[\mathcal{T}_0 \in \mathrm{Bad}_5] \leq \frac{c\sigma B}{2^n} \ .$$

Finally, we bound the chance that $\mathcal{T}_0 \in \mathrm{Bad}_6$. Consider two entries $(\mathsf{enc},i,N,A,M,C\|T,V,P)$ and $(\mathsf{vf},i,N^*,A^*,C^*\|T^*,V^*)$. If $(A,C)=(A^*,C^*)$ then since the adversary makes no vacuous

verification queries, we must have $N \neq N^*$. As a result, $V = H(L_i, A, C) \oplus \mathsf{pad}(N)$ and $V^* = H(L_i, A^*, C^*) \oplus \mathsf{pad}(N^*)$ must be different. If $(A, C) \neq (A^*, C^*)$ then since H is c-AXU, the chance that $V = H(L_i, A, C) \oplus \mathsf{pad}(N)$ and $V^* = H(L_i, A^*, C^*) \oplus \mathsf{pad}(N^*)$ are the same is at most

$$\frac{c \cdot \max\{|A|_n + |C|_n, |A^*|_n + |C^*|_n\}}{2^n} \leq \frac{c(|A|_n + |M|_n + |A^*|_n + |M^*|_n)}{2^n} \ .$$

Summing over all pairs of enc entries of the same user,

$$\Pr[\mathcal{T}_0 \in \mathrm{Bad}_6] \leq \frac{cqB}{2^n} \ .$$

Summing up,

$$\Pr[\mathcal{T}_0 \text{ is bad}] \leq \frac{cB(4\sigma + 2q)}{2^n} \ .$$

Transcript ratio. Fix a good transcript θ. Suppose that there are exactly u users in θ. Suppose that the answers for encryption queries of user i in θ consist of totally N_i bits. For each user i, partition the entries $(\mathsf{vf}, i, N, A, C \| T, V)$ in θ according to V, and let S_i be the collection of those partitions. For each $P \in S_i$, let $\mathsf{Tags}(P)$ denote the set of the corresponding tags T, and let $V(P)$ denote the corresponding value V. Note that

$$\sum_{i=1}^{u} \sum_{P \in S_i} |P| \leq q_v \ .$$

Let $\mathcal{T}_1$ and $\mathcal{T}_0$ be the random variables of the transcript of the interaction of the adversary with systems $\mathbf{S}_1$ and $\mathbf{S}_0$, respectively. On the one hand, the event $\mathcal{T}_0 = \theta$ is the composition of the following independent events:

- When we sample u keys $L_i \xleftarrow{\$} \{0,1\}^n$, they end up the same as the values defined in θ. This happens with probability 2^{-nu}.
- For each user i, the answers of its encryption queries end up the same as the values (of totally N_i bits) defined in θ. This happens with probability 2^{-N_i}.

Hence

$$\mathsf{p}_{\mathbf{S}_0}(\theta) = \Pr[\mathcal{T}_0 = \theta] = 2^{-nu} \prod_{i=1}^{u} 2^{-N_i} \ .$$

On the other hand, since θ is good, the event $\mathcal{T}_1 = \theta$ is the composition of the following independent events:

- When we sample u keys $L_i \xleftarrow{\$} \{0,1\}^n$, they end up the same as the values defined in θ. This happens with probability 2^{-nu}.
- For each user i, the (possibly truncated) answers of the distinct queries to the truly random function f_i end up the same as the values (of totally N_i bits) defined in θ. This happens with probability 2^{-N_i}.
- For each user i and each $P \in S_i$, if we query $V(P)$ to the truly random function f_i, the answer R must satisfy $(R \oplus V(P))[1:\tau] \notin \mathsf{Tags}(P)$. This happens with probability $1 - |\mathsf{Tags}(P)|/2^\tau \geq 1 - |P|/2^\tau$.

Thus

$$\mathsf{p}_{\mathbf{S}_1}(\theta) = \Pr[\mathcal{T}_1 = \theta] \quad \geq \quad 2^{-nu} \prod_{i=1}^{u} \Big(2^{-N_i} \cdot \prod_{P \in S_i} (1 - |P|/2^\tau) \Big) \ .$$

Hence

$$\frac{\mathsf{ps}_1(\theta)}{\mathsf{ps}_0(\theta)} \geq \prod_{i=1}^{u} \prod_{P \in S_i} (1 - |P|/2^\tau) \geq 1 - \sum_{i=1}^{u} \sum_{P \in S_i} |P|/2^\tau \geq 1 - q_v/2^\tau \ ,$$

where the second inequality is obtained by repeatedly using the fact that for every $x, y \in [0,1]$, $(1-x)(1-y) \geq 1-(x+y)$.

WRAPPING UP. From Lemma K.1,

$$\mathbf{Adv}^{\mathrm{unae}}_{\mathsf{CAU\text{-}C1}[H,\tau]}(\mathcal{A}) \leq \frac{cB(4\sigma+2q)}{2^n} + \frac{q_v}{2^\tau} \ .$$

Hence

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

If we remove the restriction that $\mathcal{A}$ is orderly, from Proposition 4.2,

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

L Proof of Theorem 3.1

We first describe the adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$. Adversary $\mathcal{B}_0$ runs

$$\big((K_1, N_1, A_1, M_1), \ldots, (K_s, N_s, A_s, M_s)\big) \leftarrow\!\!\$\ \mathcal{A} \ .$$

Due to the equivalence between CMT_s-3 and CMT_s-4, without loss of generality, assume that $(K_1, N_1, A_1), \ldots, (K_s, N_s, A_s)$ are distinct. Adversary $\mathcal{B}_0$ partitions $(K_1, N_1, A_1), \ldots, (K_s, N_s, A_s)$ based on the outputs under H. Let $P_1, \ldots, P_a$ be the resulting partitions. If there is a partition of at least t elements then $\mathcal{B}_0$ will output those.

Adversary $\mathcal{B}_1$ also runs

$$\big((K_1, N_1, A_1, M_1), \ldots, (K_s, N_s, A_s, M_s)\big) \leftarrow\!\!\$\ \mathcal{A} \ .$$

Let $L_i \leftarrow H(K_i, (N_i, A_i))$ for every $i \leq s$. If there are t distinct subkeys in $(L_1, N_1, \varepsilon, M_1), \ldots, (L_s, N_s, \varepsilon, M_s)$ then $\mathcal{B}_1$ will output those tuples.

Let $C_i \leftarrow \overline{\mathsf{SE}}(K_i, N_i, A_i, M_i)$; note that $C_i = \mathsf{SE.Enc}(L_i, N_i, \varepsilon, M_i)$. Assume that $\mathcal{A}$ succeeds in creating an s-way multi-collision, meaning that $C_1 = \cdots = C_s$. If there is a partition P_i of at least t elements then $\mathcal{B}_0$ creates a t-way multi-collision on H. Assume to the contrary that $|P_i| \leq t-1$ for every $i \in \{1, \ldots, a\}$. Then the number a of partitions is at least

$$\left\lceil \frac{s}{t-1} \right\rceil \geq \lceil \sqrt{s}\, \rceil = t \ .$$

Thus among the subkeys $L_1, \ldots, L_s$, there are at least t distinct ones, and $\mathcal{B}_1$ can output the corresponding tuples $(L_i, N_i, \varepsilon, M_i)$ and wins. Hence

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

as claimed.

M Proof of Theorem 3.2

Let $\mathbf{G}_0$ be game $\mathbf{G}^{\mathrm{real}}_{\mathsf{HtE}[\mathsf{SE},H]}(\mathcal{A})$ and $\mathbf{G}_2$ be game $\mathbf{G}^{\mathrm{rand}}_{\mathsf{HtE}[E,H]}(\mathcal{A})$. Consider game $\mathbf{G}_1$ in Fig. 19.

FROM $\mathbf{G}_0$ TO $\mathbf{G}_1$. Game $\mathbf{G}_1$ is similar to $\mathbf{G}_0$, but with the following differences.

Game $\mathbf{G}_1(\mathcal{A})$	$\underline{\text{NEW}()}$
$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\text{NEW},\text{ENC},\text{VF}}$	$v \leftarrow v + 1$
Return $(b' = 1)$	$L_v \leftarrow\!\!\$\ \{0,1\}^k$
$\underline{\text{ENC}(i, N, A, M)}$	$\underline{\text{VF}(i, N, A, C)}$
If $i \notin \{1, \ldots, v\}$ then return $\bot$	If $i \notin \{1, \ldots, v\}$ then return $\bot$
If $\mathsf{Users}[i, N, A] = \bot$ then	If $\mathsf{Users}[i, N, A] = \bot$ then
NEW(); $\mathsf{Users}[i, N, A] \leftarrow v$	NEW(); $\mathsf{Users}[i, N, A] \leftarrow v$
$u \leftarrow \mathsf{Users}[i, N, A]$; $C \leftarrow \mathsf{SE.Enc}(L_u, N, \varepsilon, M)$	$u \leftarrow \mathsf{Users}[i, N, A]$; $M \leftarrow \mathsf{SE.Dec}(L_u, N, \varepsilon, C)$
Return C	Return $(M \neq \bot)$

Figure 19: Game $\mathbf{G}_1$ in the proof of Theorem 3.2.

$\underline{\text{Adversary } \mathcal{D}^{\text{NEW},\text{EVAL}}}$	$\underline{\overline{\text{NEW}}}$
$v \leftarrow 0$; $u \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\overline{\text{NEW}},\overline{\text{ENC}},\overline{\text{VF}}}$	$v \leftarrow v + 1$
Return b'	
$\underline{\overline{\text{ENC}}(i, N, A, M)}$	$\underline{\overline{\text{VF}}(i, N, A, C)}$
If $i \notin \{1, \ldots, v\}$ then return $\bot$	If $i \notin \{1, \ldots, v\}$ then return $\bot$
If $\mathsf{Users}[i, N, A] = \bot$ then	If $\mathsf{Users}[i, N, A] = \bot$ then
$\mathsf{Users}[i, N, A] \leftarrow u \leftarrow u + 1$; NEW()	$\mathsf{Users}[i, N, A] \leftarrow u \leftarrow u + 1$; NEW()
$u^* \leftarrow \mathsf{Users}[i, N, A]$	$u^* \leftarrow \mathsf{Users}[i, N, A]$
$C \leftarrow \text{ENC}(u^*, N, \varepsilon, M)$; Return C	$b \leftarrow \text{VF}(u^*, N, \varepsilon, M)$; Return b

Figure 20: Constructed adversary $\mathcal{D}$ in the proof of Theorem 3.2.

- In an encryption query $\text{ENC}(i, N, A, M)$, instead of generating a genuine subkey L, if there is a prior verification query $\text{VF}(i, N, A, C)$ then we reuse the same subkey. Otherwise we'll pick L at random.
- In a verification query $\text{VF}(i, N, A, C)$, if there is a prior query $\text{ENC}(i, N, A, M)$ or $\text{VF}(i, N, A, C^*)$, then we use the prior subkey L. Otherwise, we'll generate a fresh subkey.

We bound the gap between $\mathbf{G}_0$ and $\mathbf{G}_1$ by constructing an adversary $\mathcal{B}$ attacking the (multi-user) PRF security of H. Adversary $\mathcal{B}$ runs $\mathcal{A}$ and simulates game $\mathbf{G}_0$. However, each call to $H(K_i, \cdot)$ is replaced by the corresponding call to $\text{EVAL}(i, \cdot)$. Thus

$$\Pr[\mathbf{G}_0(\mathcal{A})] - \Pr[\mathbf{G}_1(\mathcal{A})] \leq \mathbf{Adv}^{\text{prf}}_H(\mathcal{B}) \ .$$

FROM $\mathbf{G}_1$ TO $\mathbf{G}_2$. We bound the gap between $\mathbf{G}_1$ and $\mathbf{G}_2$ by constructing an adversary $\mathcal{D}$ attacking SE as in Fig. 20. Game $\mathbf{G}^{\text{real}}_{\mathsf{SE}}(\mathcal{D})$ corresponds to game $\mathbf{G}_1(\mathcal{A})$, and game $\mathbf{G}^{\text{rand}}_{\mathsf{SE}}(\mathcal{D})$ corresponds to game $\mathbf{G}_2(\mathcal{A})$. Hence

$$\Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_2(\mathcal{A})] \leq \mathbf{Adv}^{\text{mrae}}_{\mathsf{SE}}(\mathcal{D}) \ .$$

We now briefly describe how to implement the map Users in $O(\sigma_a \log(B))$ time. Tuples (i, N, A) and their values u^* are partitioned according to $(i, N, |A|)$. Within a partition, the tuples are stored in a binary search tree, with A as the key, and u^* as the value. Let $P_1, \ldots, P_d$ be the resulting partitions, and let σ_t be the size of P_t when $\mathcal{A}$ finishes querying. Since each binary search tree has

size at most B, the total cost of updates and look-ups in the trees is

$$\sum_{t=1}^{d} O(\sigma_t \log(B)) = O(\sigma_a \log(B))$$

as claimed.

Wrap-up. Summing up,

$$\mathbf{Adv}^{\mathrm{mrae}}_{\mathsf{HtE}[\mathsf{SE},H]}(\mathcal{A}) \;=\; \Pr[\mathbf{G}_0(\mathcal{A})] - \Pr[\mathbf{G}_2(\mathcal{A})] \le \mathbf{Adv}^{\mathrm{prf}}_{H}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{mrae}}_{\mathsf{SE}}(\mathcal{D}) \ .$$

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

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

N Proof of Proposition 6.1

We begin by giving a brief review of the Chi-Squared method of Dai, Hoang, and Tessaro [22].

The Chi-Squared method. Suppose that we want to bound the advantage of a computationally unbounded adversary $\mathcal{A}$ in distinguishing a "real" system $\mathbf{S}_1$ from an "ideal" system $\mathbf{S}_0$. Without loss of generality, assume that $\mathcal{A}$ is deterministic and makes exactly q queries. Since the adversary is deterministic, for any $i \le q-1$, the answers for the first i queries completely determine the first $i+1$ queries. For a system $\mathbf{S} \in \{\mathbf{S}_1, \mathbf{S}_0\}$ and strings $z_1, \ldots, z_i$, let $\mathsf{ps}_{\mathbf{S},\mathcal{A}}(z_1, \ldots, z_i)$ denote the probability that the answers for the first i queries that $\mathcal{A}$ receives when interacting with $\mathbf{S}$ are $z_1, \ldots, z_i$. If $\mathsf{ps}_{\mathbf{S},\mathcal{A}}(z_1, \ldots, z_i) > 0$, let $\mathsf{ps}_{\mathbf{S},\mathcal{A}}(z_{i+1} \mid z_1, \ldots, z_i)$ denote the conditional probability that the answer for the $(i+1)$-th query when $\mathcal{A}$ interacts with system $\mathbf{S}$ is z_{i+1}, given that the answers for the first i queries are $z_1, \ldots, z_i$ respectively.

For each $\boldsymbol{Z} = (z_1, \ldots, z_q)$, let $\boldsymbol{Z}_i = (z_1, \ldots, z_i)$ and let $\boldsymbol{Z}_0$ be the empty string. We write $\mathsf{ps}_{\mathbf{S},A}(\cdot \mid \boldsymbol{Z}_i)$ and $\mathsf{ps}_{\mathbf{S},A}(\cdot \mid \boldsymbol{Z}_0)$ to refer to probabilities $\mathsf{ps}_{\mathbf{S},A}(\cdot \mid z_1, \ldots, z_i)$ and $\mathsf{ps}_{\mathbf{S},A}(\cdot)$ respectively. We require that if $\mathsf{ps}_{\mathbf{S}_1,\mathcal{A}}(\boldsymbol{Z}_i) > 0$ then so is $\mathsf{ps}_{\mathbf{S}_0,\mathcal{A}}(\boldsymbol{Z}_i)$. For each $i \le q$ and each vector $\boldsymbol{Z}_{i-1} = (z_1, \ldots, z_{i-1})$, define

$$\chi^2(\boldsymbol{Z}_{i-1}) \;=\; \sum_{z_i} \frac{\big(\mathsf{ps}_{\mathbf{S}_1,\mathcal{A}}(z_i \mid \boldsymbol{Z}_{i-1}) - \mathsf{ps}_{\mathbf{S}_0,\mathcal{A}}(z_i \mid \boldsymbol{Z}_{i-1})\big)^2}{\mathsf{ps}_{\mathbf{S}_0,\mathcal{A}}(z_i \mid \boldsymbol{Z}_{i-1})} \ ,$$

where the sum is taken over all z_i such that $\mathsf{ps}_{\mathbf{S}_0,\mathcal{A}}(z_i \mid \boldsymbol{Z}_{i-1}) > 0$. Lemma N.1 below bounds the statistical distance $\mathsf{SD}\big(\mathsf{ps}_{\mathbf{S}_0,A}(\cdot), \mathsf{ps}_{\mathbf{S}_1,A}(\cdot)\big)$ between $\mathsf{ps}_{\mathbf{S}_0,A}(\cdot)$ and $\mathsf{ps}_{\mathbf{S}_1,A}(\cdot)$, namely the best possible distinguishing advantage of $\mathcal{A}$ between $\mathbf{S}_1$ and $\mathbf{S}_0$.

Lemma N.1 (The Chi-Squared Lemma) *[22, Lemma 3] Suppose whenever* $\mathsf{ps}_{\mathbf{S}_1,\mathcal{A}}(\boldsymbol{Z}_i) > 0$ *then* $\mathsf{ps}_{\mathbf{S}_0,\mathcal{A}}(\boldsymbol{Z}_i) > 0$. *Then*

$$\mathsf{SD}\big(\mathsf{ps}_{\mathbf{S}_0,A}(\cdot), \mathsf{ps}_{\mathbf{S}_1,A}(\cdot)\big) \le \Big(\frac{1}{2}\sum_{i=1}^{q} \mathbf{E}[\chi^2(\boldsymbol{X}_{i-1})]\Big)^{1/2} \ ,$$

where the expectation is taken over vectors $\boldsymbol{X}_{i-1}$ *of the* $i-1$ *first answers sampled according to the interaction with* $\mathbf{S}_1$.

The proof. Consider games $\mathbf{G}_1$–$\mathbf{G}_3$ in Fig. 21. Game $\mathbf{G}_1$ corresponds to game $\mathbf{G}^{\mathrm{prf}}_E(\mathcal{A})$ with challenge bit 1, and game $\mathbf{G}_3$ corresponds to game $\mathbf{G}^{\mathrm{prf}}_E(\mathcal{A})$ with challenge bit 0. Thus

$$\mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{A}) = \Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_3(\mathcal{A})] \ .$$

Game $\mathbf{G}_2$ is similar to game $\mathbf{G}_1$, but each call to $E(K_i, \cdot)$ is replaced by another call to a truly random permutation $\pi_i(\cdot)$. To bound the gap between $\mathbf{G}_1$ and $\mathbf{G}_2$, we construct a (multi-user)

Game $\mathbf{G}_1(\mathcal{A})$	Game $\mathbf{G}_2(\mathcal{A})$	Game $\mathbf{G}_3(\mathcal{A})$
$v \leftarrow 0$; $b' \leftarrow \mathcal{A}^{\text{New},\text{Eval}}$	$v \leftarrow 0$; $b' \leftarrow \mathcal{A}^{\text{New},\text{Eval}}$	$v \leftarrow 0$; $b' \leftarrow \mathcal{A}^{\text{New},\text{Eval}}$
Return $(b' = 1)$	Return $(b' = 1)$	Return $(b' = 1)$
New()	New()	New()
$v \leftarrow v + 1$	$v \leftarrow v + 1$	$v \leftarrow v + 1$
$K_v \leftarrow\!\!\$\ \{0,1\}^k$	$\pi_v \leftarrow\!\!\$\ \mathsf{Perm}(\{0,1\}^n)$	$f_v \leftarrow\!\!\$\ \mathsf{Func}(\{0,1\}^n, \{0,1\}^m)$
Eval(i, X)	Eval(i, X)	Eval(i, X)
If $i \notin \{1, \ldots, v\}$ return $\perp$	If $i \notin \{1, \ldots, v\}$ return $\perp$	If $i \notin \{1, \ldots, v\}$ return $\perp$
Return $E(K_i, X)[1 : m]$	Return $\pi_i(X)[1 : m]$	Return $f_i(X)$

Figure 21: Games in the proof of Proposition 6.1.

PRP adversary $\mathcal{B}$ as follows. Adversary $\mathcal{B}$ runs $\mathcal{A}$ and forwards $\mathcal{A}$'s queries to its corresponding oracles. Finally, when $\mathcal{A}$ outputs a bit b', so does $\mathcal{B}$. Thus game $\mathbf{G}_1$ corresponds to game $\mathbf{G}_E^{\text{prp}}(\mathcal{B})$ with challenge bit 1, and game $\mathbf{G}_2$ corresponds to game $\mathbf{G}_E^{\text{prp}}(\mathcal{B})$ with challenge bit 0. Hence

$$\mathbf{Adv}_E^{\text{prp}}(\mathcal{B}) = \Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_2(\mathcal{A})] \ .$$

Next, we will use the Chi-Squared method to show that

$$\Pr[\mathbf{G}_2(\mathcal{A})] - \Pr[\mathbf{G}_3(\mathcal{A})] \leq 2\sqrt{nBq} \cdot 2^{m/2-n} \ . \tag{2}$$

On the other hand, since the difference between $\mathbf{G}_2$ and $\mathbf{G}_3$ is whether we truncate truly random permutations or truly random functions, by using the Multi-user PRP/PRF Switching Lemma,

$$\Pr[\mathbf{G}_2(\mathcal{A})] - \Pr[\mathbf{G}_3(\mathcal{A})] \leq Bq/2^n \ ,$$

and thus

$$\Pr[\mathbf{G}_2(\mathcal{A})] - \Pr[\mathbf{G}_3(\mathcal{A})] \leq \min\{2\sqrt{nBq} \cdot 2^{m/2-n}, Bq/2^n\} \ .$$

Summing up,

$$\begin{aligned} \mathbf{Adv}_E^{\text{prf}}(\mathcal{A}) &= \Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_3(\mathcal{A})] \\ &\leq \mathbf{Adv}_E^{\text{prp}}(\mathcal{B}) + \min\{2\sqrt{nBq} \cdot 2^{m/2-n}, Bq/2^n\} \ . \end{aligned}$$

We now justify the claim in Equation (2). Let $\mathbf{S}_1$ be the system that implements game $\mathbf{G}_2$ and $\mathbf{S}_0$ be the system that implements game $\mathbf{G}_3$. Without loss of generality, assume that $\mathcal{A}$ makes exactly q queries. Let $\boldsymbol{X} = (X_1, \ldots, X_q)$ be the random variable for the q answers in $\mathbf{S}_1$, and let $\boldsymbol{X}_i = (X_1, \ldots, X_i)$ for every $i \leq q$. Fix $i \leq q$. Let U_i be the random variable for the user that the i-th query targets. Let Q_i be the number of queries for user U_i before the i-th query is made, and let $H_{i,x}$ be the the number of those Q_i queries that end up with the answer x. Let $N = 2^n$ and $M = 2^m$. Then

$$\begin{aligned} \chi^2(\boldsymbol{X}_i) &= \sum_{x \in \{0,1\}^m} M \cdot \Big(\frac{N/M - H_{i,x}}{N - Q_i} - \frac{1}{M}\Big)^2 \\ &= \frac{M}{(N-Q_i)^2} \sum_{x \in \{0,1\}^m} \Big(\frac{Q_i}{M} - H_{i,x}\Big)^2 \leq \frac{4M}{N^2} \sum_{x \in \{0,1\}^m} \Big(\frac{Q_i}{M} - H_{i,x}\Big)^2 \ , \end{aligned}$$

where the inequality is due the fact that $Q_i \leq B \leq N/2$. Taking expectation of both sides, we

obtain

$$\chi^2(\boldsymbol{X}_i) \le \frac{4M}{N^2} \sum_{x\in\{0,1\}^m} \mathbf{E}\Big[\Big(\frac{Q_i}{M} - H_{i,x}\Big)^2\Big] \ .$$

Fix $x \in \{0,1\}^m$. On the one hand, we will show that

$$\Pr\Big[\Big(\frac{Q_i}{M} - H_{i,x}\Big)^2 \ge 1.5nB\Big] \le 2^{-2n} \ . \tag{3}$$

On the other hand, as both Q_i/M and $H_{i,x}$ are smaller than B, we have

$$\Big(\frac{Q_i}{M} - H_{i,x}\Big)^2 \le B^2 \ .$$

Thus

$$\mathbf{E}\Big[\Big(\frac{Q_i}{M} - H_{i,x}\Big)^2\Big] \le \frac{B^2}{2^{2n}} + 1.5nB \le 2nB \ .$$

Hence from Lemma N.1, the distinguishing advantage of $\mathcal{A}$ against $\mathbf{S}_1$ and $\mathbf{S}_0$ is at most

$$\Big(\frac{1}{2}\sum_{i=0}^{q-1}\chi^2(\boldsymbol{X}_i)\Big)^{1/2} \le \Big(\sum_{i=0}^{q-1}\frac{2M}{N^2}\sum_{x\in\{0,1\}^m}\mathbf{E}\Big[\Big(\frac{Q_i}{M} - H_{i,x}\Big)^2\Big]\Big)^{1/2} \le \frac{2\sqrt{nMBq}}{N} \ ,$$

justifying Equation (2).

To prove the claim in Equation (3), it suffices to consider $i > 1$, as otherwise $Q_i = H_{i,x} = 0$. We will first give a much tighter bound in the single-user setting, showing that

$$\Pr\Big[\Big(\frac{Q_i}{M} - H_{i,x}\Big)^2 \ge 1.5nB\Big] \le 2^{-4n} \ .$$

We will then obtain Equation (3) for the multi-user case by applying a hybrid argument with a blow-up factor $qB \le 2^{2n}$.

THE SINGLE-USER CASE. For the single-user setting, $Q_i = i - 1$, and $H_{i,x}$ has the hypergeometric distribution $\mathrm{HypGeo}(i-1, N/M, N)$.[1] The following classic result gives a strong concentration bound for hypergeometric variables.

Lemma N.2 [34] *Let X be a random variable of distribution $\mathrm{HypGeo}(r, S, N)$, and let $\mu = \mathbf{E}[X] = rS/N$. Then for every $\lambda > 0$,*

$$\Pr[|X - \mu| \ge \lambda r] \le 2e^{-2\lambda^2 r} \ .$$

Using Lemma N.2 for $H_{i,x}$ with $\lambda = \sqrt{1.5n/(i-1)}$, we obtain

$$\Pr\left[\Big|\frac{Q_i}{M} - H_{i,x}\Big| \ge \sqrt{1.5n(i-1)}\right] \le 2e^{-3n} \le 2^{-4n} \ .$$

As $i - 1 \le B$,

$$\begin{aligned}\Pr\Big[\Big(\frac{Q_i}{M} - H_{i,x}\Big)^2 \ge 1.5nB\Big] &= \Pr\left[\Big|\frac{Q_i}{M} - H_{i,x}\Big| \ge \sqrt{1.5nB}\right] \\ &\le \Pr\left[\Big|\frac{Q_i}{M} - H_{i,x}\Big| \ge \sqrt{1.5n(i-1)}\right] \le 2^{-4n} \ .\end{aligned}$$

THE MULTI-USER CASE. Consider the following games $G_{t,s}$, with $t \in \{1, \ldots, q\}$ and $s \in \{0, \ldots, B -$

[1] The hypergeometric distribution $\mathrm{HypGeo}(r, S, N)$ describes the number of red balls when we sample uniformly without replacement of r balls from a set of S red balls and $N - S$ blue balls.

Game $\mathbf{G}_1(\mathcal{A})$	Games $\mathbf{G}_2(\mathcal{A})$, $\mathbf{G}_3(\mathcal{A})$
$v \leftarrow 0$; $b' \leftarrow\$\ \mathcal{A}^{\text{New},\text{Enc}}$; Return $(b' = 1)$	$v \leftarrow 0$; $b' \leftarrow\$\ \mathcal{A}^{\text{New},\text{Enc}}$; Return $(b' = 1)$
New()	New()
$v \leftarrow v + 1$; $K_v \leftarrow\$\ \{0,1\}^k$	$v \leftarrow v + 1$
Enc(i, M)	Enc(i, M)
If $i \notin \{1, \ldots, v\}$ return $\perp$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$M_1 \cdots M_m \leftarrow M$; $\mathsf{IV} \leftarrow\$\ \{0,1\}^n$	$M_1 \cdots M_m \leftarrow M$; $\mathsf{IV} \leftarrow\$\ \{0,1\}^n$
For $j \leftarrow 1$ to m do	For $j \leftarrow 1$ to m do
$V \leftarrow \mathsf{add}(\mathsf{IV}, j)$; $X \leftarrow E(K_i, V)$	$V \leftarrow \mathsf{add}(\mathsf{IV}, j)$; $X \leftarrow\$\ \{0,1\}^n$
$C_j \leftarrow X \oplus M_j$	If $\mathsf{Tbl}[i, V] \neq \perp$ then
Return $\mathsf{IV} \| C_1 \cdots C_m$	$\mathsf{bad} \leftarrow \mathsf{true}$; $X \leftarrow \mathsf{Tbl}[i, V]$
	$\mathsf{Tbl}[i, V] \leftarrow X$; $C_j \leftarrow X \oplus M_j$
	Return $\mathsf{IV} \| C_1 \cdots C_m$

Game $\mathbf{G}_4(\mathcal{A})$	Enc(i, M)
$v \leftarrow 0$; $b' \leftarrow\$\ \mathcal{A}^{\text{New},\text{Enc}}$; Return $(b' = 1)$	If $i \notin \{1, \ldots, v\}$ return $\perp$
New()	$C \leftarrow\$\ \{0,1\}^{\lvert M \rvert}$; $\mathsf{IV} \leftarrow\$\ \{0,1\}^n$
$v \leftarrow v + 1$	Return $\mathsf{IV} \| C$

Figure 22: Games $\mathbf{G}_1$–$\mathbf{G}_4$ in the proof of Proposition 6.3. Game $\mathbf{G}_2$ contains the corresponding highlighted code, but game $\mathbf{G}_3$ does not.

1}. In game $G_{i,r}$, one picks a random variable $V_{t,s}$ from the distribution HypGeo$(s, N/M, N)$. One wins this game if $(V_{t,s} - s/M)^2 \geq 1.5nB$.

In the multi-user setting, the adversary can be viewed as playing the qB games above simultaneously; in game $G_{t,s}$ it targets user $U_i = t$ with $Q_i = s$ queries. The probability $\Pr\left[\left(\frac{Q_i}{M} - H_{i,x}\right)^2 \geq 1.5nB\right]$ is at most the chance that the adversary wins some game. For each fixed t and s, the chance the adversary wins the game $G_{t,s}$, as shown in the single-user setting, is at most 2^{-4n}. Hence by the union bound, the chance that the adversary wins some game is at most $qB/2^{4n}$. Without loss of generality, assume that $qB \leq 2^{2n}$, otherwise the claim in this theorem is moot. Then

$$\Pr\left[\left(\frac{Q_i}{M} - H_{i,x}\right)^2 \geq 1.5nB\right] \leq \frac{qB}{2^{4n}} \leq 2^{-2n} \ .$$

O Proof of Proposition 6.3

Without loss of generality, assume that for each query Enc(i, M), the length $|M|$ is a multiple of n. Indeed, a query Enc(i, M) with a fragmentary M can be simulated by (i) padding M to obtain a full-block message M', (ii) querying Enc(i, M'), and (iii) truncating the answer.

Consider games $\mathbf{G}_1$–$\mathbf{G}_4$ in Fig. 22. From the definition,

$$\mathbf{Adv}^{\text{ind}}_{\mathsf{CTR}[E,\mathsf{add}]}(\mathcal{A}) = \Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_4(\mathcal{A})] \ .$$

We now describe the game chain. In game $\mathbf{G}_2$, each call to $E(K_i, \cdot)$ is replaced by a corresponding call to a truly random function $f_i \leftarrow\$\ \mathsf{Func}(\{0,1\}^n, \{0,1\}^n)$. These functions are lazily implemented by maintaining a table Tbl of the defined points. To bound the gap between $\mathbf{G}_1$ and $\mathbf{G}_2$, we

construct the following adversary $\mathcal{B}$ attacking the PRF security E. Adversary $\mathcal{B}$ runs $\mathcal{A}$ and simulates game $\mathbf{G}_1$, but each call to $E(K_i, \cdot)$ is replaced by a corresponding query to $\textsc{Eval}(i, \cdot)$. Then $\mathcal{B}$'s real world corresponds to game $\mathbf{G}_1$, and its ideal world corresponds to game $\mathbf{G}_2$, and thus

$$\mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{B}) = \Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_2(\mathcal{A})] \ .$$

Adversary $\mathcal{B}$ therefore makes at most σ queries, with B queries per user. From the Multi-user PRP/PRF Switching Lemma,

$$\mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{B}) \leq \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{B\sigma}{2^n} \ .$$

In game $\mathbf{G}_2$, if there are two queries $\textsc{Enc}(i, M_0)$ and $\textsc{Enc}(i, M_1)$ that internally call $f_i(\mathsf{IV}_0 + i)$ and $f_i(\mathsf{IV}_1 + j)$ respectively, and $\mathsf{IV}_0 + i = \mathsf{IV}_1 + j$, then they will get the same answer. Instead, in game $\mathbf{G}_3$, the answers are chosen independently. The two games are identical until the flag bad is set, and thus from the Fundamental Lemma of Game Playing [11],

$$\Pr[\mathbf{G}_2(\mathcal{A})] - \Pr[\mathbf{G}_3(\mathcal{A})] \leq \Pr[\mathbf{G}_3(\mathcal{A}) \text{ sets } \mathsf{bad}] \ .$$

We now bound the chance that game $\mathbf{G}_3$ sets bad. Consider the j-th encryption query. Suppose that it targets user i and let L_j be the block length of the message. Note that the adversary can choose L_j adaptively, and thus it is actually a random variable, depending on the output of the game. This query leads to L_j distinct calls to f_i. Since (i) there are at most B prior calls to f_i, (ii) marginally, the $(n-1)$-bit suffix of each call is uniformly random, and (iii) two calls from different queries are independent, the probability that this query triggers bad is at most

$$\frac{B \cdot \mathbf{E}[L_j]}{2^{n-1}} \ .$$

Summing this over all queries,

$$\Pr[\mathbf{G}_3(\mathcal{A}) \text{ sets } \mathsf{bad}] \leq \frac{2B}{2^n} \cdot \mathbf{E}\Big[\sum_j L_j\Big] \leq \frac{2B\sigma}{2^n} \ .$$

Game $\mathbf{G}_4$ is a simplification of game $\mathbf{G}_3$, and thus

$$\Pr[\mathbf{G}_4(\mathcal{A})] = \Pr[\mathbf{G}_3(\mathcal{A})] \ .$$

Summing up,

$$\begin{aligned} \mathbf{Adv}^{\mathrm{ind}}_{\mathsf{CTR}[E,\mathsf{add}]}(\mathcal{A}) &= \Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_4(\mathcal{A})] \\ &= \sum_{i=1}^{3} \Pr[\mathbf{G}_i(\mathcal{A})] - \Pr[\mathbf{G}_{i+1}(\mathcal{A})] \leq \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{3B\sigma}{2^n} \ . \end{aligned}$$

P Proof of Proposition 6.4

Consider games $\mathbf{G}_1$–$\mathbf{G}_4$ in Fig. 23. From the definition,

$$\mathbf{Adv}_{\mathsf{GMAC2}[E,G]}(\mathcal{A}) = \Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_4(\mathcal{A})] \ .$$

We now describe the game chain. In game $\mathbf{G}_2$, each call to $E(K^i_{\mathsf{out}}, \cdot)$, where K^i_{out} is the blockcipher subkey of the master key K_i, is replaced by a corresponding call to a truly random function $f_i \leftarrow\!\!\$\ \mathsf{Func}(\{0,1\}^n, \{0,1\}^n)$. These functions are lazily implemented by maintaining a table Tbl of the defined points. To bound the gap between $\mathbf{G}_1$ and $\mathbf{G}_2$, we construct the following adversary $\mathcal{B}$ attacking the PRF security E. Adversary $\mathcal{B}$ runs $\mathcal{A}$ and simulates game $\mathbf{G}_1$, but each call to $E(K_i, \cdot)$ is replaced by a corresponding query to the oracle $\textsc{Eval}(i, \cdot)$ of $\mathcal{B}$. Then $\mathcal{B}$'s real world

Game $\mathbf{G}_1(\mathcal{A})$	Games $\mathbf{G}_2(\mathcal{A})$, $\mathbf{G}_3(\mathcal{A})$
$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Eval}}$ Return $(b' = 1)$	$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Eval}}$ Return $(b' = 1)$
New() $v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^{k+n}$	New() $v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^{k+n}$
Eval$(i, (N, A, M))$ $K_{\mathsf{in}} \| K_{\mathsf{out}} \leftarrow K_i$ $X \leftarrow N \boxplus G(K_{\mathsf{in}}, M, A)$ $Y \leftarrow E(K_{\mathsf{out}}, X) \oplus X$ Return Y	Eval$(i, (N, A, M))$ $K_{\mathsf{in}} \| K_{\mathsf{out}} \leftarrow K_i$ $X \leftarrow N \boxplus G(K_{\mathsf{in}}, M, A)$; $V \leftarrow\!\!\$\ \{0,1\}^n$ If $\mathsf{Tbl}[i, X] \neq \perp$ then $\mathsf{bad} \leftarrow \mathsf{true}$; $V \leftarrow \mathsf{Tbl}[i, X]$ $\mathsf{Tbl}[i, X] \leftarrow V$; $Y \leftarrow V \oplus X$; Return Y

Game $\mathbf{G}_4(\mathcal{A})$	Enc$(i, (N, A, M))$
$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Eval}}$; Return $(b' = 1)$	If $i \notin \{1, \ldots, v\}$ return $\perp$ $Y \leftarrow\!\!\$\ \{0,1\}^n$ Return Y
New() $v \leftarrow v + 1$	

Figure 23: Games $\mathbf{G}_1$–$\mathbf{G}_4$ in the proof of Proposition 6.4. Game $\mathbf{G}_2$ contains the corresponding highlighted code, but game $\mathbf{G}_3$ does not.

corresponds to game $\mathbf{G}_1$, and its ideal world corresponds to game $\mathbf{G}_2$, and thus

$$\mathbf{Adv}^{\text{prf}}_E(\mathcal{B}) = \Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_2(\mathcal{A})] \ .$$

Adversary $\mathcal{B}$ therefore makes at most q queries, with at most B queries per user. From the Multi-user PRP/PRF Switching Lemma,

$$\mathbf{Adv}^{\text{prf}}_E(\mathcal{B}) \leq \mathbf{Adv}^{\text{prp}}_E(\mathcal{B}) + \frac{Bq}{2^n} \ .$$

In game $\mathbf{G}_2$, if there are two encryption queries on the same user i that make the same call to f_i then the two answers are the same. In contrast, in game $\mathbf{G}_3$, the two answers are independent. The two games are identical until the flag bad is set, and thus from the Fundamental Lemma of Game Playing [11],

$$\Pr[\mathbf{G}_2(\mathcal{A})] - \Pr[\mathbf{G}_3(\mathcal{A})] \leq \Pr[\mathbf{G}_3(\mathcal{A}) \text{ sets } \mathsf{bad}] \ .$$

We now bound the chance that game $\mathbf{G}_3$ sets bad, for a computationally unbounded adversary. Note that what the adversary receives are simply independent, uniformly random n-bit strings. Moreover, these strings are also independent of the event that game $\mathbf{G}_3$ sets bad. Since we consider a computationally unbounded adversary, without loss of generality, assume that the adversary is deterministic and non-adaptive (meaning all of its queries are created before seeing the outputs). Consider two queries $(i, (N, A, M))$ and $(i, (N^*, M^*, A^*))$ of the same user i. These queries make the same call to f_i if and only if

$$G(K^i_{\mathsf{in}}, A, M) \boxplus N = G(K^i_{\mathsf{in}}, A^*, M^*) \boxplus N^* \ .$$

In other words, $G(K^i_{\mathsf{in}}, A, M) \oplus G(K^i_{\mathsf{in}}, A^*, M^*)$ is either $(N \oplus N^*) \| 0^{n-r-1} \| 1$ or $(N \oplus N^*) \| 0^{n-r}$. We consider the following cases.

Case 1: $(A, M) \neq (A^*, M^*)$. Since G is c-AXU, the chance that the random variable

$$G(K^i_{\mathsf{in}}, A, M) \oplus G(K^i_{\mathsf{in}}, A^*, M^*)$$

is one of the two values above is at most

$$\frac{2c \cdot \max\{|A|_n + |M|_n, |A^*|_n + |M^*|_n\}}{2^n} \leq \frac{2c \cdot (|A|_n + |M|_n + |A^*|_n + |M^*|_n)}{2^n} \ .$$

Case 2: $(A, M) = (A^*, M^*)$, and thus we must have $N \neq N^*$. This case $G(K^i_{\mathsf{in}}, A, M) \oplus G(K^i_{\mathsf{in}}, A^*, M^*) = 0^n$, but the two values above are non-zero.

Summing over every pair of queries on the same user,

$$\Pr[\mathbf{G}_3(\mathcal{A}) \text{ sets } \mathsf{bad}] \leq \frac{2cqB}{2^n} \ .$$

Game $\mathbf{G}_4$ is a simplification of game $\mathbf{G}_3$, and thus

$$\Pr[\mathbf{G}_4(\mathcal{A})] = \Pr[\mathbf{G}_3(\mathcal{A})] \ .$$

Summing up,

$$\begin{aligned} \mathbf{Adv}_{\mathsf{GMAC2}[E,G]}(\mathcal{A}) &= \Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_4(\mathcal{A})] \\ &= \sum_{i=1}^{3} \Pr[\mathbf{G}_i(\mathcal{A})] - \Pr[\mathbf{G}_{i+1}(\mathcal{A})] \leq \mathbf{Adv}^{\mathrm{prp}}_E(\mathcal{B}) + \frac{(2c+1)qB}{2^n} \ . \end{aligned}$$

Q Proof of Theorem 6.5

Define H_0 via $H_0(K, N) = Y_1 \cdots Y_{2k/n}$, where $Y_i = E(K, [i]_{n-r})[1 : n/2]$. Let H_1 be the Davies-Meyer construction $\mathsf{DM}[E]$. Let h be the cascade of H_0 and H_1 as illustrated in Fig. 7. Below, we will construct an adversary $\mathcal{B}$ breaking the s-way multi-collision resistance of h.

Suppose that $\mathcal{A}$ outputs $\big((N_1, K_1, M_1, A_1), \ldots, (N_s, K_s, M_s, A_s)\big)$. Let $T_i \| C_i$ be the ciphertext of (N_i, A_i, M_i) under SE with key K_i, let L_i be the corresponding subkey for the blockcipher, and suppose that $T_i = \mathsf{DM}[E](L_i, X_i)$. Note that $L_i = H_0(K_i, N_i)$ and $T_i \leftarrow H_1(L_i, X_i)$. In other words, $T_i = h(K_i, N_i, X_i)$. If $\mathcal{A}$ wins then we must have $T_1 = \cdots = T_s$, creating an s-way multi-collision on h. We will use this observation to construct $\mathcal{B}$.

Adversary $\mathcal{B}$ runs $\big((N_1, K_1, M_1, A_1), \ldots, (N_s, K_s, M_s, A_s)\big) \leftarrow\!\!\$\ \mathcal{A}$. Then for each $i \leq s$, it computes the internal X_i that is used to derive the tag T_i under the Davies-Meyer construction. Finally, $\mathcal{B}$ outputs $\big((K_1, N_1, X_1), \ldots, (K_s, N_s, X_s)\big)$. This output is legitimate since $K_1, \ldots, K_s$ are distinct. To obtain $X_1, \ldots, X_s$, adversary $\mathcal{B}$ has to make $2s$ additional blockcipher calls to get the s hash keys. If $\mathcal{A}$ wins then $\mathcal{B}$ successfully creates an s-way multi-collision on h, and thus

$$\mathbf{Adv}^{\text{cmt-1}}_{\mathsf{SE},s}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{coll}}_{h,s}(\mathcal{B}) \ .$$

From Proposition 4.5, one can construct adversaries $\mathcal{D}_0$ and $\mathcal{D}_1$ such that

$$\mathbf{Adv}^{\mathrm{coll}}_{h,s}(\mathcal{B}) \leq \max\big\{\mathbf{Adv}^{\mathrm{coll}}_{H_0,t}(\mathcal{D}_0), \mathbf{Adv}^{\mathrm{coll}}_{H_1,t}(\mathcal{D}_1)\big\} \ .$$

Also, each of $\mathcal{D}_0$ and $\mathcal{D}_1$ runs $\mathcal{B}$ and then calls H_0 on s inputs. Note that an s-way multi-collision on H_0 is also a multi-collision on $\mathsf{ITP}[E, r, n]$ with $\mathsf{pad}(N, j) = N \| [j+2]_{n-r}$. Hence

$$\mathbf{Adv}^{\mathrm{coll}}_{H_0,t}(\mathcal{D}_0) \leq \mathbf{Adv}^{\mathrm{coll}}_{\mathsf{ITP}[E,r,n],t}(\mathcal{D}_0) \ .$$

Summing up,

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

Since a call to H_0 leads to at most 4 blockcipher calls, each $\mathcal{D}_i$ effectively runs $\mathcal{A}$ and then makes at most $6s$ other blockcipher calls.

Game $\mathbf{G}_1(\mathcal{A})$	Game $\mathbf{G}_2(\mathcal{A})$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Enc},\text{Vf}}$; return $(b'=1)$	$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Enc},\text{Vf}}$; return $(b'=1)$
New()	New()
$v \leftarrow v+1$	$v \leftarrow v+1$
$K_v \leftarrow\!\!\$\ \{0,1\}^k$	For $N \in \mathcal{N}$ do
	$K_{\mathsf{in}}^{v,N} \leftarrow\!\!\$\ \{0,1\}^n$; $K_{\mathsf{out}}^{v,N} \leftarrow\!\!\$\ \{0,1\}^k$
Enc(i,N,A,M)	Enc(i,N,A,M)
$K_{\mathsf{in}} \| K_{\mathsf{out}} \leftarrow \mathsf{KD1}[E,k+n](K_i,N)$	$K_{\mathsf{in}} \leftarrow K_{\mathsf{in}}^{i,N}$; $K_{\mathsf{out}} \leftarrow K_{\mathsf{out}}^{i,N}$
$\mathsf{IV} \leftarrow \mathsf{GMAC2}[E,H](K_{\mathsf{in}} \| K_{\mathsf{out}},N,A,M)$	$\mathsf{IV} \leftarrow \mathsf{GMAC2}[E,H](K_{\mathsf{in}} \| K_{\mathsf{out}},N,A,M)$
$C \leftarrow \mathsf{CTR}[E,\mathsf{add}].\mathsf{Enc}(K_{\mathsf{out}},M;\mathsf{IV})$	$C \leftarrow \mathsf{CTR}[E,\mathsf{add}].\mathsf{Enc}(K_{\mathsf{out}},M;\mathsf{IV})$
Return C	Return C
Vf(i,N,A,C)	Vf(i,N,A,C)
If $i \notin \{1,\ldots,v\}$ return $\perp$	If $i \notin \{1,\ldots,v\}$ return $\perp$
$K_{\mathsf{in}} \| K_{\mathsf{out}} \leftarrow \mathsf{KD1}[E,k+n](K,N)$	$K_{\mathsf{in}} \leftarrow K_{\mathsf{in}}^{i,N}$; $K_{\mathsf{out}} \leftarrow K_{\mathsf{out}}^{i,N}$
$M \leftarrow \mathsf{CTR}[E,\mathsf{add}].\mathsf{Dec}(K_{\mathsf{out}},C)$	$M \leftarrow \mathsf{CTR}[E,\mathsf{add}].\mathsf{Dec}(K_{\mathsf{out}},C)$
$\mathsf{IV} \leftarrow \mathsf{GMAC2}[E,H](K_{\mathsf{in}} \| K_{\mathsf{out}},N,A,M)$	$\mathsf{IV} \leftarrow \mathsf{GMAC2}[E,H](K_{\mathsf{in}} \| K_{\mathsf{out}},N,A,M)$
If $\mathsf{IV} \neq C[1:n]$ then return $\perp$	If $\mathsf{IV} \neq C[1:n]$ then return $\perp$
Return M	Return M

Figure 24: Games $\mathbf{G}_1$ and $\mathbf{G}_2$ in the proof of Theorem 6.6.

R Proof of Theorem 6.6

From Proposition 4.2, without loss of generality, we can assume that $\mathcal{A}$ is orderly. The difference in the advantage of $\mathcal{A}$ is at most $q/2^n$; we will account for this difference in the final bound.

Let $\mathsf{CTR}[f,\mathsf{add}](M)$ be a variant of $\mathsf{CTR}[E,\mathsf{add}](K,M)$ in which each call to $E_K(\cdot)$ is replaced by a corresponding call to $f(\cdot)$. Let $\mathsf{CTR}[\mathsf{Perm}(\{0,1\}^n),\mathsf{add}]$ be the idealized version of $\mathsf{CTR}[E,\mathsf{add}]$ in which each call to $E(K_i,\cdot)$ is replaced by a corresponding call to $\pi_i \leftarrow\!\!\$\ \mathsf{Perm}(\{0,1\}^n)$. Let $\mathsf{GMAC2}[E,H](K_{\mathsf{in}},N,A,M)$ be a variant of $\mathsf{GMAC2}[E,H](K_{\mathsf{in}} \| K_{\mathsf{out}},N,A,M)$ in which each call to $E(K_{\mathsf{out}},\cdot)$ is replaced by a corresponding call to $f(\cdot)$. Let $\mathsf{GMAC2}[\mathsf{Perm}(\{0,1\}^n),H]$ be the idealized version of $\mathsf{GMAC2}[E,H]$ in which each call to $E(K_{\mathsf{out}}^i,\cdot)$ is replaced by a corresponding call to $\pi_i \leftarrow\!\!\$\ \mathsf{Perm}(\{0,1\}^n)$.

Consider games $\mathbf{G}_1$–$\mathbf{G}_8$ in Fig. 24, Fig. 25, and Fig. 26. Game $\mathbf{G}_1$ corresponds to game $\mathbf{G}^{\text{real}}_{\mathsf{CAU\text{-}SIV\text{-}C1}[E,H,\mathsf{add}]}(\mathcal{A})$, and game $\mathbf{G}_8$ to game $\mathbf{G}^{\text{rand}}_{\mathsf{CAU\text{-}SIV\text{-}C1}[E,H,\mathsf{add}]}(\mathcal{A})$. Then

$$\mathbf{Adv}^{\text{mrae}}_{\mathsf{CAU\text{-}SIV\text{-}C1}[E,H,\mathsf{add}]}(\mathcal{A}) = \Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_8(\mathcal{A})] \ .$$

We now describe the game chain. In game $\mathbf{G}_2$, instead of using $\mathsf{KD1}$ to derive subkeys for each (i,N), we pick the subkeys uniformly at random. To bound the gap between $\mathbf{G}_1$ and $\mathbf{G}_2$, we construct an adversary $\mathcal{D}$ attacking the PRF security of $\mathsf{KD1}$ as follows. Adversary $\mathcal{D}$ runs $\mathcal{A}$ and simulates game $\mathbf{G}_1$, but each call to $\mathsf{KD1}[E,k+n](K_i,\cdot)$ is replaced by a corresponding call to Eval$(i,\cdot)$. Then

$$\mathbf{Adv}^{\text{prf}}_{\mathsf{KD1}[E,k+n]}(\mathcal{D}) = \Pr[\mathbf{G}_1(\mathcal{A})] - \Pr[\mathbf{G}_2(\mathcal{A})] \ .$$

Adversary $\mathcal{D}$ makes at most q queries with at most D queries per user. Using Lemma 6.2 with $t = 1 + \lceil k/n \rceil \leq 3$, one can construct an adversary $\mathcal{B}_0$ making at most $6q$ queries with at most $6D$

Game $\mathbf{G}_3(\mathcal{A})$	Game $\mathbf{G}_4(\mathcal{A})$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Enc},\text{Vf}}$; return $(b'=1)$	$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Enc},\text{Vf}}$; return $(b'=1)$
New()	New()
$v \leftarrow v+1$	$v \leftarrow v+1$
For $N \in \mathcal{N}$ do	For $N \in \mathcal{N}$ do
$\quad K_{\mathsf{in}}^{v,N} \leftarrow\!\!\$\ \{0,1\}^n$	$\quad K_{\mathsf{in}}^{v,N} \leftarrow\!\!\$\ \{0,1\}^n$
$\quad f_{i,N} \leftarrow\!\!\$\ \mathsf{Func}(\{0,1\}^n,\{0,1\}^n)$	$\quad f_{i,N} \leftarrow\!\!\$\ \mathsf{Func}(\{0,1\}^n,\{0,1\}^n)$
	$\quad g_{i,N} \leftarrow\!\!\$\ \mathsf{Func}(\{0,1\}^n,\{0,1\}^n)$
Enc(i,N,A,M)	Enc(i,N,A,M)
$\mathsf{IV} \leftarrow \mathsf{GMAC2}[f_{i,N},H](K_{\mathsf{in}}^{i,N},N,A,M)$	$\mathsf{IV} \leftarrow \mathsf{GMAC2}[f_{i,N},H](K_{\mathsf{in}}^{i,N},N,A,M)$
$C \leftarrow \mathsf{CTR}[f_{i,N},\mathsf{add}].\mathsf{Enc}(M;\mathsf{IV})$	$C \leftarrow \mathsf{CTR}[g_{i,N},\mathsf{add}].\mathsf{Enc}(M;\mathsf{IV})$
Return C	Return C
Vf(i,N,A,C)	Vf(i,N,A,C)
If $i \notin \{1,\ldots,v\}$ return $\perp$	If $i \notin \{1,\ldots,v\}$ return $\perp$
$M \leftarrow \mathsf{CTR}[f_{i,N},\mathsf{add}].\mathsf{Dec}(C)$	$M \leftarrow \mathsf{CTR}[g_{i,N},\mathsf{add}].\mathsf{Dec}(C)$
$\mathsf{IV} \leftarrow \mathsf{GMAC2}[f_{i,N},H](K_{\mathsf{in}}^{i,N},N,A,M)$	$\mathsf{IV} \leftarrow \mathsf{GMAC2}[f_{i,N},H](K_{\mathsf{in}}^{i,N},N,A,M)$
If $\mathsf{IV} = C[1:n]$ then return $\perp$	If $\mathsf{IV} = C[1:n]$ then return $\perp$
Return M	Return M

Figure 25: Games $\mathbf{G}_3$–$\mathbf{G}_4$ in the proof of Theorem 6.6.

queries per user such that

$$\mathbf{Adv}^{\text{prf}}_{\mathsf{KD1}[E,k+n]}(\mathcal{D}) \le \mathbf{Adv}^{\text{prp}}_{E}(\mathcal{B}_0) + \frac{6\sqrt{nDq}}{2^{3n/4}} \ .$$

In game $\mathbf{G}_3$, each call to $E(K_{\mathsf{out}}^{i,N},\cdot)$ is replaced by a corresponding call to a truly random function $f_{i,N} \leftarrow\!\!\$\ \mathsf{Func}(\{0,1\}^n,\{0,1\}^n)$. To bound the gap between $\mathbf{G}_2$ and $\mathbf{G}_3$, we construct an adversary $\mathcal{B}_1$ attacking the PRF security of E as follows. It runs $\mathcal{A}$ and simulates game $\mathbf{G}_2$, and keeps an array Users that translates a pair (i,N) of $\mathcal{A}$ to its user index. In particular, in each encryption/verification query of $\mathcal{A}$ to user i with nonce N, if $\mathsf{Users}[i,N]$ is not defined then $\mathcal{B}_1$ creates a new user v via the New oracle and stores $\mathsf{Users}[i,N] \leftarrow v$, otherwise $\mathcal{B}_1$ retrieves $v \leftarrow \mathsf{Users}[i,N]$. Moreover, each call to $E(K_{\mathsf{out}}^{i,N},\cdot)$ is replaced by a corresponding call to Eval$(v,\cdot)$. Then

$$\mathbf{Adv}^{\text{prf}}_{E}(\mathcal{B}_1) = \Pr[\mathbf{G}_2(\mathcal{A})] - \Pr[\mathbf{G}_3(\mathcal{A})] \ .$$

Adversary $\mathcal{B}_1$ makes at most $\sigma+q$ queries, with at most $2B$ queries per user. From the Multi-user PRP/PRF Switching Lemma,

$$\mathbf{Adv}^{\text{prf}}_{E}(\mathcal{B}_1) \le \mathbf{Adv}^{\text{prp}}_{E}(\mathcal{B}_1) + \frac{2B(\sigma+q)}{2^n} \ .$$

We now have two adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$ attacking the PRP security of E. We can use the standard hybrid argument to create a unified adversary $\mathcal{B}$ attacking E as follows. Adversary $\mathcal{B}$ tosses a fair coin $b \leftarrow\!\!\$\ \{0,1\}$, and then runs $\mathcal{B}_b$. Then

$$\mathbf{Adv}^{\text{prp}}_{E}(\mathcal{B}) = \frac{1}{2}\Big(\mathbf{Adv}^{\text{prp}}_{E}(\mathcal{B}_0) + \mathbf{Adv}^{\text{prp}}_{E}(\mathcal{B}_1)\Big) \ .$$

Back to the game chain, recall that in game $\mathbf{G}_3$, each encryption/verification query on user i and nonce N results in running $\mathsf{GMAC2}$ and CTR on the same random function $f_{i,N}$. However, due to a domain separation, $\mathsf{GMAC2}$ runs $f_{i,N}$ on inputs starting with 0, whereas CTR runs $f_{i,N}$ on inputs starting with 1. In game $\mathbf{G}_4$, we instead run $\mathsf{GMAC2}$ and CTR on independent random functions

Game $\mathbf{G}_5(\mathcal{A})$	Game $\mathbf{G}_6(\mathcal{A})$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{NEW},\text{ENC},\text{VF}}$; return $(b'=1)$	$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{NEW},\text{ENC},\text{VF}}$; return $(b'=1)$
NEW()	NEW()
$v \leftarrow v+1$	$v \leftarrow v+1$
For $N \in \mathcal{N}$ do	For $N \in \mathcal{N}$ do
$K_{\mathsf{in}}^{v,N} \leftarrow\!\!\$\ \{0,1\}^n$	$K_{\mathsf{in}}^{v,N} \leftarrow\!\!\$\ \{0,1\}^n$
$\pi_{v,N} \leftarrow\!\!\$\ \mathsf{Perm}(\{0,1\}^n)$; $\pi^*_{v,N} \leftarrow\!\!\$\ \mathsf{Perm}(\{0,1\}^n)$	$\pi_{v,N} \leftarrow\!\!\$\ \mathsf{Perm}(\{0,1\}^n)$; $\pi^*_{v,N} \leftarrow\!\!\$\ \mathsf{Perm}(\{0,1\}^n)$
ENC(i,N,A,M)	ENC(i,N,A,M)
$\mathsf{IV} \leftarrow \mathsf{GMAC2}[\pi_{i,N},H](K_{\mathsf{in}}^{i,N},N,A,M)$	$\mathsf{IV} \leftarrow\!\!\$\ \{0,1\}^n$; $\mathsf{Tbl}[i,N,A,M] \leftarrow \mathsf{IV}$
$C \leftarrow \mathsf{CTR}[\pi^*_{i,N},\mathsf{add}].\mathsf{Enc}(M;\mathsf{IV})$	$C \leftarrow \mathsf{CTR}[\pi^*_{i,N},\mathsf{add}].\mathsf{Enc}(M;\mathsf{IV})$
Return C	Return C
VF(i,N,A,C)	VF(i,N,A,C)
If $i \notin \{1,\ldots,v\}$ return $\perp$	If $i \notin \{1,\ldots,v\}$ return $\perp$
$M \leftarrow \mathsf{CTR}[\pi^*_{i,N},\mathsf{add}].\mathsf{Dec}(C)$	$M \leftarrow \mathsf{CTR}[\pi^*_{i,N},\mathsf{add}].\mathsf{Dec}(C)$
$\mathsf{IV} \leftarrow \mathsf{GMAC2}[\pi_{i,N},H](K_{\mathsf{in}}^{i,N},N,A,M)$	$\mathsf{IV} \leftarrow\!\!\$\ \{0,1\}^n$
If $\mathsf{IV} = C[1:n]$ then return $\perp$	If $\mathsf{Tbl}[i,N,A,M] \neq \perp$ then
Return M	$\mathsf{IV} \leftarrow \mathsf{Tbl}[i,N,A,M]$
	$\mathsf{Tbl}[i,N,A,M] \leftarrow \mathsf{IV}$
	If $\mathsf{IV} = C[1:n]$ then return $\perp$
	Return M

Game $\mathbf{G}_7(\mathcal{A})$	Game $\mathbf{G}_8(\mathcal{A})$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{NEW},\text{ENC},\text{VF}}$; return $(b'=1)$	$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{NEW},\text{ENC},\text{VF}}$; return $(b'=1)$
NEW()	NEW()
$v \leftarrow v+1$	$v \leftarrow v+1$
ENC(i,N,A,M)	ENC(i,N,A,M)
If $i \notin \{1,\ldots,v\}$ return $\perp$	If $i \notin \{1,\ldots,v\}$ return $\perp$
$\mathsf{IV} \leftarrow\!\!\$\ \{0,1\}^n$	$C \leftarrow\!\!\$\ \{0,1\}^{\lvert M\rvert+n}$
$C \leftarrow \mathsf{CTR}[\pi^*_{i,N},\mathsf{add}].\mathsf{Enc}(M;\mathsf{IV})$	Return C
Return C	
VF(i,N,A,C)	VF(i,N,A,C)
If $i \notin \{1,\ldots,v\}$ return $\perp$	If $i \notin \{1,\ldots,v\}$ return $\perp$
Return false	Return false

Figure 26: Games $\mathbf{G}_5$–$\mathbf{G}_8$ in the proof of Theorem 6.6.

$f_{i,N}, g_{i,N} \leftarrow\!\!\$\ \mathsf{Func}(\{0,1\}^n,\{0,1\}^n)$. Thanks to the domain separation,

$$\Pr[\mathbf{G}_4(\mathcal{A})] = \Pr[\mathbf{G}_3(\mathcal{A})] \ .$$

In game $\mathbf{G}_5$, instead of using random functions $f_{i,N}, g_{i,N} \leftarrow\!\!\$\ \mathsf{Func}(\{0,1\}^n,\{0,1\}^n)$, we use random permutations $\pi_{i,N}, \pi^*_{i,N} \leftarrow\!\!\$\ \mathsf{Perm}(\{0,1\}^n)$. We make at most $\sigma+q$ calls to these permutations, with at most $2B$ calls per user. From the Multi-user PRP/PRF Switching Lemma,

$$\Pr[\mathbf{G}_4(\mathcal{A})] - \Pr[\mathbf{G}_5(\mathcal{A})] \leq \frac{2B(\sigma+q)}{2^n} \ .$$

In game $\mathbf{G}_6$, instead of running GMAC2 to generate IVs, we use truly random functions that are lazily implemented via keeping a table Tbl of defined points. To bound the gap between $\mathbf{G}_5$ and $\mathbf{G}_6$, we will generate an adversary $\mathcal{B}_2$ attacking the PRF security $\mathsf{GMAC2}[\mathsf{Perm}(\{0,1\}^n),H]$. It

runs $\mathcal{A}$ and simulates game $\mathbf{G}_5$, and keeps an array Users that translates a pair (i, N) of $\mathcal{A}$ to its user v like $\mathcal{B}_1$. Moreover, each call to $\mathsf{GMAC2}(K_{\mathsf{in}}^{i,N}, N, A, M)$ is replaced by a corresponding call to $\textsc{Eval}(v, (N, A, M))$. Then

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{GMAC2}[\mathsf{Perm}(\{0,1\}^n),H]}(\mathcal{B}_2) = \Pr[\mathbf{G}_5(\mathcal{A})] - \Pr[\mathbf{G}_6(\mathcal{A})] \ .$$

Adversary $\mathcal{B}_2$ makes at most q queries with at most B blocks per user. Then from Proposition 6.4,

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{GMAC2}[\mathsf{Perm}(\{0,1\}^n),H]}(\mathcal{B}_2) \leq \frac{(2c+1)qB}{2^n} \ .$$

In game $\mathbf{G}_7$, each verification query will return false. To bound the gap between $\mathbf{G}_6$ and $\mathbf{G}_7$, consider a verification query $\textsc{Vf}(i, N, A, C)$ in game $\mathbf{G}_6$. Let IV^* be the initialization vector in C, and let M be the tentative decrypted message by running CTR on C. This query can return true if and only if IV^* is the same as the targeting IV.

- If there is no prior query $\textsc{Enc}(i, N, A, M)$ then the chance that $\mathsf{IV}^* = \mathsf{IV}$ is at most $1/2^n$.
- If there is a prior query $C' \leftarrow \textsc{Enc}(i, N, A, M)$ then IV is the initialization vector of C'. Moreover, due to the restriction on $\mathcal{A}$, we must have $C \neq C'$. Since $C' \neq C$ and decrypting them under CTR yields the same message, their initialization vectors IV and IV^* must be different.

Hence in all cases, the chance that this verification query can return true is at most $1/2^n$. Summing this over at most q verification queries,

$$\Pr[\mathbf{G}_6(\mathcal{A})] - \Pr[\mathbf{G}_7(\mathcal{A})] \leq \frac{q}{2^n} \ .$$

In game $\mathbf{G}_8$, for each query $\textsc{Enc}(i, N, A, M)$, we simply return a fresh random answer $C \leftarrow\!\!\$\ \{0,1\}^{|M|+n}$. To bound the gap between $\mathbf{G}_7$ and $\mathbf{G}_8$, we will generate an adversary $\mathcal{B}_3$ attacking the chosen-plaintext security of $\mathsf{CTR}[\mathsf{Perm}(\{0,1\}^n), \mathsf{add}]$. Adversary $\mathcal{B}_3$ runs $\mathcal{A}$ and simulates game $\mathbf{G}_7$, and keeps an array Users that translates a pair (i, N) of $\mathcal{A}$ to its user v like $\mathcal{B}_1$. In addition, each call to $\mathsf{CTR}[\pi^*_{i,N}, \mathsf{add}].\mathsf{Enc}$ is replaced by a corresponding call to $\textsc{Enc}(v, \cdot)$. Then

$$\mathbf{Adv}^{\mathrm{ind}}_{\mathsf{CTR}[\mathsf{Perm}(\{0,1\}^n),\mathsf{add}]}(\mathcal{B}_3) = \Pr[\mathbf{G}_7(\mathcal{A})] - \Pr[\mathbf{G}_8(\mathcal{A})] \ .$$

Adversary $\mathcal{B}_3$'s queries consists of at most σ blocks with at most B blocks per user. Thus from Proposition 6.3,

$$\mathbf{Adv}^{\mathrm{ind}}_{\mathsf{CTR}[\mathsf{Perm}(\{0,1\}^n),\mathsf{add}]}(\mathcal{B}_3) \leq \frac{3\sigma B}{2^n} \ .$$

Summing up,

$$\mathbf{Adv}^{\mathrm{mrae}}_{\mathsf{CAU\text{-}SIV\text{-}C1}[E,H,\mathsf{add}]}(\mathcal{A}) \leq 2 \cdot \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{6\sqrt{nDq}}{2^{3n/4}} + \frac{7\sigma B + (2c+5)qB + q}{2^n} \ .$$

If we remove the restriction that the adversary is orderly,

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{mrae}}_{\mathsf{CAU\text{-}SIV\text{-}C1}[E,H,\mathsf{add}]}(\mathcal{A}) &\leq 2 \cdot \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{6\sqrt{nDq}}{2^{3n/4}} + \frac{7\sigma B + (2c+5)qB + 2q}{2^n} \\
&\leq 2 \cdot \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{6\sqrt{nDq}}{2^{3n/4}} + \frac{7\sigma B + (2c+7)qB}{2^n} \ .
\end{aligned}$$

S Proof of Theorem 7.2

a) We construct the adversary $\mathcal{B}_0$ as follows. It runs

$$\big((N_1, K_1, M_1, A_1), \ldots, (N_s, K_s, M_s, A_s)\big) \leftarrow\!\!\$\ \mathcal{A}_0 \ .$$

Adversary $\mathcal{B}_0$ then outputs $(K_1, \ldots, K_s)$. Let $P_i \| C_i$ be the ciphertext by encrypting (N_i, A_i, M_i) under $\mathsf{UtC}[\mathsf{F}, \mathsf{SE}]$ with key K_i, where $(P_i, L_i) \leftarrow \mathsf{F}(K_i, N_i)$. If $\mathcal{A}_0$ wins then we must have $P_1 = \cdots =$

Game $\mathbf{G}_0(\mathcal{A}_1)$	Game $\mathbf{G}_1(\mathcal{A}_1)$
$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}_1^{\text{NEW},\text{ENC},\text{VF}}$ Return $(b' = 1)$	$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}_1^{\text{NEW},\text{ENC},\text{VF}}$ Return $(b' = 1)$
NEW() $v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k$	NEW() $v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k$
ENC(i, N, A, M) If $i \notin \{1, \ldots, v\}$ return $\bot$ $(P, L) \leftarrow \mathsf{F}(K_i, N)$ $C \leftarrow \mathsf{SE.Enc}(L, N, A, M)$ Return $P \| C$	ENC(i, N, A, M) If $i \notin \{1, \ldots, v\}$ return $\bot$ $P \leftarrow\!\!\$\ \{0,1\}^\ell$; $L \leftarrow\!\!\$\ \{0,1\}^k$ If $\mathsf{Keys}[i, N] \neq \bot$ then $(P, L) \leftarrow \mathsf{Keys}[i, N]$ $\mathsf{Keys}[i, N] \leftarrow (P, L)$; $C \leftarrow \mathsf{SE.Enc}(L, N, A, M)$ Return $P \| C$
VF$(i, N, A, P^* \| C)$ If $i \notin \{1, \ldots, v\}$ return $\bot$ $(P, L) \leftarrow \mathsf{F}(K_i, N)$ If $P^* \neq P$ then return false $V \leftarrow \mathsf{SE.Dec}(L, N, A, C)$ Return $(V \neq \bot)$	VF$(i, N, A, P^* \| C)$ If $i \notin \{1, \ldots, v\}$ return $\bot$ $P \leftarrow\!\!\$\ \{0,1\}^\ell$; $L \leftarrow\!\!\$\ \{0,1\}^k$ If $\mathsf{Keys}[i, N] \neq \bot$ then $(P, L) \leftarrow \mathsf{Keys}[i, N]$ Else $\mathsf{Keys}[i, N] \leftarrow (P, L)$ If $P^* \neq P$ then return false $V \leftarrow \mathsf{SE.Dec}(L, N, A, C)$; Return $(V \neq \bot)$

Game $\mathbf{G}_2(\mathcal{A}_1)$	NEW()
$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}_1^{\text{NEW},\text{ENC},\text{VF}}$ Return $(b' = 1)$	$v \leftarrow v + 1$
ENC(i, N, A, M) If $i \notin \{1, \ldots, v\}$ return $\bot$ $C \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.len}(\lvert M \rvert)}$ $P \leftarrow\!\!\$\ \{0,1\}^\ell$; Return $P \| C$	VF$(i, N, A, P^* \| C)$ If $i \notin \{1, \ldots, v\}$ return $\bot$ Return false

Figure 27: Games $\mathbf{G}_0$–$\mathbf{G}_2$ in the proof of Theorem 7.2.

P_s, meaning that $\mathcal{B}_0$ also breaks the binding security of F. Hence $\mathbf{Adv}^{\text{cmt-1}}_{\mathsf{UtC}[\mathsf{F},\mathsf{SE}],s}(\mathcal{A}_0) \leq \mathbf{Adv}^{\text{bind}}_{\mathsf{F},s}(\mathcal{B}_0)$ as claimed.

b) Consider games $\mathbf{G}_0$–$\mathbf{G}_2$ in Fig. 27. Game $\mathbf{G}_0$ corresponds to $\mathbf{G}^{\text{real}}_{\mathsf{UtC}[\mathsf{F},\mathsf{SE}]}(\mathcal{A}_1)$, and game $\mathbf{G}_2$ to game $\mathbf{G}^{\text{rand}}_{\mathsf{UtC}[\mathsf{F},\mathsf{SE}]}(\mathcal{A}_1)$. Thus

$$\mathbf{Adv}^{\text{unae}}_{\mathsf{UtC}[\mathsf{F},\mathsf{SE}]}(\mathcal{A}_1) = \Pr[\mathbf{G}_0(\mathcal{A}_1)] - \Pr[\mathbf{G}_2(\mathcal{A}_1)] \ .$$

Below, we will describe the transition in the game chain.

In game $\mathbf{G}_1$, each call to $\mathsf{F}(K_i, \cdot)$ is replaced by a call to a corresponding truly random function $f_i \leftarrow\!\!\$\ \mathsf{Func}(\{0,1\}^r, \{0,1\}^\ell \times \{0,1\}^k)$; the latter is lazily implemented by keeping an array Keys of defined points. To bound the gap between $\mathbf{G}_0$ and $\mathbf{G}_1$, we construct an adversary $\mathcal{B}_1$ attacking the PRF security of F as follows. It runs $\mathcal{A}_1$ and simulates game $\mathbf{G}_0$, but each call to $\mathsf{F}(K_i, \cdot)$ will be replaced by a call to $\mathcal{B}_1$'s oracle EVAL$(i, \cdot)$. When $\mathcal{A}_1$ outputs its guess, $\mathcal{B}_1$ outputs the same guess. Then $\mathcal{B}_1$'s real world corresponds to game $\mathbf{G}_0$, and its ideal world corresponds to game $\mathbf{G}_1$. In other words,

$$\mathbf{Adv}^{\text{prf}}_{\mathsf{F}}(\mathcal{B}_1) = \Pr[\mathbf{G}_0(\mathcal{A}_1)] - \Pr[\mathbf{G}_1(\mathcal{A}_1)] \ .$$

To bound the gap between $\mathbf{G}_1$ and $\mathbf{G}_2$, we construct the following adversary $\mathcal{B}_2$. It runs $\mathcal{A}_1$ and simulates game $\mathbf{G}_1$, and keeps an array Users that translates a pair (i, N) of $\mathcal{A}_1$ to its user index. In particular, in each encryption/verification query of $\mathcal{A}_1$ to user i with nonce N, if $\mathsf{Users}[i, N]$ is not defined then $\mathcal{B}_2$ creates a new user v via the NEW oracle and stores $\mathsf{Users}[i, N] \leftarrow v$, otherwise $\mathcal{B}_2$ retrieves $v \leftarrow \mathsf{Users}[i, N]$. Moreover, for each encryption query (i, N, A, M) of $\mathcal{A}_2$, the call to $\mathsf{SE.Enc}$ in game $\mathbf{G}_1$ will be replaced by a query (v, N, A, M) to the ENC oracle of $\mathcal{B}_2$. Likewise, for each verification query (i, N, A, C) of $\mathcal{A}_2$, instead of calling $\mathsf{SE.Dec}$ and checking if the decrypted message is $\perp$ as in game $\mathbf{G}_1$, adversary $\mathcal{B}_2$ queries (v, N, A, C) to its VF oracle.

We note that $\mathcal{B}_2$ does not query $C \leftarrow \text{ENC}(v, N, A, M)$ first and then query $\text{VF}(v, N, A, C)$. Assume to the contrary that $\mathcal{B}_2$ violates this restriction. This can only happen if $\mathcal{A}_1$ first queries $P \| C \leftarrow \text{ENC}(i, N, A, M)$ and then queries $\text{VF}(i, N, A, P^* \| C)$ with $P^* \neq P$. However, in that case $\mathcal{B}_2$'s implementation of the verification oracle of $\mathcal{A}_1$ will first retrieve the same prior P, and then return false (due to the failure of the checking $P^* \neq P$) without calling $\mathsf{SE.Dec}$, leading to a contradiction.

Thus game $\mathbf{G}^{\text{real}}_{\mathsf{SE}}(\mathcal{B}_2)$ corresponds to game $\mathbf{G}_1$, and game $\mathbf{G}^{\text{rand}}_{\mathsf{SE}}(\mathcal{B}_2)$ corresponds to game $\mathbf{G}_2$. Hence

$$\mathbf{Adv}^{\text{unae}}_{\mathsf{SE}}(\mathcal{B}_2) = \Pr[\mathbf{G}_1(\mathcal{A}_1)] - \Pr[\mathbf{G}_2(\mathcal{A}_1)] \ .$$

Summing up,

$$\begin{aligned} \mathbf{Adv}^{\text{unae}}_{\mathsf{UtC}[\mathsf{F},\mathsf{SE}]}(\mathcal{A}_1) &= \Pr[\mathbf{G}_0(\mathcal{A}_1)] - \Pr[\mathbf{G}_2(\mathcal{A}_1)] \\ &= \sum_{i=0}^{1} \Pr[\mathbf{G}_i(\mathcal{A}_1)] - \Pr[\mathbf{G}_{i+1}(\mathcal{A}_1)] = \mathbf{Adv}^{\text{prf}}_{\mathsf{F}}(\mathcal{B}) + \mathbf{Adv}^{\text{unae}}_{\mathsf{SE}}(\mathcal{B}_2) \end{aligned}$$

as claimed.

T Proof of Theorem 7.3

a) Let $G : \{0,1\}^k \times \mathcal{N} \to \{0,1\}^\ell$ be the hash function that $G(K, N) = P$, where $(P, L) \leftarrow \mathsf{F}(K, N)$. Let h be the cascade of G and H as in Fig. 7. Below, we will construct an adversary $\mathcal{A}'$ attacking the multi-collision resistance of h.

Suppose that $\mathcal{A}_0$ outputs $\big((N_1, K_1, M_1, A_1), \ldots, (N_s, K_s, M_s, A_s)\big)$. Let $T_i \| C_i$ be the ciphertext when one encrypts (N_i, A_i, M_i) under $\mathsf{RtC}[\mathsf{F}, \mathsf{SE}, H]$ with key K_i. Let $(P_i, L_i) \leftarrow \mathsf{F}(K_i, N_i)$. Then $T_i = h(K_i, N_i, C_i[1:n])$. Note that if $\mathcal{A}_0$ wins then $T_1 = \cdots = T_s$, leading to an s-way multi-collision on h.

Adversary $\mathcal{A}'$ first runs $\big((N_1, K_1, M_1, A_1), \ldots, (N_s, K_s, M_s, A_s)\big) \leftarrow\!\!\$\ \mathcal{A}_0$. It then encrypts (N_1, M_1, A_1) via $\mathsf{RtC}[\mathsf{F}, \mathsf{SE}, H]$ under key K_1, obtaining the ciphertext $T_1 \| C_1$. It then outputs

$$\big((K_1, N_1, C_1[1:n]), \ldots, (K_s, N_s, C_1[1:n])\big) \ .$$

Note that the same C_1 is used in all s triples. This output is legitimate because $K_1, \ldots, K_s$ are distinct. If $\mathcal{A}_0$ wins then $C_1 = \cdots = C_s$, and thus $h(K_i, N_i, C_1[1:n]) = h(K_i, N_i, C_i[1:n]) = T_i$, and $\mathcal{A}'$ successfully creates an s-way multi-collision on h. Hence

$$\mathbf{Adv}^{\text{cmt-1}}_{\mathsf{RtC}[\mathsf{F},\mathsf{SE},H],s}(\mathcal{A}_0) \leq \mathbf{Adv}^{\text{coll}}_{h,s}(\mathcal{A}') \ .$$

From Proposition 4.5, one can construct adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$ such that

$$\mathbf{Adv}^{\text{coll}}_{h,s}(\mathcal{A}') \leq \max\left\{\mathbf{Adv}^{\text{coll}}_{G,t}(\mathcal{B}_0, t), \mathbf{Adv}^{\text{coll}}_{H,t}(\mathcal{B}_1)\right\} \ .$$

Note that attacking the multi-collision resistance of G is the same as attacking the binding security of F, and thus

$$\mathbf{Adv}^{\text{coll}}_{h,s}(\mathcal{A}') \leq \max\left\{\mathbf{Adv}^{\text{bind}}_{\mathsf{F},t}(\mathcal{B}_0), \mathbf{Adv}^{\text{coll}}_{H,t}(\mathcal{B}_1)\right\} \ .$$

Game $\mathbf{G}_0(\mathcal{A}_1)$	Game $\mathbf{G}_1(\mathcal{A}_1)$
$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}_1^{\text{New},\text{Enc},\text{Vf}}$	$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}_1^{\text{New},\text{Enc},\text{Vf}}$
Return $(b' = 1)$	Return $(b' = 1)$
New()	New()
$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k$	$v \leftarrow v + 1$
Enc(i, N, A, M)	Enc(i, N, A, M)
If $i \notin \{1, \ldots, v\}$ return $\perp$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$(P, L) \leftarrow \mathsf{F}(K_i, N)$	$P \leftarrow\!\!\$\ \{0,1\}^\ell$; $L \leftarrow\!\!\$\ \{0,1\}^\lambda$
$C \leftarrow \mathsf{SE.Enc}(L, N, A, M)$	If $\mathsf{Keys}[i, N] \neq \perp$ then $(P, L) \leftarrow \mathsf{Keys}[i, N]$
$T \leftarrow H(P, C[1:n])$	$\mathsf{Keys}[i, N] \leftarrow (P, L)$; $C \leftarrow \mathsf{SE.Enc}(L, N, A, M)$
Return $T \| C$	$T \leftarrow H(P, C[1:n])$; Return $T \| C$
Vf$(i, N, A, T \| C)$	Vf$(i, N, A, T \| C)$
If $i \notin \{1, \ldots, v\}$ return $\perp$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$(P, L) \leftarrow \mathsf{F}(K_i, N)$	$P \leftarrow\!\!\$\ \{0,1\}^\ell$; $L \leftarrow\!\!\$\ \{0,1\}^\lambda$
$T^* \leftarrow H(P, C[1:n])$	If $\mathsf{Keys}[i, N] \neq \perp$ then $(P, L) \leftarrow \mathsf{Keys}[i, N]$
If $T \neq T^*$ then return false	$\mathsf{Keys}[i, N] \leftarrow (P, L)$; $T^* \leftarrow H(P, C[1:n])$
$V \leftarrow \mathsf{SE.Dec}(L, N, A, C)$	If $T \neq T^*$ then return false
Return $(V \neq \perp)$	$V \leftarrow \mathsf{SE.Dec}(L, N, A, C)$; Return $(V \neq \perp)$

Figure 28: Games $\mathbf{G}_0$ and $\mathbf{G}_1$ in the proof of Theorem 7.3.

Summing up,

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

Each of $\mathcal{B}_0$ and $\mathcal{B}_1$ runs $\mathcal{A}'$ and then runs G on s inputs. Thus effectively, each of them runs $\mathcal{A}$ and then uses $\mathsf{RtC[F, SE}, H]$ to encrypt one of the s messages of $\mathcal{A}_0$, and then runs F on s inputs.

b) Consider games $\mathbf{G}_0$–$\mathbf{G}_4$ in Fig. 28 and Fig. 29. Game $\mathbf{G}_0$ corresponds to game $\mathbf{G}^{\text{real}}_{\mathsf{RtC[F,SE},H]}(\mathcal{A}_1)$, and game $\mathbf{G}_4$ to game $\mathbf{G}^{\text{rand}}_{\mathsf{RtC[F,SE},H]}(\mathcal{A}_1)$. Thus

$$\mathbf{Adv}^{\text{mrae}}_{\mathsf{RtC[F,SE},H]}(\mathcal{A}_1) = \Pr[\mathbf{G}_0(\mathcal{A}_1)] - \Pr[\mathbf{G}_4(\mathcal{A}_1)] \ .$$

Below, we will describe the transition in the game chain.

In game $\mathbf{G}_1$, each call to F with a master key K_i is replaced by a call to a corresponding truly random function $f_i \leftarrow\!\!\$\ \mathsf{Func}(\mathcal{N}, \{0,1\}^\ell \times \{0,1\}^\lambda)$; the latter is lazily implemented by keeping an array Keys of defined points. To bound the gap between $\mathbf{G}_0$ and $\mathbf{G}_1$, we construct an adversary $\mathcal{B}_2$ attacking the PRF security of F as follows. It runs $\mathcal{A}_1$ and simulates game $\mathbf{G}_0$, but each call to $\mathsf{F}(K_i, \cdot)$ will be replaced by a call to the oracle Eval$(i, \cdot)$. When $\mathcal{A}_1$ outputs its guess, $\mathcal{B}_2$ outputs the same guess. Then $\mathcal{B}_2$'s real world corresponds to game $\mathbf{G}_0$, and its ideal world corresponds to game $\mathbf{G}_1$. In other words,

$$\mathbf{Adv}^{\text{prf}}_{\mathsf{F}}(\mathcal{B}_2) = \Pr[\mathbf{G}_0(\mathcal{A}_1)] - \Pr[\mathbf{G}_1(\mathcal{A}_1)] \ .$$

In game $\mathbf{G}_2$, instead of running $\mathsf{SE.Enc}$ to generate a genuine ciphertext C, we will sample a truly random string of the same length. In addition, instead of running $\mathsf{SE.Dec}$ and comparing the decrypted message to $\perp$, we will instead return false. As a result, the oracle Vf will always return false, and the code can be simplified accordingly.

To bound the gap between $\mathbf{G}_1$ and $\mathbf{G}_2$, we construct the following adversary $\mathcal{B}_3$ attacking the

Game $\mathbf{G}_2(\mathcal{A}_1)$	Games $\mathbf{G}_3(\mathcal{A}_1)$, $\mathbf{G}_4(\mathcal{A}_1)$
$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}_1^{\text{NEW},\text{ENC},\text{VF}}$ Return $(b' = 1)$	$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}_1^{\text{NEW},\text{ENC},\text{VF}}$ Return $(b' = 1)$
NEW() $v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k$	NEW() $v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k$
ENC(i, N, A, M) If $i \notin \{1, \ldots, v\}$ return $\perp$ $P \leftarrow\!\!\$\ \{0,1\}^\ell$ If $\mathsf{Keys}[i, N] \neq \perp$ then $P \leftarrow \mathsf{Keys}[i, N]$ $\mathsf{Keys}[i, N] \leftarrow P$; $C \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.len}(\lvert M \rvert)}$ $T \leftarrow H(P, C[1:n])$ Return $T \Vert C$	ENC(i, N, A, M) If $i \notin \{1, \ldots, v\}$ return $\perp$ $C \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.len}(\lvert M \rvert)}$ $R \leftarrow C[1:n]$; $T \leftarrow\!\!\$\ \{0,1\}^n$ If $\mathsf{Tbl}[i, N, R] \neq \perp$ then $\mathsf{bad} \leftarrow \mathsf{true}$; $T \leftarrow \mathsf{Tbl}[i, N, R]$ $\mathsf{Tbl}[i, N, R] \leftarrow T$; return $T \Vert C$
VF$(i, N, A, T \Vert C)$ If $i \notin \{1, \ldots, v\}$ return $\perp$ Return false	VF$(i, N, A, T \Vert C)$ If $i \notin \{1, \ldots, v\}$ return $\perp$ Return false

Figure 29: Games $\mathbf{G}_2$–$\mathbf{G}_4$ in the proof of Theorem 7.3. Game $\mathbf{G}_3$ contains the highlighted code, but game $\mathbf{G}_4$ does not.

misuse-resistance security of SE. It runs $\mathcal{A}_1$ and simulates game $\mathbf{G}_1$, and keeps an array Users that translates a pair (i, N) of $\mathcal{A}_1$ to its user index. In particular, in each encryption/verification query of $\mathcal{A}_1$ to user i with nonce N, if $\mathsf{Users}[i, N]$ is not defined then $\mathcal{B}_3$ creates a new user v via the NEW oracle and stores $\mathsf{Users}[i, N] \leftarrow v$, otherwise $\mathcal{B}_3$ retrieves $v \leftarrow \mathsf{Users}[i, N]$. Moreover, for each encryption query (i, N, A, M) of $\mathcal{A}_1$, the call to $\mathsf{SE.Enc}$ in game $\mathbf{G}_1$ will be replaced by a query (v, N, A, M) to the ENC oracle of $\mathcal{B}_3$. Likewise, for each verification query (i, N, A, C) of $\mathcal{A}_1$, instead of calling $\mathsf{SE.Dec}$ and checking if the decrypted message is $\perp$ as in game $\mathbf{G}_1$, adversary $\mathcal{B}_3$ queries (v, N, A, C) to its VF oracle.

We note that $\mathcal{B}_3$ does not query $C \leftarrow \text{ENC}(v, N, A, M)$ first and then query $\text{VF}(v, N, A, C)$. Assume to the contrary that $\mathcal{B}_3$ violates this restriction. This can only happen if $\mathcal{A}_1$ first queries $T \Vert C \leftarrow \text{ENC}(i, N, A, M)$ and then queries $\text{VF}(i, N, A, T' \Vert C)$ with $T' \neq T$. However, in that case $\mathcal{B}_3$'s implementation of the verification oracle of $\mathcal{A}_1$ will first recompute T and then return false (due to the failure of the checking $T' \neq T$) without calling $\mathsf{SE.Dec}$, leading to a contradiction.

Thus game $\mathbf{G}^{\text{real}}_{\mathsf{SE}}(\mathcal{B}_2)$ corresponds to game $\mathbf{G}_1$, and game $\mathbf{G}^{\text{rand}}_{\mathsf{SE}}(\mathcal{B}_2)$ corresponds to game $\mathbf{G}_2$. Hence

$$\mathbf{Adv}^{\text{mrae}}_{\mathsf{SE}}(\mathcal{B}_2) = \Pr[\mathbf{G}_1(\mathcal{A}_1)] - \Pr[\mathbf{G}_2(\mathcal{A}_1)] \ .$$

In game $\mathbf{G}_3$, each call to H under an ENC query for user i and nonce N is replaced by a call to a truly random function $f_{i,N} \leftarrow\!\!\$\ \mathsf{Func}(\{0,1\}^n, \{0,1\}^n)$; the latter is lazily implemented by keeping an array Tbl of defined points. To bound the gap between game $\mathbf{G}_2$ and game $\mathbf{G}_3$, we will construct another adversary $\mathcal{B}_4$ attacking the PRF security of H. It runs $\mathcal{A}_1$ and simulates game $\mathbf{G}_2$, and also keeps an array Users that translates a pair (i, N) of $\mathcal{A}_1$ to its user v like adversary $\mathcal{B}_3$. In the implementation of $\text{ENC}(i, N, A, M)$, instead of calling H, adversary $\mathcal{B}_4$ will retrieve the corresponding user $v \leftarrow \mathsf{Users}[i, N]$, and then runs $\text{EVAL}(v, \cdot)$. Then $\mathcal{B}_4$'s real world corresponds to game $\mathbf{G}_2$, and its ideal world to game $\mathbf{G}_3$. In other words,

$$\mathbf{Adv}^{\text{prf}}_{H}(\mathcal{B}_4) = \Pr[\mathbf{G}_2(\mathcal{A}_1)] - \Pr[\mathbf{G}_3(\mathcal{A}_1)] \ .$$

Recall that in game $\mathbf{G}_3$, for two ENC queries of the same user and nonce, if their ciphertexts $T_0 \| C_0$ and $T_1 \| C_1$ satisfy $C_0[1:n] = C_1[1:n]$ then we must have $T_0 = T_1$. In contrast, in $\mathbf{G}_4$, each string T is sampled independent of prior ones. Games $\mathbf{G}_3$ and $\mathbf{G}_4$ are identical until the flag bad is set, and thus from the Fundamental Lemma of Game Playing [11],

$$\Pr[\mathbf{G}_3(\mathcal{A}_1)] - \Pr[\mathbf{G}_4(\mathcal{A}_1)] \leq \Pr[\mathbf{G}_4(\mathcal{A}_1) \text{ sets } \mathsf{bad}] \ .$$

We now bound the chance that game $\mathbf{G}_4$ sets bad. For each ENC query, because (i) there are at most B prior ENC queries of the same user and nonce, and (ii) the ciphertexts of the queries are uniformly and independently sampled, the chance that this query triggers bad is at most $B/2^n$. Summing up over at most q encryption queries, by the union bound,

$$\Pr[\mathbf{G}_4(\mathcal{A}_1) \text{ sets } \mathsf{bad}] \leq \frac{Bq}{2^n} \ .$$

Summing up,

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{mrae}}_{\mathsf{RtC}[\mathsf{F},\mathsf{SE}]}(\mathcal{A}_1) &= \Pr[\mathbf{G}_0(\mathcal{A}_1)] - \Pr[\mathbf{G}_4(\mathcal{A}_1)] \\
&= \sum_{i=0}^{3} \Pr[\mathbf{G}_i(\mathcal{A}_1)] - \Pr[\mathbf{G}_{i+1}(\mathcal{A}_1)] \\
&\leq \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{F}}(\mathcal{B}_2) + \mathbf{Adv}^{\mathrm{mrae}}_{\mathsf{SE}}(\mathcal{B}_3) + \mathbf{Adv}^{\mathrm{prf}}_{H}(\mathcal{B}_4) + \frac{Bq}{2^n}
\end{aligned}$$

as claimed.