Robust AE With Committing Security

Viet Tung Hoang[1] and Sanketh Menda[2]

[1] Department of Computer Science, Florida State University, USA
tvhoang@cs.fsu.edu
[2] Department of Computer Science, Cornell Tech, USA
sm2289@cornell.edu

Abstract. There has been a recent interest to develop and standardize Robust Authenticated Encryption (Robust AE) schemes. NIST, for example, is considering an Accordion mode (a wideblock tweakable blockcipher), with Robust AE as a primary application. On the other hand, recent attacks and applications suggest that encryption needs to be committing. Indeed, committing security is also a design consideration in the Accordion mode. Yet it is unclear how to build a Robust AE with committing security.

In this work, we give a modular solution for this problem. We first show how to transform any wideblock tweakable blockcipher TE to a Robust AE scheme SE that commits just the key. The overhead is cheap, just a few finite-field multiplications and blockcipher calls. If one wants to commit the entire encryption context, one can simply hash the context to derive a 256-bit subkey, and uses SE on that subkey. The use of 256-bit key on SE only means that it has to rely on AES-256 but doesn't require TE to have 256-bit key.

Our approach frees the Accordion designs from consideration of committing security. Moreover, it gives a big saving for several key-committing applications that don't want to pay the inherent hashing cost of full committing.

Keywords: Robust authenticated encryption · committing security

1 Introduction

Authenticated Encryption (AE) is widely used in practice to provide data privacy and authenticity. Yet standard AE schemes such as GCM are both fragile and inflexible. On the one hand, if some misuse happens, say nonce repetition, then security is completely broken. On the other hand, for standard AE schemes, the ciphertext C must be sufficiently longer than the message M, so that forgeries will never happen in practice. In particular the *ciphertext expansion* $\tau = |C|-|M|$ is typically 128 bits. But several applications, such as Voice-over-IP or IoT, demand shorter expansion (say 64 bits, or even 32 bits) to minimize latency or energy consumption. One cannot simply truncate the tag of a standard AE

scheme because forgeries will happen sooner or later, and once a forgery happen, all security guarantees are voided.[3]

ROBUST AE. There has been a long line of work to deal with the situation above [26, 18, 2, 3], culminating in the Robust AE notion of Hoang, Krovetz, and Rogaway [18]. Instead of using a fixed expansion τ, Robust AE explicitly accepts a user-defined choice of τ for each encryption. If τ is small then forgeries of course will happen, but just occasionally, roughly once per 2^τ attempts. Robust AE also provides misuse resistance for nonce reuse [26] and protects against releases of unverified decrypted messages [2]. Because of such strong guarantees, standard agencies are actively seeking Robust AE schemes for standardization. The UK National Cyber Security Centre, for example, recently release their own Robust AE schemes [9]. NIST is also considering an Accordion mode for a wideblock tweakable blockcipher (TBC), with Robust AE as a primary application.

THE NEED FOR COMMITTING SECURITY. Robust AE aims to provide the best security possible, but it only covers privacy and authenticity. Recent attacks, such as the Partitioning-Oracle attack on password-based encryption [20], highlight the need for encryption to be committing. This guarantee is also needed by several recent applications, such as Facebook's Message Franking [16], Amazon Cloud encryption, Subscribe with Google [1], or TLS Oracle [22]. Most applications require committing just the key K, but some need to commit all the four inputs (K, N, A, M) of encryption. Due to such demand, it is desirable to build a scheme that provides both Robust AE and committing security. Indeed, committing security is also a property that NIST are considering in the call for the Accordion mode.

OBSTACLES. Unfortunately, existing Robust AE schemes such as AEZ [18] or HCTR2 [14] do not offer (full) committing security. The obvious reason is their hashing of the associated data via a universal hash instead of a collision-resistant one. However, there is a subtle, quantitative reason. In particular, suppose that we only need s bits of expansion and target s bits of committing security.[4] In prior constructions of committing AE schemes [1, 4, 11, 5], the common approach is to make the tag a commitment of (K, N, A), and the message will be committed due to decryption correctness. However, this approach doesn't work for Robust AE schemes. First, Robust AE can only be realized via the Encode-then-Encipher (EtE) paradigm [7]: encode the message with s-bit redundancy (say padding it with 0^s) and then encipher it with a wideblock TBC. The EtE method has no tag, defeating the prior approach of building committing AE.

[3] If the expansion is short and an adversary can obtain decrypted messages, standard AE schemes are *inherently* insecure because the encryption algorithm makes just one pass over data. Specifically, two ciphertexts of the same prefix would decrypt to two messages of the same prefix.

[4] Generic attacks [5] show that we can at best hope for s-bit committing security given s-bit expansion. Moreover, given that committing attacks are offline, we want to achieve s bits of committing security instead of a "birthday-bound" $s/2$ bits of security.

Next, birthday attacks suggest that the commitment must be at least $2s$ bits, but we only have s-bit space.

Given the birthday attacks, at the first glance, it seems that to provide s-bit of committing security, we are doomed to use at least $2s$ bits of expansion. However, a closer inspection reveals that the attacks only require that the ciphertext must be at least $2s$ bits. Thus as long as messages are at least s bits long[5] then the attacks do not apply. This gives a way out of the impossibility, but it is unclear how to exploit this opening.

Related work. Chen et al. [13] show that AEZ [18] offers 64-bit key-committing security for 128-bit expansion, assuming that the underlying tweakable blockcipher is modeled as ideal. (The result is tight, with a matching attack.) However, this security guarantee is weak, because given 128-bit expansion, we want 128-bit security, not merely 64-bit. Moreover, committing security has never been a design goal of AEZ, and thus the security by accident here gives no insight on how one should build a committing Robust AE scheme.

A very recent work by Bellare and Hoang [5] considers adding s bits of committing security to a base AE scheme using just s bits of expansion (assuming that messages are at least s bits). Their work only deals with tag-based AE schemes and thus doesn't apply to Robust AE. However, implicitly their work contains a technical tool that is central to our construction. We will elaborate later how to use their ideas for our setting.

Contributions. We initiate the study of committing Robust AE. By extending the definitions of Bellare and Hoang [4], we formalize two notions of committing security: (1) the CMT notion that commit all inputs (K, N, A, τ, M) of the encryption algorithm, and (2) the CMT-1 notion that commits just the key K.

Achieving CMT security demands that one hashes the associated data A with a cryptographic hash function, such as SHA-2 or SHA-3. While this cost is $O(1)$ in theory, the actual relative overhead is huge for small data. We therefore consider a modular route. First, we focus on building a CMT-1 Robust AE scheme SE that is enough for most applications and doesn't have to pay the hashing penalty. Next, for applications that demand the full CMT security, one can *non-intrusively* add CMT security to SE via the Hash-then-Encrypt (HtE) transform of Hoang and Bellare [4]: first hash (K, N, A, τ) to derive a subkey L, and then encrypt with key L, the empty nonce, the empty AD, and expansion τ. Using HtE means that SE needs to use 256-bit key, but this aligns well with (i) NIST's requirement that an Accordion mode must support 256-bit key, and (ii) the fact that our CMT-1 construction has to use a blockcipher of 256-bit key anyway.

For CMT-1 security, we build a transform EwC that turns any wideblock TBC TE to another $\overline{\mathsf{TE}}$ such that using the latter in the EtE method provides CMT-1 security. While EwC uses a 256-bit key, it doesn't require the base TE

[5] This assumption is reasonable. Indeed, existing Robust AE schemes can't encrypt tiny messages, as AES-based wideblock TBC can only efficiently encipher messages of at least 128 bits.

to have a 256-bit key. This allows us to leverage a wealth of existing wideblock TBC constructions and future Accordion submissions. The overhead of EwC is cheap, just a few finite-field multiplications and blockcipher calls. The underlying blockcipher E uses a 256-bit key, and thus can be instantiated via AES-256 or Rijndael-256. Moreover, EwC only uses E in the forward direction, which saves code size in hardware implementation.

If the underlying blockcipher E has 256-bit block length (such as Rijndael-256), then EwC would encipher messages of at least 512 bits, and provides s bits of CMT-1 security when used in EtE with s bits expansion. If E only has 128-bit block length (such as AES-256) then the CMT-1 security of EwC is more nuanced, because implicitly it takes a parameter $\ell < 128$, and enciphers messages at least $256 + \ell$ bits. Using $\mathsf{EwC}[\ell]$ in EtE still allows one to use any expansion s but only guarantees to deliver CMT-1 security when $s > \ell$, and in that case it provides just $\ell - 8$ bits of CMT-1 security.

At the bird's-eye view, EwC is a four-round (unbalanced) Feistel-like structure. (See Fig. 15 for an illustration.) The first and last rounds, following Naor and Reingold [24], are based on an AXU hash function. Since the input length of the AXU is short, it amounts to just one or two finite-field multiplications for each hashing. The second round uses TE. The third round uses a collision-resistant PRF H and a *committing concealer*, a new primitive that we will discuss later. The function H only needs to deal with short inputs and has to produce a 256-bit output. Thus if the blockcipher E has 256-bit block length, we can directly instantiate H via the Davies-Meyer construction, meaning $H(K, M) = E_K(M) \oplus M$. If E has just 128-bit blockcipher, we show how to build H via "doubling" Davies-Meyer, meaning $H(K, M) = (E_K(U) \oplus U) \| (E_K(V) \oplus V)$, where $U = M \| 0$ and $V = M \| 1$.

Cost. The overhead of EwC is listed in Table 1. For each instantiation of the blockcipher (AES-256 or Rijndael-256), we list two values for the number of blockcipher calls because (i) if one only uses the standalone EwC for key-committing security, we can cache the subkeys, but (ii) if one uses EwC with the HtE transform for full committing security, then we must account for the cost of subkey generation. While the subkey generation seems expensive (say six AES calls), these blockcipher calls are fully parallelizable. On platforms with vector AES instructions, these AES calls would take almost the same amount of time as couple AES calls.

Block length n	**Blockcipher**	**Blockcipher calls**	**Mults in $\mathsf{GF}(2^n)$**
128	AES-256	10 (4, if subkeys are cached)	4
256	Rijndael-256	6 (2, if subkes are cached)	2

Table 1: The overhead of the EwC transform. The third column shows the number of blockcipher calls; two numbers are given, one includes the cost of subkey generation and another doesn't. The last column shows the number of finite-field multiplications.

Implementing a performant version of EwC is tricky. The major problem comes from having three AES keys (one for the wideblock TBC, another for the Davies-Meyer, and yet another for the committing concealer). Overall, that generates a lot of AES subkeys, and it is tricky to ensure that we have no register spill. See Fig. 17 for experimental data for the overhead of EwC on HCTR2. The cost is significant for small data, but becomes negligible for messages bigger than 1KB.

Another way to get CMT security. So far, to get full CMT security, we have to compose EwC with the HtE transform. But this means we no longer cache the subkeys. Moreover, we have to pay for a key setup cost for the wideblock TBC. To improve the running time, we give a variant EwC2 of EwC that directly achieves full CMT security when used in EtE. This construction avoids all the issues above and also cuts the cost of the Davies-Meyer in the third round of our Feistel-like structure. The drawback is that we need to process the tweak twice, one via a cryptographic hash function like SHA-512 (which is inherent for CMT security), and another via the wideblock TBC. Still, if we use EwC2 in EtE, the tweak is usually short, and in the wideblock TBC, it will be often processed by a fast AXU hash function.

Committing concealer. Central to our EwC/EwC2 transforms is a new primitive that we call *committing concealer*. Committing concealer can be viewed as a blockcipher but the security requirement is different. Traditionally, we want a blockcipher to be a strong PRP; if we build it from a Feistel network, we need at least four rounds. For committing concealer, we only need it to be a *one-time* strong PRP (meaning that the adversary can make just a single query per user), and thus we can build it from just two-round Feistel. Still, we want committing concealer to commit the key if used only on the set of messages that are encoded with s-bit redundancy.

If we have a blockcipher E of 256-bit block length, one can directly use it as a committing concealer; the key-committing property would be justified by modeling E as an ideal cipher. However, it is tricky if E only has 128-bit block length. The core idea is implicitly in the recent work of Bellare and Hoang [5]. Technically, they need to commit messages up to $m < n$ bits to produce a commitment of $(m+n)$-bit length and nearly n bits of binding security. They give a construction via the SIV paradigm [26], but alternatively, their construction can be viewed as padding the message with zeros, and then enciphering it with a committing concealer. Their (implicit) committing concealer is based on a two-round unbalanced Feistel, where the round functions are implemented via the Davies-Meyer construction on a blockcipher of block length n. See Fig. 13 for an illustration.

The use of ideal-cipher model. If one uses our EwC/EwC2 transforms in the EtE construction, one would have Robust AE security in the standard model. But the committing security has to be justified in the ideal-cipher model. The use of idealized models in cryptographic constructions has always been a controversial issue, and sometimes it can stand in the way towards adoption. The UK

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

Fig. 1: Game defining (multi-user) PRF security of F.

National Cyber Security Centre, for example, do not target committing security in their Robust AE schemes due to concerns about the use of the ideal-cipher model [9]. We argue that as long as the Robust AE is still justified in the standard model, we can only gain by additionally providing committing security (even in the ideal-cipher model).

2 Preliminaries

2.1 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 $f : \mathsf{Dom} \to \mathsf{Rng}$. We use $\perp$ as a special symbol to denote rejection, and it is assumed to be outside $\{0,1\}^*$. 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. For an integer $n \ge 1$, let $\{0,1\}^{\le n}$ denote the set of all bit strings whose length is at most n, and let $\{0,1\}^{\ge n}$ denote the set of all bit strings whose length is at least n.

2.2 Some Standard Primitives

Collision resistance. Let $H : \mathsf{Dom} \to \mathsf{Rng}$ be a function. A collision for H is a pair (X_1, X_2) of distinct points in Dom such that $H(X_1) = H(X_2)$. For an adversary $\mathcal{A}$, define its advantage in breaking the collision resistance of H as

$$\mathbf{Adv}^{\mathrm{coll}}_{H}(\mathcal{A}) = \Pr[(X_1, X_2) \text{ is a collision for } H]$$

where the probability is over $(X_1, X_2) \leftarrow\!\!\$\ \mathcal{A}$.

AXU hash. Let $G : \mathcal{K} \times \mathcal{M} \to \{0,1\}^n$ be a keyed hashed function. We say that G is *c-almost XOR-universal* (c-AXU) if for any distinct $X, Y \in \mathcal{M}$ and any $\Delta \in \{0,1\}^n$,

$$\Pr_{K \leftarrow\!\!\$\ \mathcal{K}}[G_K(X) \oplus G_K(Y) = \Delta] \le \frac{c}{2^n} \ .$$

Game $\mathbf{G}_E^{\pm\mathsf{prp}}(\mathcal{A})$	$\textsc{Enc}(i, M)$
$v \leftarrow 0;\ \ b \leftarrow\!\!\$\ \{0,1\}$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Dec}}$	$C_1 \leftarrow E(K_i, M);\ \ C_0 \leftarrow \Pi_i(M)$
Return $(b' = b)$	Return C_b
$\textsc{New}()$	$\textsc{Dec}(i, C)$
$v \leftarrow v + 1;\ \ K_v \leftarrow\!\!\$\ \{0,1\}^k$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$\Pi_v \leftarrow\!\!\$\ \mathsf{Perm}(n)$	$M_1 \leftarrow E^{-1}(K_i, C);\ \ M_0 \leftarrow \Pi_i^{-1}(C)$
	Return M_b

Fig. 2: Game defining (multi-user) strong PRP security of E. Here $\mathsf{Perm}(n)$ denotes the set of all permutations in $\{0,1\}^n$

PRF. For a function $\mathsf{F} : \mathcal{K} \times \mathsf{Dom} \to \mathsf{Rng}$ and an adversary $\mathcal{A}$, we define the advantage of $\mathcal{A}$ in breaking the (multi-user) PRF security of F as

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

where game $\mathbf{G}^{\mathsf{prf}}_{\mathsf{F}}(\mathcal{A})$ is shown in Fig. 1.

PRP. 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) strong-PRP security of E as

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

where game $\mathbf{G}^{\pm\mathsf{prp}}_{E}(\mathcal{A})$ is shown in Fig. 2.

(Wideblock) Tweakable Blockcipher. A *tweakable blockcipher* (TBC) TE consists of two deterministic algorithms $\mathsf{TE.Enc}$ and $\mathsf{TE.Dec}$, and is associated with a key space $\mathcal{K}$, a message space $\mathcal{M}$, and a tweak space $\mathcal{T}$. The enciphering algorithm $\mathsf{TE.Enc}$ takes as input a key $K \in \mathcal{K}$, a message $M \in \mathcal{M}$, a tweak $T \in \mathcal{T}$, and outputs a ciphertext $C \leftarrow \mathsf{TE.Enc}(K, T, M)$. The deciphering algorithm $\mathsf{TE.Dec}$ takes as input (K, T, C) and produces $M \leftarrow \mathsf{TE.Dec}(K, T, C)$. For correctness, we require that deciphering reverses enciphering, meaning that if $C \leftarrow \mathsf{TE.Enc}(K, T, M)$ then $\mathsf{TE.Dec}(K, T, C) = M$.

In this paper, we consider *wideblock* TBC, meaning that the message space consists of messages of different length, say $\mathcal{M} = \{0,1\}^{\geq m}$. We require that enciphering preserves the message length, meaning that $|C| = |M|$.

Define the advantage of an adversary $\mathcal{A}$ breaking the strong tweakable-PRP security of TE as

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathsf{TE}}(\mathcal{A}) = 2\Pr[\mathbf{G}^{\pm\widetilde{\mathsf{prp}}}_{\mathsf{TE}}(\mathcal{A})] - 1 \ ,$$

where game $\mathbf{G}^{\pm\widetilde{\mathsf{prp}}}_{\mathsf{TE}}(\mathcal{A})$ is shown in Fig. 3.

Game $\mathbf{G}^{\pm\widetilde{\mathsf{prp}}}_{\mathsf{TE}}(\mathcal{A})$	$\textsc{Enc}(i, T, M)$
$v \leftarrow 0$; $b \leftarrow\$ \{0,1\}$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$b' \leftarrow\$ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Dec}}$	$C_1 \leftarrow \mathsf{TE.Enc}(K_i, T, M)$; $C_0 \leftarrow \Pi_{i,T}(M)$
Return $(b' = b)$	Return C_b
$\textsc{New}()$	$\textsc{Dec}(i, T, C)$
$v \leftarrow v + 1$; $K_v \leftarrow\$ \{0,1\}^k$	If $i \notin \{1, \ldots, v\}$ return $\perp$
For $T \in \mathcal{T}$ do $\Pi_{v,T} \leftarrow\$ \mathrm{LP}(\mathcal{M})$	$M_1 \leftarrow \mathsf{TE.Dec}(K_i, T, C)$; $M_0 \leftarrow \Pi^{-1}_{i,T}(C)$
	Return M_b

Fig. 3: Game defining (multi-user) strong tweakable-PRP security of TE. Here $\mathrm{LP}(\mathcal{M})$ denote the set of permutations π on $\mathcal{M}$ that are length-preserving, meaning $|\pi(M)| = |M|$ for every $M \in \mathcal{M}$.

2.3 Robust Authenticated Encryption

Syntax. An *robust authenticated encryption* (RAE) scheme SE consists of two deterministic algorithms $\mathsf{SE.Enc}$ and $\mathsf{SE.Dec}$; it is associated with a nonce space $\mathcal{N}$, a message space $\mathcal{M}$, key space $\mathcal{K}$, and an expansion space $\mathcal{X}$. The encryption algorithm takes as input a key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$, associated data $A \in \{0,1\}^*$, a message $M \in \mathcal{M}$, and an expansion $\tau \in \mathcal{X}$, and returns a ciphertext $C \leftarrow \mathsf{SE.Enc}(K, N, A, M, \tau)$ such that $|C| = |M| + \tau$. The decryption algorithm takes as input (K, N, A, C, τ) and returns either a message $M \in \mathcal{M}$ or a leakage $L \notin \mathcal{M}$. The correctness requirement says that decryption reverses encryption, namely if $C \leftarrow \mathsf{SE.Enc}(K, N, A, M, \tau)$ then $\mathsf{SE.Dec}(K, N, A, C, \tau)$ returns M.

We say that SE is *tidy* [23] if $M \leftarrow \mathsf{SE.Dec}(K, N, A, C, \tau)$ and $M \in \mathcal{M}$ imply that $\mathsf{SE.Enc}(K, N, A, M, \tau)$ returns C. Combining correctness and tidiness means that functions $\mathsf{SE.Enc}(K, N, A, \cdot, \tau)$ and $\mathsf{SE.Dec}(K, N, A, \cdot, \tau)$ are the inverse of each other. The schemes we consider will be tidy.

Standard AE schemes are a special case of RAE schemes where the set $\mathcal{X}$ is a singleton, meaning that the expansion is a constant, say 128. In general, RAE schemes support a large range of expansion values, typically $\{0, 1, \ldots, 128\}$, and the expansion value can dynamically change within the same session.

RAE security. Let SE be an RAE scheme of message space $\mathcal{M}$. Define the advantage of an adversary $\mathcal{A}$ breaking the RAE security of SE with respect to a (stateful) simulator Sim as

$$\mathbf{Adv}^{\mathrm{rae}}_{\mathsf{SE},\mathsf{Sim}}(\mathcal{A}) = 2 \cdot \Pr[\mathbf{G}^{\mathrm{rae}}_{\mathsf{SE},\mathsf{Sim}}(\mathcal{A})] - 1 \ ,$$

where game $\mathbf{G}^{\mathrm{rae}}_{\mathsf{SE},\mathsf{Sim}}(\mathcal{A})$ is defined in Fig. 4. To prevent trivial attacks, we forbid the adversary from first querying $C \leftarrow \textsc{Enc}(i, N, A, M, \tau)$ and then querying $\textsc{Dec}(i, N, A, C, \tau)$.

Game $\mathbf{G}^{\mathrm{rae}}_{\mathsf{SE},\mathsf{Sim}}(\mathcal{A})$	$\textsc{Enc}(i, N, A, M, \tau)$
$v \leftarrow 0$; $b \leftarrow\$ \{0,1\}$; $st \leftarrow \varepsilon$	If $i \notin \{1, \ldots, v\}$ return $\bot$
$b' \leftarrow\$ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Dec}}$	$C_1 \leftarrow \mathsf{SE.Enc}(K_i, N, A, M, \tau)$
Return $(b' = b)$	$C_0 \leftarrow \Pi_{i,N,A,\tau}(M)$
$\textsc{New}()$	Return C_b
$v \leftarrow v + 1$; $K_v \leftarrow\$ \{0,1\}^k$	$\textsc{Dec}(i, N, A, C, \tau)$
For $(N, A, \tau) \in \mathcal{N} \times \{0,1\}^* \times \mathcal{X}$ do	If $i \notin \{1, \ldots, v\}$ return $\bot$
$\quad \Pi_{v,N,A,\tau} \leftarrow\$ \mathrm{Inj}(\tau)$	$M_1 \leftarrow \mathsf{SE.Dec}(K_i, N, A, C, \tau)$
	$M_0 \leftarrow \Pi^{-1}_{i,N,A,\tau}(C)$
	If $M_0 = \bot$ then
	$\quad (M_0, st) \leftarrow\$ \mathsf{Sim}(i, N, A, C, \tau, st)$
	Return M_b

Fig. 4: Game defining (multi-user) RAE security of SE with respect to a simulator Sim. Here $\mathrm{Inj}(\tau)$ denote the set of injective functions $f : \{0,1\}^* \to \{0,1\}^*$ such that $|f(M)| = |M| + \tau$ for all messages M.

$\mathsf{EtE}[\mathsf{TE}].\mathsf{Enc}(K, N, A, M, \tau)$	$\mathsf{EtE}[\mathsf{TE}].\mathsf{Dec}(K, N, A, C, \tau)$
$V \leftarrow \mathsf{pad}(M, \tau)$; $T \leftarrow (N, A, \tau)$	$T \leftarrow (N, A, \tau)$
$C \leftarrow \mathsf{TE.Enc}(K, T, V)$	$V \leftarrow \mathsf{TE.Dec}(K, T, C)$; $M \leftarrow \mathsf{unpad}(V, \tau)$
Return C	If $M \neq \bot$ then return M
	Else return $(\bot, V)$ // Decryption leakage

Fig. 5: The EtE method, with decryption leakage on invalid ciphertexts.

In the game above, the simulator Sim is only called on invalid ciphertexts to simulate the decryption leakage, but is *not* given any information on messages of encryption queries. The simulator is stateful and explicitly maintains its state st.

Encode-then-Encipher (EtE). Hoang, Krovetz, and Rogaway [18] show that to achieve RAE security, one has to use the Encode-then-Enciphering (EtE) paradigm of Bellare and Rogaway [7]. We now recall the details of the EtE construction. Let $\mathsf{pad} : \{0,1\}^* \times \mathcal{X} \to \{0,1\}^*$ be a padding scheme such that (1) for any $\tau \in \mathcal{X}$, the function $\mathsf{pad}(\cdot, \tau)$ is injective, and let $\mathsf{unpad}(\cdot, \tau) : \{0,1\}^* \to \{0,1\}^* \cup \{\bot\}$ be its inverse, and (2) If $Y \leftarrow \mathsf{pad}(X, \tau)$ then $|Y| = |X| + \tau$. For example, we can let $\mathsf{pad}(M, \tau) = M \| 0^\tau$. Let TE be a wideblock TBC with tweak space $\mathcal{N} \times \{0,1\}^* \times \mathcal{X}$. The scheme $\mathsf{EtE}[\mathsf{TE}]$ is specified in Fig. 5; it has nonce space $\mathcal{N}$ and expansion space $\mathcal{X}$. Informally, to encrypt M under (K, N, A, τ), we pad M with $\mathsf{pad}(\cdot, \tau)$ and then encipher it with tweak (N, A, τ). On decryption, we first recover $V \leftarrow \mathsf{TE.Dec}(K, T, C)$ and check the unpadding. If the unpadding fails then we model the decryption leakage as $L \leftarrow (\bot, V)$. (Note that since we require $L \notin \mathcal{M}$, the invalidity symbol has to be included.)

Game $\mathbf{G}^{\mathrm{cmt}}_{\mathsf{SE}}(\mathcal{A})$
$\big((K_1, N_1, A_1, M_1, \tau_1), (K_2, N_2, A_2, M_2, \tau_2)\big) \leftarrow\!\!\$\ \mathcal{A}$
$C_1 \leftarrow \mathsf{SE.Enc}(K_1, N_1, A_1, M_1, \tau_1)$; $C_2 \leftarrow \mathsf{SE.Enc}(K_2, N_2, A_2, M_2, \tau_2)$
Return $\big((C_1 = C_2) \wedge (K_1, N_1, A_1, M_1, \tau_1) \neq (K_2, N_2, A_2, M_2, \tau_2)\big)$

Fig. 6: Game defining committing security, encryption-based style.

Game $\mathbf{G}^{\mathrm{cmtd}}_{\mathsf{SE}}(\mathcal{A})$
$\big(C, (K_1, N_1, A_1, \tau_1), (K_2, N_2, A_2, \tau_2)\big) \leftarrow\!\!\$\ \mathcal{A}$
$M_1 \leftarrow \mathsf{SE.Dec}(K_1, N_1, A_1, C_1, \tau_1)$; $M_2 \leftarrow \mathsf{SE.Dec}(K_2, N_2, A_2, C, \tau_2)$
Return $\big((M_1 \in \mathcal{M}) \wedge (M_2 \in \mathcal{M}) \wedge (K_1, N_1, A_1, \tau_1) \neq (K_2, N_2, A_2, \tau_2)\big)$

Fig. 7: Game defining committing security, decryption-based style. Here $\mathcal{M}$ is the message space.

3 Committing Security for RAE Schemes

In this section, we give a definitional treatment of committing security for RAE schemes. The definitions are a straightforward extension of the work of Bellare and Hoang [4] for standard AE schemes. The main issue is whether to restrict adversaries from generating collisions on the same, or possibly different, expansions τ_1 and τ_2. It is easy to see that allowing different ones is a strictly stronger security goal, and so we opt for it.

3.1 Definitions

Committing security for RAE schemes. For an adversary $\mathcal{A}$, we define its advantage in breaking the committing security of an RAE scheme SE as

$$\mathbf{Adv}^{\mathrm{cmt}}_{\mathsf{SE}}(\mathcal{A}) = \Pr[\mathbf{G}^{\mathrm{cmt}}_{\mathsf{SE}}(\mathcal{A})] \ ,$$

where game $\mathbf{G}^{\mathrm{cmt}}_{\mathsf{SE}}(\mathcal{A})$ is defined in Fig. 6. Informally, committing security means that an adversary cannot produce a ciphertext collision. This generalizes the notion CMT-4 security of Bellare and Hoang [4].

The definition above uses an encryption-based style where the adversary specifies the messages and the game encrypts them to compare the ciphertexts. Alternatively, we can define a decryption-based one as follows. Define

$$\mathbf{Adv}^{\mathrm{cmtd}}_{\mathsf{SE}}(\mathcal{A}) = \Pr[\mathbf{G}^{\mathrm{cmtd}}_{\mathsf{SE}}(\mathcal{A})] \ ,$$

where game $\mathbf{G}^{\mathrm{cmtd}}_{\mathsf{SE}}(\mathcal{A})$ is defined in Fig. 7. Informally, this means that a ciphertext cannot be properly decrypted under two different contexts (K_1, N_1, A_1, τ_1) and (K_2, N_2, A_2, τ_2).

Relations. Following Bellare and Hoang [4], we show that CMT and CMT-D security are equivalent.

Game $\mathbf{G}^{\text{cmt-1}}_{\mathsf{SE}}(\mathcal{A})$
$\big((K_1, N_1, A_1, M_1, \tau_1), (K_2, N_2, A_2, M_2, \tau_2)\big) \leftarrow\!\!\$\ \mathcal{A}$
$C_1 \leftarrow \mathsf{SE.Enc}(K_1, N_1, A_1, M_1, \tau_1)$; $C_2 \leftarrow \mathsf{SE.Enc}(K_2, N_2, A_2, M_2, \tau_2)$
Return $\big((C_1 = C_2) \wedge (K_1 \neq K_2)\big)$

Fig. 8: Game defining CMT-1 security.

$\triangleright$ CMTD $\longrightarrow$ CMT: First we show that CMTD implies CMT. Let SE be an RAE scheme with message space $\mathcal{M}$. Consider an adversary $\mathcal{A}_e$ that attacks the CMT security of SE. We now construct an adversary $\mathcal{A}_d$ attacking the CMTD security of SE. It runs $\mathcal{A}_e$ to get $(K_1, N_1, A_1, M_1, \tau_1)$ and $(K_2, N_2, A_2, M_2, \tau_2)$. It then computes $C \leftarrow \mathsf{SE.Enc}(K_1, N_1, A_1, M_1, \tau_1)$, and outputs $(C, (K_1, N_1, A_1, \tau_1), (K_2, N_2, A_2, \tau_2))$.

For analysis, without loss of generality, assume that $\mathcal{A}_e$ outputs distinct tuples $(K_1, N_1, A_1, M_1, \tau_1)$ and $(K_2, N_2, A_2, M_2, \tau_2)$. Suppose that $\mathcal{A}_e$ wins its game, meaning that $\mathsf{SE.Enc}(K_2, N_2, A_2, M_2, \tau_2)$ is also C. From the correctness of SE, we have $\mathsf{SE.Dec}(K_i, N_i, A_i, C, \tau_i) = M_i \in \mathcal{M}$ for each $i \in \{1, 2\}$. If $(K_1, N_1, A_1, \tau_1) = (K_2, N_2, A_2, \tau_2)$ then $M_1 = M_2$, which is a contradiction. Hence $(K_1, N_1, A_1, \tau_1) \neq (K_2, N_2, A_2, \tau_2)$, thus $\mathcal{A}_d$ also wins its game. Therefore,

$$\mathbf{Adv}^{\text{cmtd}}_{\mathsf{SE}}(\mathcal{A}_d) \geq \mathbf{Adv}^{\text{cmt}}_{\mathsf{SE}}(\mathcal{A}_e) \ .$$

$\triangleright$ CMT $\dashrightarrow$ CMTD: Conversely, we show that for tidy schemes, CMT implies CMTD. Let SE be a tidy RAE scheme with message space $\mathcal{M}$. Consider an adversary $\mathcal{A}_d$ that attacks the CMTD security of SE. We now construct an adversary $\mathcal{A}_e$ that attacks the CMT security of SE. It runs $\mathcal{A}_d$ to get $(C, (K_1, N_1, A_1, \tau_1), (K_2, N_2, A_2, \tau_2))$, and gets $M_i \leftarrow \mathsf{SE.Dec}(K_i, N_i, A_i, C, \tau_i)$ for each $i \in \{1, 2\}$. It then outputs $((K_1, N_1, A_1, M_1, \tau_1), (K_2, N_2, A_2, M_2, \tau_2))$.

For analysis, without loss of generality, assume that $\mathcal{A}_d$ outputs distinct tuples $(K_1, N_1, A_1, \tau_1), (K_2, N_2, A_2, \tau_2)$. Suppose that $\mathcal{A}_d$ wins its game, meaning that $M_1 \in \mathcal{M}$ and $M_2 \in \mathcal{M}$. Since SE is tidy, we have $\mathsf{SE.Enc}(K_i, N_i, A_i, M_i, \tau_i) = C$ for each $i \in \{1, 2\}$, and thus $\mathcal{A}_e$ also wins its game. Hence

$$\mathbf{Adv}^{\text{cmt}}_{\mathsf{SE}}(\mathcal{A}_e) \geq \mathbf{Adv}^{\text{cmtd}}_{\mathsf{SE}}(\mathcal{A}_d) \ .$$

CMT-1 SECURITY. The notions CMT and CMTD above commit the entire context (K, N, A, M, τ). Many applications however only need to commit just the key. Following Bellare and Hoang [4], define the advantage of an adversary breaking the CMT-1 of an RAE scheme SE as

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

where game $\mathbf{G}^{\text{cmt-1}}_{\mathsf{SE}}(\mathcal{A})$ is defined in Fig. 8.

DISCUSSION. Note that for CMT security, in the special case that the message is empty, the ciphertext is a τ-bit commitment of the AD. This means that CMT security requires hashing AD by a collision-resistant hash function such as SHA-

$\underline{\mathsf{HtE}[H,\mathsf{SE}].\mathsf{Enc}(K,N,A,M,\tau)}$	$\underline{\mathsf{HtE}[H,\mathsf{SE}].\mathsf{Dec}(K,N,A,C,\tau)}$
$L \leftarrow H(K,N,A,\tau)$	$L \leftarrow H(K,N,A,\tau)$
$C \leftarrow \mathsf{SE}.\mathsf{Enc}(L,\varepsilon,\varepsilon,M,\tau)$	$M \leftarrow \mathsf{SE}.\mathsf{Dec}(K,\varepsilon,\varepsilon,C,\tau)$
Return C	Return M

Fig. 9: The HtE transform.

512 or SHA-3. While this overhead is constant, it's expensive for small messages. In contrast, CMT-1 security only needs to commit a short input (namely the key), and we can use, for example, the Davies-Meyer construction with very low overhead.

Lower bounds. RAE schemes cannot achieve commitment security for small expansion. For example, if an adversary is allowed to choose $\tau_1 = \tau_2 = 0$, then any ciphertext will decrypt to some message under any context. More formally, let SE be a tidy RAE scheme of expansion space $\mathcal{X}$. Let λ be the minimum value in $\mathcal{X}$, and let ℓ be the smallest message length that SE supports. Prior generic committing attacks on standard AE schemes do apply to RAE schemes if we restrict to λ-bit expansion, and treat decryption leakage as a symbol $\perp \notin \mathcal{M}$. In particular, from the attacks of Bellare and Hoang [5], one can at best hope for $\min\{\lambda, (\lambda+\ell)/2\}$ bits of CMT/CMT-1 security for SE.

The term $\lambda + \ell$ is also the smallest message length of the underlying wide-block TBC TE of SE. Practical constructions of TE typically require that $\lambda + \ell$ to be reasonably large, say 128, because otherwise TE has to include a Format-Preserving Encryption scheme [6], which is expensive. For our schemes in Section 6, $\lambda + \ell$ is even larger, say 344 for the AES-based instantiation (if we want 80-bit committing security), or 512 for the Rijndael-256-based one.

3.2 From CMT-1 to CMT security

We extend the Hash-then-Encrypt (HtE) transform of Bellare and Hoang [4] for RAE. This transform turns a CMT-1 secure scheme SE to a CMT-secure one $\mathsf{HtE}[H, \mathsf{SE}]$,

The HtE transform. Let SE be an RAE scheme of key space $\{0,1\}^k$, nonce space $\{\varepsilon\}$, and expansion space $\mathcal{X}$. Let $H : \mathcal{K} \times (\mathcal{N} \times \{0,1\}^* \times \mathcal{X}) \to \{0,1\}^k$ be a (keyed) hash function. The code of $\mathsf{HtE}[H, \mathsf{SE}]$ is specified in Fig. 9. The idea is simple. We first hash $L \leftarrow H(K,N,A,\tau)$ to derive a subkey L, and then run SE to encrypt M with the subkey key L and tweak $(\varepsilon,\varepsilon,\tau)$. Note that the AD A is only processed once, because we use SE with the empty AD. The overhead of HtE is essentially optimal, since the hashing of AD is required for achieving CMT security. Therefore, in this paper, we only focus on building CMT-1 secure RAE scheme.

CMT security of HtE. The following result shows that if H is collision-resistant then HtE promotes CMT-1 security to CMT one.

Proposition 1. *Let* SE *be an RAE scheme of key space* $\{0,1\}^k$*, nonce space* $\mathcal{N}$*, and expansion space* $\mathcal{X}$*. Let* $H : \{0,1\}^k \times (\mathcal{N} \times \{0,1\}^* \times \mathcal{X}) \to \{0,1\}^k$ *be a (keyed) hash function. For any adversary* $\mathcal{A}$*, we can build adversaries* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *such that*

$$\mathbf{Adv}^{\mathrm{cmt}}_{\mathsf{HtE}[H,\mathsf{SE}]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{coll}}_{H}(\mathcal{B}_1) + \mathbf{Adv}^{\mathrm{cmt\text{-}1}}_{\mathsf{SE}}(\mathcal{B}_2) \ .$$

Adversary $\mathcal{B}_1$ *has the same running time and uses the same amount of resource as* $\mathcal{A}$*. Adversary* $\mathcal{B}_2$ *runs* $\mathcal{A}$ *and then runs* H *on* $\mathcal{A}$*'s inputs.*

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

$$\big((K_1, N_1, A_1, M_1, \tau_1), (K_2, N_2, A_2, M_2, \tau_2)\big) \leftarrow\!\!\$\ \mathcal{A} \ .$$

It then outputs $\big((K_1, (N_1, A_1, \tau_1)), (K_2, (N_2, A_2, \tau_2)\big)$.

Adversary $\mathcal{B}_2$ also runs

$$\big((K_1, N_1, A_1, M_1, \tau_1), (K_2, N_2, A_2, M_2, \tau_2)\big) \leftarrow\!\!\$\ \mathcal{A} \ .$$

Let $L_i \leftarrow H(K_i, (N_i, A_i, \tau_i))$ for every $i \in \{1,2\}$. Adversary $\mathcal{B}_2$ then outputs $\big((L_1, \varepsilon, \varepsilon, M_1, \tau_1), (L_2, \varepsilon, \varepsilon, M_2, \tau_2)\big)$.

For analysis, let $C_i \leftarrow \mathsf{HtE}[H, \mathsf{SE}].\mathsf{Enc}(K_i, N_i, A_i, M_i, \tau_i)$ for each $i \in \{1,2\}$. Note that $C_i = \mathsf{SE}.\mathsf{Enc}(L_i, \varepsilon, \varepsilon, M_i, \tau_i)$. Assume that $\mathcal{A}$ succeeds in creating a collision, meaning that $C_1 = C_2$. If $L_1 = L_2$ then $\mathcal{B}_1$ also creates a collision. Suppose that $L_1 \neq L_2$. Then $\mathcal{B}_2$ creates a collision. Hence

$$\mathbf{Adv}^{\mathrm{cmt}}_{\mathsf{HtE}[H,\mathsf{SE}]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{coll}}_{H}(\mathcal{B}_1) + \mathbf{Adv}^{\mathrm{cmt\text{-}1}}_{\mathsf{SE}}(\mathcal{B}_2)$$

as claimed. □

HtE PRESERVES RAE SECURITY. The following result shows that if H is a PRF then HtE preserves the RAE security.

Proposition 2. *Let* SE *be an RAE scheme of key space* $\{0,1\}^k$*, nonce space* $\{\varepsilon\}$*, and expansion space* $\mathcal{X}$*. Let* $H : \{0,1\}^k \times (\mathcal{N} \times \{0,1\}^* \times \mathcal{X}) \to \{0,1\}^k$ *be a (keyed) hash function. Consider an adversary* $\mathcal{A}$ *of* q *queries, with at most* B *queries per (user, nonce, AD, expansion). For any simulator* Sim_b*, we can build adversaries* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *and a simulator* Sim_a *such that*

$$\mathbf{Adv}^{\mathrm{rae}}_{\mathsf{HtE}[H,\mathsf{SE}],\mathsf{Sim}_a}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{prf}}_{H}(\mathcal{B}_1) + \mathbf{Adv}^{\mathrm{rae}}_{\mathsf{SE},\mathsf{Sim}_b}(\mathcal{B}_2) \ .$$

The running time of Sim_a *is about the same as that of* Sim_b*. Adversary* $\mathcal{B}_1$ *makes* q *queries. Its running time is about that of* $\mathcal{A}$ *plus the cost of using* SE *to encrypt/decrypt* $\mathcal{A}$*'s queries. Adversary* $\mathcal{B}_2$ *has about the same running time as* $\mathcal{A}$ *and also makes* q *queries of the total length as* $\mathcal{A}$*, but it makes only* B *queries per user.*

Proof. We first construct the simulator Sim_a. It maintains a counter v for the current number of users and a state st^* for Sim_b, and lazily maintains a map Tbl to translate a tuple (i, N, A, τ) to a user u. On (i, N, A, C, τ, st), it parses st into (v, st^*, Tbl). It then checks if $\mathsf{Tbl}[i, N, A, \tau]$ is defined. If not then it increments v and sets $\mathsf{Tbl}[i, N, A, \tau] \leftarrow v$. In any case, it gets $u \leftarrow \mathsf{Tbl}[i, N, A, \tau]$ and runs

$(M, st^*) \leftarrow\!\!\$\ \mathsf{Sim}_b(u, \varepsilon, \varepsilon, C, \tau, st^*)$. It then update its own state $st \leftarrow (v, \mathsf{Tbl}, st^*)$, and returns (M, st).

Consider the following sequence of games. Game G_0 corresponds to game $\mathbf{G}^{\mathrm{rae}}_{\mathsf{HtE}[H,\mathsf{SE}],\mathsf{Sim}_a}(\mathcal{A})$ with challenge bit 1. Game G_1 is identical to game G_0, except that instead of using $H(K_i, \cdot, \cdot, \cdot)$ to derive the subkeys for user i, we use a truly random function $f_i : \mathcal{N} \times \{0,1\}^* \times \mathcal{X} \to \{0,1\}^k$. To bound the gap between G_0 and G_1, we construct an adversary $\mathcal{B}_1$ attacking the (multi-user) PRF security of H. It runs $\mathcal{A}$ and simulates game G_0, but each call to $H(K_i, \cdot)$ is replaced by a corresponding call to $\textsc{Eval}(i, \cdot)$. Then

$$\mathbf{Adv}^{\mathrm{prf}}_H(\mathcal{B}_1) = \Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})] \ .$$

Next, game G_2 corresponds to game $\mathbf{G}^{\mathrm{rae}}_{\mathsf{HtE}[H,\mathsf{SE}],\mathsf{Sim}_a}(\mathcal{A})$ with challenge bit 0. To bound the gap between G_1 and G_2, we construct an adversary $\mathcal{B}_2$ attacking the (multi-user) RAE security of SE. It runs $\mathcal{A}$. For each encryption query (i, N, A, M, τ) of $\mathcal{A}$, it calls $C \leftarrow \textsc{Enc}(u, \varepsilon, \varepsilon, M, \tau)$ for the effective user $u = (i, N, A, \tau)$, and returns C to $\mathcal{A}$. (This means $\mathcal{B}_2$ must lazily maintain a map from $\mathbb{N} \times \mathcal{N} \times \{0,1\}^* \times \mathcal{X} \to \mathbb{N}$ to translate (i, N, A, τ) to an integer u.) Likewise, for each decryption query (i, N, A, C, τ) of $\mathcal{A}$, it returns $\textsc{Dec}(u, \varepsilon, \varepsilon, C, \tau)$ for the effective user $u = (i, N, A, \tau)$. Hence

$$\mathbf{Adv}^{\mathrm{rae}}_{\mathsf{SE},\mathsf{Sim}_b}(\mathcal{B}_2) = \Pr[G_1(\mathcal{A})] - \Pr[G_2(\mathcal{A})] \ .$$

Summing up,

$$\begin{aligned}\mathbf{Adv}^{\mathrm{rae}}_{\mathsf{HtE}[H,\mathsf{SE}],\mathsf{Sim}_a}(\mathcal{A}) &= \Pr[G_0(\mathcal{A})] - \Pr[G_2(\mathcal{A})] \\ &= (\Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})]) + (\Pr[G_1(\mathcal{A})] - \Pr[G_2(\mathcal{A})]) \\ &= \mathbf{Adv}^{\mathrm{prf}}_H(\mathcal{B}_1) + \mathbf{Adv}^{\mathrm{rae}}_{\mathsf{SE},\mathsf{Sim}_b}(\mathcal{B}_2) \ .\end{aligned}$$

This concludes the proof. □

Instantiation. To have strong committing security, the key length k needs to be 256-bit. If the nonce length is fixed then one can instantiate $H(K, N, A, \tau)$ via SHA-512$(K\|N\|A\|[\tau]_{16})[1:256]$ or SHA-3$(K\|N\|A\|[\tau]_{16})[1:256]$, where $[\tau]_{16}$ is a 16-bit representation of the number of τ. We stress that one should avoid using SHA-256, because of the extension attack.

4 Fast Collision-Resistant PRF From Blockcipher

Recall that in CMT-1 security, we want to commit a short string (namely the key). This doesn't require a fully-fledged collision-resistant hash like SHA-512 or SHA-3. Instead, one can use cheaper constructions like Davies-Meyer applied to AES, as first suggested in the context of commitment security in [4]. Running Davies-Meyer on a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ however can only provide an n-bit commitment. In this section, we investigate how to produce a λ-bit commitment, where $n \le \lambda \le 2n$. Specifically, we want to build a collision-resistant PRF $H : \{0,1\}^k \times \{0,1\}^{n-1} \to \{0,1\}^\lambda$ on top of E with $\lambda/2$ bits of security. Below, we show how to do that by "doubling" Davies-Meyer.

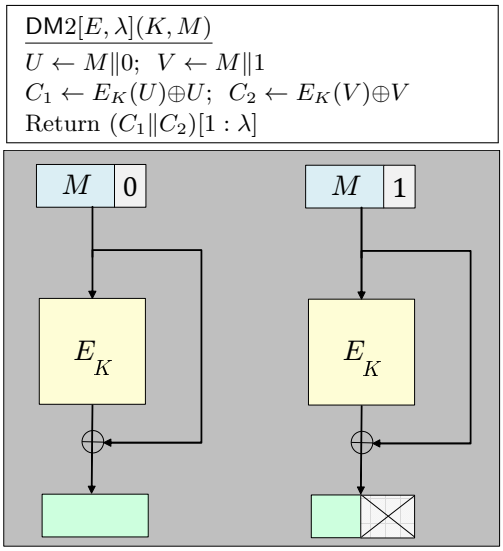

Fig. 10: The DM2 hashing.

THE DOUBLE DAVIES-MEYER HASH. The code of the hash function $\mathsf{DM2}[E, \lambda] : \{0,1\}^k \times \{0,1\}^{n-1} \to \{0,1\}^\lambda$ is given in Fig. 10. Informally, to hash M with key K, we run two Davies-Meyer, one with $(K, M\|0)$, and another with $(K, M\|1)$, and then truncate the concatenated output. For $\lambda = n$, we recover the conventional Davies-Meyer construction.

COLLISION RESISTANCE OF DM2. The following result confirms that $\mathsf{DM2}[E, \lambda]$ has $\lambda/2$-bit collision resistance if E is modeled as an ideal cipher.

Proposition 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 $n \le \lambda \le 2n$. Then for any adversary $\mathcal{A}$ that makes at most q ideal-cipher queries,*

$$\mathbf{Adv}^{\mathrm{coll}}_{\mathsf{DM2}[E,\lambda]}(\mathcal{A}) \le \frac{8q^2}{2^\lambda} + \frac{2}{2^n} \ .$$

Proof. Without loss of generality, assume that the adversary does not make redundant queries. That is, it does not repeat prior queries, and if it queries $C \leftarrow E_K(P)$ then later it will not query $E_K^{-1}(C)$, and if it queries $P \leftarrow E_K^{-1}(C)$ then it will not later query $E_K(P)$. For each query $C \leftarrow E(K, P)$ we store a log entry $(K, P, C \oplus P)$. Likewise, for each query $P \leftarrow E^{-1}(K, C)$, we store a log entry $(K, P, C \oplus P)$. For an n-bit string P, let $\overline{P}$ denote the string obtained by flipping the last bit of P.

If a query results in a log entry (K, P, C) and there is no prior entry $(K, \overline{P}, C^*)$ then we immediately grant the adversary a free query $E_K(\overline{P})$ and store the corresponding log entry. These free queries can only help the adversary. As a

result, if we sort the log entries according to their querying order, then the ith query is the granted ones, for every even $i \le 2q$.

Without loss of generality, assume that the right-hand side of the claimed bound is smaller than 1; otherwise the bound is moot. That is, $q \le 2^{n-1}$. Thus for each entry (K, P, C), conditioning on prior entries, the value X is uniformly chosen from a set of at least $2^n - q \ge 2^{n-1}$ members. Let $r = \lambda - n$.

Suppose that $\mathcal{A}$ outputs (K_1, M_1, K_2, M_2). Let Bad be the event that there are entries $(K_1, M_1\|0, X_1), (K_1, M_1\|1, X_1^*), (K_2, M_2\|0, X_2), (K_2, M_2\|1, X_2^*)$ in the logs. We now bound the advantage of the adversary depending on whether Bad happens.

If Bad does not happen. If Bad does not happen, because of the symmetry and the way we grant free queries, without loss of generality, suppose that there is no entry $(K_1, M_1\|0, X_1)$. In that case, the chance that $E(K_1, M_1\|0) \oplus (M_1\|0) = E(K_2, M_2\|0) \oplus (M_2\|0)$ is at most 2^{1-n}.

If Bad happens. By symmetry, without loss of generality, assume that among the four corresponding entries, $(K_1, M_1\|0, X_1)$ happens first (meaning that it is a non-granted query). For every (i, j) such that i is odd and $1 \le i < j \le 2q$, let $\mathrm{Bad}_{i,j}$ be the event that the query of $(K_1, M_1\|0, X_1)$ is the ith query, and the first query of $(K_1, M_2\|0, X_2)$ and $(K_2, M_2\|1, X_2^*)$ is the jth query. Then

$$\mathrm{Bad} = \bigcup \mathrm{Bad}_{i,j} \ .$$

Note that there are at most

$$(2q-1) + (2q-3) + \cdots + 1 = q^2$$

pairs (i, j). Fix one such pair. We now bound the adversary's advantage assuming that $\mathrm{Bad}_{i,j}$ happens. We consider the following cases.

Case 1: The jth query creates the entry $(K_2, M_2\|0, X_2)$. Then $X_2 = X_1$ with probability at most 2^{1-n}. Moreover, conditioning on $X_2 = X_1$, because the entries $(K_1, M_1\|1, X_1^*)$ and $(K_2, M_2\|1, X_2^*)$ corresponds to the granted queries, the conditional probability that $X_2^*[1:r] = X_1^*[1:r]$ is at most 2^{1-r}. Summing up over at most q^2 pairs (i, j), the chance that this case happens is at most $4q^2/2^{n+r} = 4q^2/2^\lambda$.

Case 2: The jth query creates the entry $(K_2, M_2\|1, X_2)$. Then the entries $(K_1, M_1\|1, X_1^*)$ and $(K_2, M_2\|0, X_2)$ corresponds to the granted queries. Thus the chance that $(X_1^*[1:r], X_2) = (X_2^*[1:r], X_1)$ is at most $4/2^\lambda$. Summing up over at most q^2 pairs (i, j), the chance that this case happens is at most $4q^2/2^\lambda$.

Wrapping up. Summing up all cases,

$$\mathbf{Adv}^{\mathrm{coll}}_{\mathsf{DM2}[E,\lambda]}(\mathcal{A}) \le \frac{8q^2}{2^\lambda} + \frac{2}{2^n}$$

as claimed. □

PRF security of DM2. The following result shows that if we model E as a good PRF then DM2 is also a good PRF.

Proposition 4. *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Let $n \leq \lambda \leq 2n$. Then for any adversary $\mathcal{A}$ that makes at most q queries, we can construct an adversary $\mathcal{B}$ of about the same time and $2q$ queries such that*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{DM2}[E,\lambda]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{B}) \ .$$

Proof. Without loss of generality, assume that $\mathcal{A}$ does not repeat a prior query. Consider the following sequence of games. Game G_0 is game $\mathbf{G}^{\mathsf{prf}}_{\mathsf{DM2}[E,\lambda]}(\mathcal{A})$ with challenge bit 1. Game G_1 is identical to game G_0, except that each call to $E(K_i, \cdot)$ is replaced with a corresponding call to a truly random function $f_i : \{0,1\}^n \to \{0,1\}^n$. To bound the gap between the two games, we construct an adversary $\mathcal{B}$ attacking the (multi-user) PRF security of E as follows. It runs $\mathcal{A}$ and simulates game G_0, but each call to $E(K_i, \cdot)$ is replaced by a corresponding call to $\textsc{Eval}(i, \cdot)$. Then

$$\mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{B}) = \Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})] \ .$$

Let G_2 be game $\mathbf{G}^{\mathsf{prf}}_{\mathsf{DM2}[E,\lambda]}(\mathcal{A})$ with challenge bit 0. We now bound the gap between G_1 and G_2 for a computationally unbounded adversary $\mathcal{A}$. Without loss of generality, assume that $\mathcal{A}$ is deterministic and never repeats a prior query. Note that in game G_1, thanks to the domain separation in $\mathsf{DM2}$, each f_i is never called on the same input twice, and thus effectively, in game G_1, for each $\textsc{Eval}$ call, adversary $\mathcal{A}$ receives a truly random answer. Likewise, in game G_2, for each $\textsc{Eval}$ call, adversary $\mathcal{A}$ receives a truly random answer. Hence

$$\Pr[G_1(\mathcal{A})] = \Pr[G_2(\mathcal{A})] \ .$$

Summing up,

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{DM2}[E,\lambda]}(\mathcal{A}) &= \Pr[G_0(\mathcal{A})] - \Pr[G_2(\mathcal{A})] \\ &= (\Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})]) + (\Pr[G_1(\mathcal{A})] - \Pr[G_2(\mathcal{A})]) \\ &= \mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{B}) \ .\end{aligned}$$

This concludes the proof. □

5 Committing Concealer

In this section, we formalize a new primitive that we call a *committing concealer*. Functionality wise, a committing concealer $\mathsf{C} : \mathcal{K} \times \{0,1\}^m \to \{0,1\}^m$ is simply a blockcipher on $\{0,1\}^m$, with C^{-1} denoting its inverse. But traditionally a blockcipher is only secure if it is a strong PRP, but we will weaken this security goal in order to allow more efficient constructions. Looking ahead to our Feistel-based approach to committing concealers, we'll show a weaker security goal that allows us to get by with a two-round Feistel network, rather than the four rounds that would be required to achieve security as a strong PRP [21].

HIDING SECURITY. Our weaker security goal is what we call hiding security. It requires a committing concealer be a *one-time* strong PRP, meaning that

Game $\mathbf{G}^{\mathrm{hide}}_{\mathsf{C}}(\mathcal{A})$	$\textsc{Enc}(i, M)$
$v \leftarrow 0$; $b \leftarrow\!\!\$\ \{0,1\}$; $\mathit{Used} \leftarrow \emptyset$	If $i \notin \{1, \ldots, v\} \backslash \mathit{Used}$ return $\perp$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Dec}}$	$C_1 \leftarrow \mathsf{C}(K_i, M)$; $C_0 \leftarrow\!\!\$\ \{0,1\}^{\lvert M\rvert}$
Return $(b' = b)$	$\mathit{Used} \leftarrow \mathit{Used} \cup \{i\}$; Return C_b
$\textsc{New}()$	$\textsc{Dec}(i, C)$
$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k$	If $i \notin \{1, \ldots, v\} \backslash \mathit{Used}$ return $\perp$
	$M_1 \leftarrow \mathsf{C}^{-1}(K_i, C)$; $M_0 \leftarrow\!\!\$\ \{0,1\}^{\lvert M\rvert}$
	$\mathit{Used} \leftarrow \mathit{Used} \cup \{i\}$; Return M_b

Fig. 11: Game defining hiding security of C. The game maintains a set *Used* of users that $\mathcal{A}$ has queried.

Game $\mathbf{G}^{\mathrm{bind}}_{\mathsf{C},\mathsf{encode}}(\mathcal{A})$

$(K_1, M_1, K_2, M_2) \leftarrow\!\!\$\ \mathcal{A}$

Return $\big(K_1 \neq K_2\big) \wedge \big(\mathsf{C}(K_1, \mathsf{encode}(M_1)) = \mathsf{C}(K_2, \mathsf{encode}(M_2))$

Fig. 12: Game defining binding security of C.

an adversary is allowed only a single query per user. In particular, define the advantage of an adversary $\mathcal{A}$ breaking the hiding security of C as

$$\mathbf{Adv}^{\mathrm{hide}}_{\mathsf{C}}(\mathcal{A}) = \Pr[\mathbf{G}^{\mathrm{hide}}_{\mathsf{C}}(\mathcal{A})] \ ,$$

where game $\mathbf{G}^{\mathrm{hide}}_{\mathsf{C}}(\mathcal{A})$ is defined in Fig. 11.

Binding security. Let $s \leq m$ be an integer, and let $\{0,1\}^{\leq m-s}$ denote the set of bit strings whose length is at most $m - s$. Let $\mathsf{encode} : \{0,1\}^{\leq m-s} \to \{0,1\}^m$ be a function. Define the binding advantage of an adversary $\mathcal{A}$ against C with respect to encode as

$$\mathbf{Adv}^{\mathrm{bind}}_{\mathsf{C},\mathsf{encode}}(\mathcal{A}) = \Pr[\mathbf{G}^{\mathrm{bind}}_{\mathsf{C},\mathsf{encode}}(\mathcal{A})] \ ,$$

where game $\mathbf{G}^{\mathrm{bind}}_{\mathsf{C},\mathsf{encode}}(\mathcal{A})$ is defined in Fig. 12. Informally, we want the ciphertext of C to be a commitment of the key, if we restrict to messages in $\{\mathsf{encode}(X) \mid X \in \{0,1\}^{\leq m-s}\}$.

Relation to prior work. Bellare and Hoang [5] recently consider how to add committing security to a standard AE scheme without expanding the ciphertext length. Their method requires that the scheme is tag-based, meaning that the ciphertext consists of a ciphertext core C^* and a tag T, and one can recover the message from just C^* (but of course without authenticity guarantees). As such, their method doesn't work for SIV constructions [26] (because the tag is needed for decryption), or EtE constructions (because there is no tag).

The work of Bellare and Hoang relies on an invertible PRF (IPF) that is also collision-resistant. Their construction of collision-resistant IPF encodes the

message and then enciphers it with what is implicitly a committing concealer. In Section 5.2 we will study this committing concealer construction.

5.1 Committing Concealer from Ideal Cipher

As a warmup, we show that a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ can be used directly as a committing concealer if we model E as an ideal cipher. Still, if we want E to have strong binding security, the block length n needs to be at least 256 bits, meaning that we can't instantiate E from AES.

HIDING SECURITY. If E is modeled as a strong PRP then it obviously has good hiding security. We state the formal result for completeness.

Proposition 5. *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Then for any adversary $\mathcal{A}$,*

$$\mathbf{Adv}^{\mathrm{hide}}_{E}(\mathcal{A}) \le \mathbf{Adv}^{\pm\mathrm{prp}}_{E}(\mathcal{A}) \enspace .$$

BINDING SECURITY. The following result, from Bellare and Hoang [5], shows that if we model E as an ideal cipher then it also has good binding security, for any encoding mechanism. Due to the term $q^2/2^n$, if we aim for strong binding security, we need $n \ge 256$, meaning we need to instantiate E from, say Rijndael-256.

Proposition 6 ([5]). *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we model as an ideal cipher. Let $\mathsf{encode} : \{0,1\}^{\le n-s} \to \{0,1\}^n$ be a function. Then for any adversary $\mathcal{A}$ that makes at most q ideal-cipher queries,*

$$\mathbf{Adv}^{\mathrm{bind}}_{E,\mathsf{encode}}(\mathcal{A}) \le \frac{4q}{2^s} + \frac{2q^2}{2^n} \enspace .$$

5.2 Committing Concealer from Two-Round Feistel

In this section, we show how to build a committing concealer FF from a two-round Feistel network. The round functions are built on top of a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ that we can instantiate via AES.

THE FF CONSTRUCTION. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Let $\ell < n$ be an integer. Define $\mathsf{pad}(\cdot,0) : \{0,1\}^n \to \{0,1\}^n$ via $\mathsf{pad}(X,0) = X[1:n-1]\|0$, and define $\mathsf{pad}(\cdot,1) : \{0,1\}^\ell \to \{0,1\}^n$ via $\mathsf{pad}(Y,1) = Y\|1^{n-\ell}$. Note that pad is a domain separation in the sense that $\mathsf{pad}(X,0) \ne \mathsf{pad}(Y,1)$ for any X, Y.

The committing concealer $\mathsf{FF}[E,\ell]$ has message space $\{0,1\}^{n+\ell}$. It is a two-round unbalanced Feistel network, where the left-hand side is n bits, and the right-hand side is ℓ bits. The round functions are based on the Davies-Meyer construction of E. See Fig. 13 for the code and also an illustration. This committing concealer is implicit in the recent work of Bellare and Hoang [5].

HIDING SECURITY OF FF. The following result shows that FF has good hiding security, assuming that E is a good PRF.

$\underline{\mathsf{C}(K, L\|R)}$ // $\|L\| = n$	$\underline{\mathsf{C}^{-1}(K, A\|B)}$ // $\|A\| = n$
$U \leftarrow \mathsf{pad}(L, 0)$	$V \leftarrow \mathsf{pad}(B, 1)$
$B \leftarrow \big(E_K(U) \oplus U\big)[1 : \|R\|] \oplus R$	$L \leftarrow \big(E_K(V) \oplus V\big) \oplus A$
$V \leftarrow \mathsf{pad}(B, 1)$	$U \leftarrow \mathsf{pad}(L, 0)$
$A \leftarrow \big(E_K(V) \oplus V\big) \oplus L$	$R \leftarrow (E_K(U) \oplus U)[1 : \|B\|] \oplus B$
Return $A\|B$	Return $L\|R$

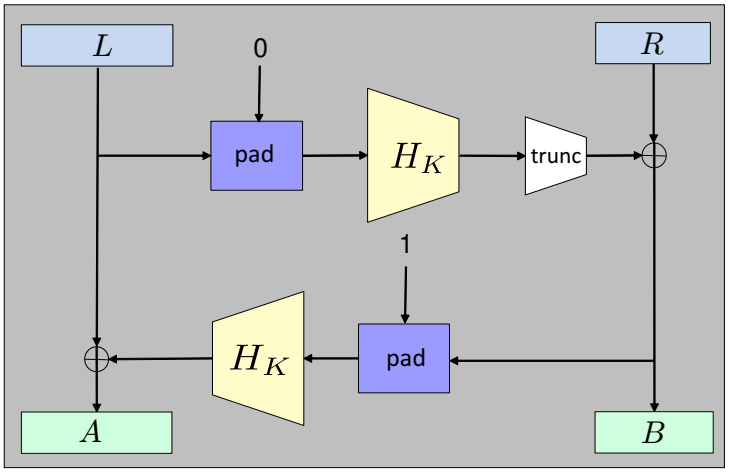

Fig. 13: The committing concealer $\mathsf{C} = \mathsf{FF}[E, \ell]$. In the illustration, the function H denotes the Davies-Meyer construction on E, meaning $H(K, M) = E_K(M) \oplus M$.

Proposition 7. *Let n, ℓ be integers such that $\ell < n$. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Then for an adversary $\mathcal{A}$ that makes at most q queries, we can construct an adversary $\mathcal{B}$ of about the same time and $2q$ queries, with two queries per user, such that*

$$\mathbf{Adv}^{\mathrm{hide}}_{\mathsf{FF}[E,\ell]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{B}) \ .$$

Proof. Consider the following sequence of games. Game G_0 coincides with game $\mathbf{G}^{\mathrm{hide}}_{\mathsf{C}}(\mathcal{A})$ with challenge bit 1. Game G_1 is identical to game G_1, except that each $E(K_i, \cdot)$ is replaced by a truly random function $f_i : \{0,1\}^n \to \{0,1\}^n$. To bound the gap between the two games, we construct an adversary $\mathcal{B}$ attacking the (multi-user) PRF security of E as follows. It runs $\mathcal{A}$ and simulates game G_0, but each call to $E(K_i, \cdot)$ is replaced by a corresponding call to $\textsc{Eval}(i, \cdot)$. Then

$$\mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{B}) = \Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})] \ .$$

Note that in game G_1, each f_i is never run on the same input twice, thanks to the domain separation in FF and the requirement that the adversary can make a single query per user. Then we can rewrite game G_1 as game G_2 in Fig. 14, and the two games are equivalent. Note that effectively, in game G_2, each $\textsc{Enc}/\textsc{Dec}$

Game $G_2(\mathcal{A})$	NEW()
$v \leftarrow 0$; $\mathit{Used} \leftarrow \emptyset$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\text{NEW},\text{ENC},\text{DEC}}$	$v \leftarrow v + 1$
Return $(b' = 1)$	
ENC$(i, L\|R)$	DEC$(i, A\|B)$
If $i \notin \{1, \ldots, v\} \setminus \mathit{Used}$ return $\perp$	If $i \notin \{1, \ldots, v\} \setminus \mathit{Used}$ return $\perp$
$U \leftarrow \mathsf{pad}(L, 0)$; $X \leftarrow\!\!\$\ \{0,1\}^n$	$V \leftarrow \mathsf{pad}(B, 1)$; $Y \leftarrow\!\!\$\ \{0,1\}^n$
$B \leftarrow (X \oplus U)[1 : \lvert R\rvert] \oplus R$	$L \leftarrow (Y \oplus V) \oplus A$
$V \leftarrow \mathsf{pad}(B, 1)$; $Y \leftarrow\!\!\$\ \{0,1\}^n$	$U \leftarrow \mathsf{pad}(L, 0)$; $X \leftarrow\!\!\$\ \{0,1\}^n$
$A \leftarrow (Y \oplus V) \oplus L$	$R \leftarrow (X \oplus U)[1 : \lvert B\rvert] \oplus B$
$\mathit{Used} \leftarrow \mathit{Used} \cup \{i\}$; Return $A\|B$	$\mathit{Used} \leftarrow \mathit{Used} \cup \{i\}$; Return $L\|R$

Fig. 14: Game G_2 in the proof of Proposition 7.

query returns a truly random answer. Thus game G_2 is equivalent to game $\mathbf{G}^{\text{hide}}_{\mathsf{C}}(\mathcal{A})$ with challenge bit 0. Summing up,

$$\begin{aligned}\mathbf{Adv}^{\text{hide}}_{\mathsf{FF}[E,\ell]}(\mathcal{A}) &= \Pr[G_0(\mathcal{A})] - \Pr[G_2(\mathcal{A})] \\ &= (\Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})]) + (\Pr[G_1(\mathcal{A})] - \Pr[G_2(\mathcal{A})]) \\ &= \mathbf{Adv}^{\text{prf}}_{E}(\mathcal{B}) \ .\end{aligned}$$

This concludes the proof. □

BINDING SECURITY OF FF. Let $1 \le t \le n$. Let $\mathsf{encode} : \{0,1\}^{\le n-t} \to \{0,1\}^{n+\ell}$ be a function such that $\mathsf{encode}(X)$ must end with $0^{\ell+1}$. The following result of Bellare and Hoang [5] shows that $\mathsf{FF}[E, \ell]$ has about $(\ell - \log_2(n))$ bits of binding security in the ideal-cipher model.

Proposition 8 ([5]). *Let n, ℓ be integers such that $n \ge 32$ and $\ell < n$. 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* encode *be as above. Then for an adversary $\mathcal{A}$ that makes at most q ideal-cipher queries,*

$$\mathbf{Adv}^{\text{bind}}_{\mathsf{FF}[E,\ell],\mathsf{encode}}(\mathcal{A}) \le \frac{4(n+\ell)q + 5}{2^\ell} \ .$$

6 A Committing Transform for Wideblock TBC

Let TE be a wideblock TBC with message space $\{0,1\}^{\ge k}$, key space $\{0,1\}^k$, and tweak space $\mathcal{T}$. Our goal is to turn it into a wideblock TBC $\overline{\mathsf{TE}}$ of the same tweak space and key space such that using $\overline{\mathsf{TE}}$ in the EtE transform gives a scheme of both CMT-1 and RAE security. We achieve this via the Encipher-with-Concealer (EwC) transform below.

$\underline{\mathsf{EwC}[\mathsf{TE},\mathsf{C},H,G].\mathsf{Enc}(K,T,M)}$	$\underline{\mathsf{EwC}[\mathsf{TE},\mathsf{C},H,G].\mathsf{Dec}(K,T,C)}$
$P\|\|S \leftarrow M$ // $\|S\| = m$	$C_1\|\|C_2 \leftarrow C$ // $\|C_2\| = m$
$J \leftarrow H_K([0]_r)[1:n]$; $U \leftarrow P \oplus G_J(S)$	$J \leftarrow H_K([0]_r)[1:n]$
$J^* \leftarrow H_K([1]_r)[1:n]$	$J^* \leftarrow H_K([1]_r)[1:n]$; $V \leftarrow C_1 \oplus G_{J^*}(C_2)$
$L \leftarrow H_K([2]_r)$; $I \leftarrow H_K([3]_r)$	$L \leftarrow H_K([2]_r)$; $I \leftarrow H_K([3]_r)$
$V \leftarrow \mathsf{TE}.\mathsf{Enc}(L,T,U)$	$U \leftarrow \mathsf{TE}.\mathsf{Dec}(L,T,V)$
$R \leftarrow H_I(U[1:r] \oplus V[1:r])$	$R \leftarrow H_I(U[1:r] \oplus V[1:r])$
$C_2 \leftarrow \mathsf{C}(R,S)$; $C_1 \leftarrow V \oplus G_{J^*}(C_2)$	$S \leftarrow \mathsf{C}^{-1}(R,C_2)$; $P \leftarrow U \oplus G_J(S)$
Return $C_1\|\|C_2$	Return $P\|\|S$

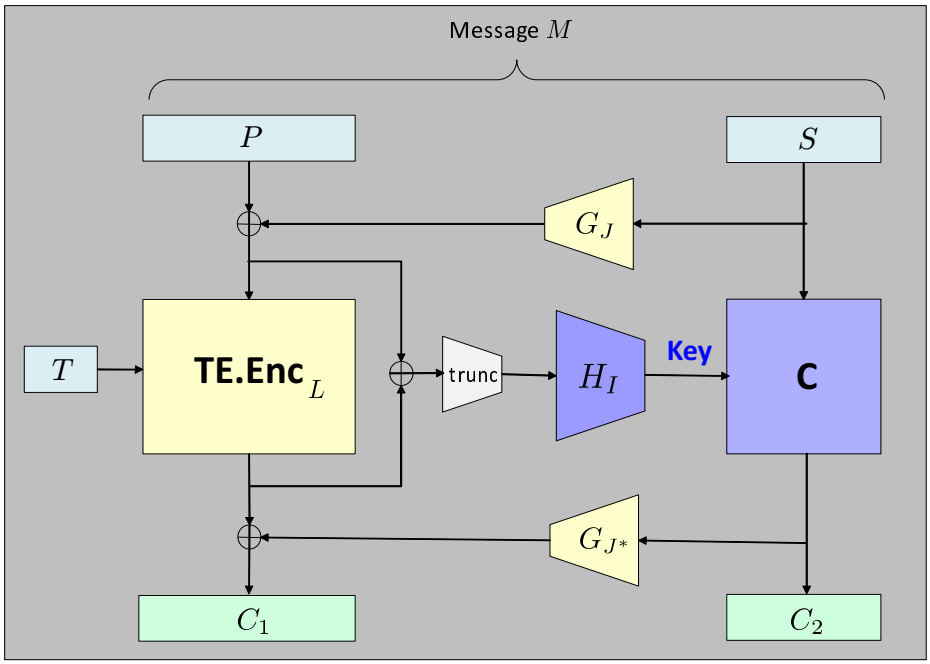

Fig. 15: The EwC transform. We omit the derivation of subkeys J, J^*, L, I in the illustration.

THE EwC TRANSFORM. Let C be a committing concealer of message space $\{0,1\}^m$ and key space $\{0,1\}^k$. Let $G : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$ be a c-AXU hash function, with $n \le k$. Let $H : \{0,1\}^k \times \{0,1\}^r \to \{0,1\}^k$ be a collision-resistant PRF, with $r \le n$. For an integer $i \in \{0,\ldots,2^r-1\}$, let $[i]_r$ denote the r-bit representation of i. For two strings X and Y with $|X| \le |Y|$, we write $X \oplus Y$ to denote $(X\|0^{|Y|-|X|}) \oplus Y$. The code of the transform $\mathsf{EwC}[\mathsf{TE},\mathsf{C},H,G]$ is given in Fig. 15. The scheme has message space $\{0,1\}^{\ge m+k}$.

Informally, we use a four-round Feistel-like structure, where the right-hand side is m bits. Following Naor and Reingold [24], the first and last rounds are implemented via the AXU hash G (whose output is padded with 0's). In the second

round, we encipher the intermediate left-hand side U via $V \leftarrow \mathsf{TE.Enc}(L, T, U)$, where the subkey L is derived via $L \leftarrow H_K([2]_r)$. In the third round, we derive a one-time key $R \leftarrow H_I(U[1:r] \oplus V[1:r])$ for C, where $I \leftarrow H_K([3]_r)$, and use C to encipher the intermediate right-hand side.

Since we only need to use the r-bit prefix of the output of TE for C, on long messages, the evaluation of TE and C can be parallelized, for off-the-shelf constructions of TE such as AEZ [18] or HCTR2 [14]. Conversely, on decryption, this allows fast rejection of invalid ciphertexts, which is important to resist denial-of-service attacks. The subkeys J, J^*, L, I can be cached and the cost of their derivation can be ignored if EwC is used in a stand-alone way. Still, if we compose it with the HtE transform in Section 3.2 then we have to account for this key derivation. However, in that case, the overhead of EwC is negligible compared to the hashing cost in HtE.

Discussion. Structurally, EwC resembles the Hash-CTR-Hash (HCH) method [10] but there are nuances in the design. Here the message is split into two *uneven* halves, one of $|M| - O(1)$ bits, and the other just $O(1)$ bits. HCH runs the universal hash on the big half, meaning the hashing cost is $\Theta(|M|)$. In contrast, EwC runs the universal hash on the small half, and thus the hashing cost is merely $O(1)$. On the other hand, while both have to encrypt $\Theta(|M|)$ bits with their base encryption schemes, EwC has to use the expensive TE but HCH only needs to run the cheap CTR.

Instantiations. If we want to use AES, we can instantiate H from the $\mathsf{DM2}$ construction in Section 4, with AES-256 as the underlying blockcipher, and instantiate G from GHASH or POLYVAL [17]. This means $k = 256$ and $r = 127$ and $n = 128$ and $c = 1.5$. The committing concealer C can be built from the FF construction in Section 5.2, again on AES-256. If we want to achieve around $\ell - 8$ bits of CMT-1 security, with $\ell < 128$, the input length of C should be $m = \ell + 128$. In addition, in EtE, if we want CMT-1 security, the minimum expansion must be $\ell + 1$ bits. Moreover, EwC only uses AES-256 in the forward direction. The overhead of EwC (assuming that the subkeys are cached) is four multiplications in $\mathsf{GF}(2^{128})$ and four AES-256 calls plus an AES key setup. If AES-NI is available, the AES cost is approximately three sequential AES-256, because a good implementation can hide the key setup cost, and running two parallel AES calls in $\mathsf{DM2}$ has the same cost as one.[6]

If we instead have a blockcipher of 256-bit block length, say Rijndael-256, the instantiation is much simpler. In particular, we can instantiate H from the Davies-Meyer construction of Rijndael-256, and C directly from Rijndael-256, meaning $k = m = r = 256$. Moreover, we can pick $n = 256$ and instantiate $G_J(X)$ as $X \times J$, where $\times$ denotes the finite-field multiplication in $\mathsf{GF}(2^{256})$, meaning $c = 1$. Thus the overhead of EwC in this case is two multiplications in

[6] If we count the cost of subkey generation, we need six extra AES calls (instead of eight). In particular, since J and J^* are 128-bit long, each only needs one AES call (instead of two). These six AES are fully parallel, so running them costs as much as one AES call in AES-NI platforms.

$\mathsf{GF}(2^{256})$ and two sequential Rijndael-256 calls plus the Rijndael-256 key setup cost (that can be hidden with a good implementation if AES-NI is available).[7]

For both instantiations, in EtE, we can pad with either 10^* or 0^*.

CMT-1 SECURITY OF $\mathsf{EtE}[\mathsf{EwC}]$. Suppose that C has good binding security with respect to an encoding $\mathsf{encode} : \{0,1\}^{\le m-s} \to \{0,1\}^m$. Define the following padding mechanism in EtE. If we have a message M and want to pad $\tau \ge s$ bits to it, we parse M to $M_1 \| M_2$, with $|M_2| = m - \tau$, and then output $M_1 \| \mathsf{encode}(M_2)$. (We assume that one can efficiently recover the message X from $\mathsf{encode}(X)$ and $|X|$, so that EtE is decryptable under the padding above.) For example, if $\mathsf{encode}(X) = X \| 0^{m-|X|}$, the padding mechanism simply adds 0^τ to the message. The following result shows that $\mathsf{EtE}[\mathsf{EwC}]$ has good CMT-1 security. Intuitively, we have a commitment chain $K \xrightarrow{H} I \xrightarrow{H} R \xrightarrow{\mathsf{C}} C_2$, and thus C_2 is a commitment of K.

Theorem 9. *Let $\overline{\mathsf{TE}} = \mathsf{EwC}[\mathsf{TE}, \mathsf{C}, H, G]$ be as above. For any adversary $\mathcal{A}$, we can construct adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

$$\mathbf{Adv}^{\text{cmt-1}}_{\mathsf{EtE}[\overline{\mathsf{TE}}]}(\mathcal{A}) \le \mathbf{Adv}^{\text{coll}}_{H}(\mathcal{B}_1) + \mathbf{Adv}^{\text{bind}}_{\mathsf{C},\mathsf{encode}}(\mathcal{B}_2) \ .$$

The running time of each $\mathcal{B}_i$ is about that of $\mathcal{A}$ plus the time to run $\mathsf{EtE}[\overline{\mathsf{TE}}]$ on $\mathcal{A}$'s output.

Proof. We first construct adversary $\mathcal{B}_1$. It runs

$$\big((K, N, A, P\|S, \tau), (K^*, N^*, A^*, P^*\|S^*, \tau^*)\big) \leftarrow\!\!\$\ \mathcal{A} \ .$$

It then runs $\mathsf{EtE}[\overline{\mathsf{TE}}]$ on $(K, N, A, P\|S, \tau)$ to obtain the subkeys I and $R \leftarrow H(I, X)$. It also runs $\mathsf{EtE}[\overline{\mathsf{TE}}]$ on $(K^*, N^*, A^*, P^*\|S^*, \tau^*)$ to obtain (I^*, R^*, X^*). If $I = I^*$ then it outputs $((K, [3]_r), (K^*, [3]_r))$. Otherwise, it outputs $((I, X), (I^*, X^*))$.

Next, we construct $\mathcal{B}_2$. It runs

$$\big((K, N, A, P\|S, \tau), (K^*, N^*, A^*, P^*\|S^*, \tau^*)\big) \leftarrow\!\!\$\ \mathcal{A} \ .$$

It then runs $\mathsf{EtE}[\overline{\mathsf{TE}}]$ on those inputs to obtain subkeys R and R^*, and outputs $((R, S), (R^*, S^*))$.

For analysis, let $C_1\|C_2$ and $C_1^*\|C_2^*$ be the corresponding ciphertexts. Suppose that $\mathcal{A}$ can create a ciphertext collision, meaning $C_1 = C_1^*$ and $C_2 = C_2^*$ and $K \ne K^*$. If $R = R^*$ then $\mathcal{B}_1$ also creates a collision. If $R \ne R^*$ then $\mathcal{B}_2$ creates a collision. Hence

$$\mathbf{Adv}^{\text{cmt-1}}_{\mathsf{EtE}[\overline{\mathsf{TE}}]}(\mathcal{A}) \le \mathbf{Adv}^{\text{coll}}_{H}(\mathcal{B}_1) + \mathbf{Adv}^{\text{bind}}_{\mathsf{C},\mathsf{encode}}(\mathcal{B}_2) \ .$$

This concludes the proof. □

STRONG TWEAKABLE-PRP SECURITY OF EwC. The following result shows that EwC is a strong tweakable-PRP. The proof is deferred to Section 8.

[7] If we count the cost of subkey generation, we need four extra (parallel) Rijndael-256 calls.

$\mathsf{EwC2}[\mathsf{TE},\mathsf{C},\mathsf{KD},H,G].\mathsf{Enc}(K,T,M)$	$\mathsf{EwC2}[\mathsf{TE},\mathsf{C},\mathsf{KD},H,G].\mathsf{Dec}(K,T,C)$
$P\|S \leftarrow M$ // $\|S\| = m$	$C_1\|C_2 \leftarrow C$ // $\|C_2\| = m$
$J \leftarrow \mathsf{KD}_K([0]_r)[1:n]$; $U \leftarrow P \oplus G_J(S)$	$J \leftarrow \mathsf{KD}_K([0]_r)[1:n]$
$J^* \leftarrow \mathsf{KD}_K([1]_r)[1:n]$	$J^* \leftarrow \mathsf{KD}_K([1]_r)[1:n]$; $V \leftarrow C_1 \oplus G_{J^*}(C_2)$
$L \leftarrow \mathsf{KD}_K([2]_r)$; $I \leftarrow \mathsf{KD}_K([3]_r)$	$L \leftarrow \mathsf{KD}_K([2]_r)$; $I \leftarrow \mathsf{KD}_K([3]_r)$
$V \leftarrow \mathsf{TE}.\mathsf{Enc}(L,T,U)$	$U \leftarrow \mathsf{TE}.\mathsf{Dec}(L,T,V)$
$R \leftarrow H_I(T\|(U[1:r] \oplus V[1:r]))$	$R \leftarrow H_I(T\|(U[1:r] \oplus V[1:r]))$
$C_2 \leftarrow \mathsf{C}(R,S)$; $C_1 \leftarrow V \oplus G_{J^*}(C_2)$	$S \leftarrow \mathsf{C}^{-1}(R,C_2)$; $P \leftarrow U \oplus G_J(S)$
Return $C_1\|C_2$	Return $P\|S$

Fig. 16: The $\mathsf{EwC2}$ transform.

Theorem 10. *Let $\overline{\mathsf{TE}} = \mathsf{EwC}[\mathsf{TE},\mathsf{C},H,G]$ be as above. For any adversary $\mathcal{A}$ making q queries, with at most B queries per user, we can construct adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ of at most $4q$ queries such that*

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\overline{\mathsf{TE}}}(\mathcal{A}) \le 2\mathbf{Adv}^{\mathrm{prf}}_{H}(\mathcal{B}_1) + \mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\mathsf{TE}}(\mathcal{B}_2) + \mathbf{Adv}^{\mathrm{hide}}_{\mathsf{C}}(\mathcal{B}_3) + \frac{6cqB}{2^r} + \frac{6qB}{2^m} \ .$$

The running time of each $\mathcal{B}_i$ is about that of $\mathcal{A}$ plus the time to run $\overline{\mathsf{TE}}$ on $\mathcal{A}$'s queries.

Another way to get CMT security. To achieve full CMT security, one can combine the Hash-then-Encrypt transform with EwC. However, this means that we now can no longer cache the subkeys (I,J,J^*,L) and have to account for the cost of subkey derivation. Moreover, we'll have to pay the key setup cost for TE. To improve the speed, we give another wideblock TBC $\mathsf{EwC2}$ such that $\mathsf{EtE}[\mathsf{EwC2}]$ directly achieves CMT security; the code is in Fig. 16. This resembles EwC but here the subkeys are derived with a collision-resistant PRF $\mathsf{KD} : \{0,1\}^k \times \{0,1\}^r \to \{0,1\}^k$ (that we'll again instantiate via the $\mathsf{DM2}$ construction), and the hash function $H : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^k$ needs to hash $(I,T\|(U[1:r] \oplus V[1:r]))$ instead of just $(I,U[1:r] \oplus V[1:r])$. Because the hash H takes a long input now, we have to instantiate it via, say (truncated) SHA-512.

In $\mathsf{EwC2}$ we can cache the subkeys (I,J,J^*,L) and avoid the key setup cost for TE. We also manage to eliminate the use of $\mathsf{DM2}$ to generate the subkey for C. The drawback is that we need to process the tweak T twice, one by the hash H, and another via TE. However, when we use TE in EtE, the tweak T is (N,A,τ), which is usually short. Moreover, in modern wideblock TBC constructions like AEZ or HCTR2, the tweak is processed via an AXU hash, and thus its processing is cheap.

CMT security of $\mathsf{EtE}[\mathsf{EwC2}]$. The following Theorem 11 formally confirms that $\mathsf{EtE}[\mathsf{EwC2}]$ does achieve CMT security. Informally, when we use $\mathsf{EwC2}$ within EtE (meaning $T = (N,A,\tau)$), we have the committing chain $(K,N,A,\tau) \xrightarrow{\mathsf{KD}} (I,N,A,\tau) \xrightarrow{H} R \xrightarrow{\mathsf{C}} C_2$.

Theorem 11. *Let* $\overline{\mathsf{TE}} = \mathsf{EwC}[\mathsf{TE}, \mathsf{C}, \mathsf{KD}, H, G]$ *be as above. For any adversary* $\mathcal{A}$*, we can construct adversaries* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ *such that*

$$\mathbf{Adv}^{\mathrm{cmt}}_{\mathsf{EtE}[\overline{\mathsf{TE}}]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{coll}}_{\mathsf{KD}}(\mathcal{B}_1) + \mathbf{Adv}^{\mathrm{coll}}_{H}(\mathcal{B}_2) + \mathbf{Adv}^{\mathrm{bind}}_{\mathsf{C},\mathsf{encode}}(\mathcal{B}_3) \ .$$

The running time of each $\mathcal{B}_i$ *is about that of* $\mathcal{A}$ *plus the time to run* $\mathsf{EtE}[\overline{\mathsf{TE}}]$ *on* $\mathcal{A}$*'s output.*

Proof. We first construct adversary $\mathcal{B}_1$. It runs

$$\big((K, N, A, P \| S, \tau), (K^*, N^*, A^*, P^* \| S^*, \tau^*)\big) \leftarrow\!\!\$\ \mathcal{A} \ .$$

It then outputs $((K, [3]_r), (K^*, [3]_r))$.

Next, we construct $\mathcal{B}_2$. It runs

$$\big((K, N, A, P \| S, \tau), (K^*, N^*, A^*, P^* \| S^*, \tau^*)\big) \leftarrow\!\!\$\ \mathcal{A} \ .$$

It then runs $\mathsf{EtE}[\overline{\mathsf{TE}}]$ on those inputs to obtain (I, U, V) and (I^*, U^*, V^*). Let $X = U[1:r] \oplus V[1:r]$ and $X^* = U^*[1:r] \oplus V^*[1:r]$. It then outputs

$$\Big(\big(I, (N, A, \tau) \| X\big), \big(I^*, (N^*, A^*, \tau^*) \| X^*\big)\Big) \ .$$

Finally, we construct $\mathcal{B}_3$. It runs

$$\big((K, N, A, P \| S, \tau), (K^*, N^*, A^*, P^* \| S^*, \tau^*)\big) \leftarrow\!\!\$\ \mathcal{A} \ .$$

It then runs $\mathsf{EtE}[\overline{\mathsf{TE}}]$ on those inputs to obtain subkeys R and R^* of C, and outputs $((R, S), (R^*, S^*))$.

For analysis, let $C_1 \| C_2$ and $C_1^* \| C_2^*$ be the corresponding ciphertexts. Suppose that $\mathcal{A}$ can create a ciphertext collision, meaning $C_1 = C_1^*$ and $C_2 = C_2^*$ and $K \ne K^*$. If $I = I^*$ then $\mathcal{B}_1$ creates a collision. If $I \ne I^*$ but $R = R^*$ then $\mathcal{B}_2$ creates a collision. Finally, if $R \ne R^*$ then $\mathcal{B}_3$ creates a collision. Hence

$$\mathbf{Adv}^{\mathrm{cmt}}_{\mathsf{EtE}[\overline{\mathsf{TE}}]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{coll}}_{\mathsf{KD}}(\mathcal{B}_1) + \mathbf{Adv}^{\mathrm{coll}}_{H}(\mathcal{B}_2) + \mathbf{Adv}^{\mathrm{bind}}_{\mathsf{C},\mathsf{encode}}(\mathcal{B}_3) \ .$$

This concludes the proof. □

Strong tweakable-PRP security of EwC2. The following result shows that EwC2 is a strong tweakable-PRP. The proof is similar to that of Theorem 12.

Theorem 12. *Let* $\overline{\mathsf{TE}} = \mathsf{EwC}[\mathsf{TE}, \mathsf{C}, \mathsf{KD}, H, G]$ *be as above. For any adversary* $\mathcal{A}$ *making* q *queries, with at most* B *queries per user, we can construct adversaries* $\mathcal{B}_1$*,* $\mathcal{B}_2$*,* $\mathcal{B}_3$*, and* $\mathcal{B}_4$ *of at most* $4q$ *queries such that*

$$\mathbf{Adv}^{\widetilde{\pm\mathrm{prp}}}_{\overline{\mathsf{TE}}}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{KD}}(\mathcal{B}_1) + \mathbf{Adv}^{\mathrm{prf}}_{H}(\mathcal{B}_2) + \mathbf{Adv}^{\widetilde{\pm\mathrm{prp}}}_{\mathsf{TE}}(\mathcal{B}_3)$$
$$+ \mathbf{Adv}^{\mathrm{hide}}_{\mathsf{C}}(\mathcal{B}_4) + \frac{6cqB}{2^r} + \frac{6qB}{2^m} \ .$$

The running time of each $\mathcal{B}_i$ *is about that of* $\mathcal{A}$ *plus the time to run* $\overline{\mathsf{TE}}$ *on* $\mathcal{A}$*'s queries.*

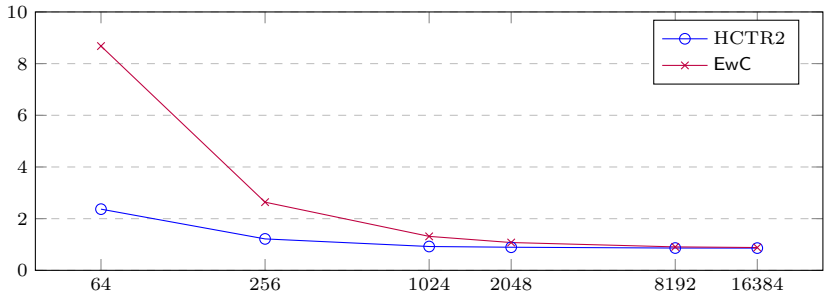

Fig. 17: Performance of EwC transform. The graph shows CPU cycles-per-byte (y-axis, lower is better) for encrypting messages of varying sizes (x-axis, in bytes) on a x86_64 processor.

7 Performance

Schemes. We start from HCTR2 [14] as our baseline wideblock tweakable block cipher. We use HCTR2 on AES-256 since the Accordion call targets 256-bit (key-recovery) security, but note that one can use EwC on HCTR2-AES-128 as well. We implement EwC with the committing concealer C being the FF construction from Section 5.2, the hash H being the DM2 construction from Section 4, the AXU hash G being POLYVAL [17], and set $\ell = 120$.

Experimental setup. We evaluated the schemes on a range of message lengths from 64 bytes to 16384 bytes. For each message length, after warming up the operation 2048 invocations, we measured the elapsed time in cycles for 2048 invocations. We divided this elapsed time by the number of encrypted bytes to compute the number of *cycles per byte*. To minimize variance, we repeated this eight times, checked that the standard deviation across the repetitions was less than 0.05 cycles per byte, and took the mean.

Our benchmarking program used the implementation accompanying the paper [14] and was executed on an Intel i7-1360P, on a specified core running at 2.4GHz with frequency scaling disabled.

Results. The statistics is given in Fig. 17. Overall, the overhead is significant for small data, but becomes negligible for messages bigger than 1KB.

8 Proof of Theorem 12

8.1 A Technique To Simplify Game-based Proofs

Our proof relies on a novel use of the H-coefficient technique [25, 12] to simplify game-based proofs [8]. This technique is generic, and in this section, we will elaborate on its details.

The setting. Suppose we have a construction based on an indistinguishability-based primitive Π, such as a PRF, an encryption scheme, or in our case, a

committing concealer. The security notion of this construction involves bounding the gap between the real game G_0 (where Π is used) and an ideal game G_2. The standard approach is to (i) define an intermediate game G_1 where Π is replaced by its ideal reference, (ii) give a reduction to bound the gap between G_0 and G_1, and (iii) bound the (information-theoretic) gap between G_1 and G_2. This approach doesn't work if in step (ii), the reduction only works as long as G_0 doesn't set a flag bad. For example, in our case, we have to derive the key for the committing concealer, and the key is uniformly random only when G_0 doesn't set bad.

To deal with the situation above, define G_1 as the analogue of G_0 where Π is replaced by its ideal reference; this means that G_1 also includes a flag bad. The reduction still works as usual, but keeps track of the flag bad. If bad is set, then the reduction returns 1, suggesting that it's interacting with the real world. Otherwise, it follows the guess of the adversary.

In this case, the reduction advantage Δ doesn't bound $\Pr[G_0] - \Pr[G_1]$. Instead, $\Pr[G_0] - \Pr[G_1] \le \Delta + \Pr[G_1 \text{ sets } \mathsf{bad}]$. This means that we need to bound (1) the chance that G_1 sets bad, and also (2) the gap $\Pr[G_1] - \Pr[G_2]$. Here we only consider information-theoretic bounds, meaning the adversary is computationally unbounded, and thus can be assumed to be deterministic. There are many techniques [25, 12, 19, 15] for simplifying the analysis in (2), but none considers (1). In this section, we show how to extend the H-coefficient technique [25, 12] to *simultaneously* bound both (1) and (2). That is, not only can we simplify the analysis of (1), but we can kill two birds with one stone.

The H-coefficient technique. Following [19], it is convenient to consider interactions of a distinguisher $\mathcal{A}$ with an abstract system $\mathbf{S}$ which answers $\mathcal{A}$'s queries. This system takes inputs and produces outputs, and is randomized and possibly stateful. The interaction between an adversary $\mathcal{A}$ and a system $\mathbf{S}$ defines a *transcript* $\theta = \big((u_1, v_1), \ldots, (u_q, v_q)\big)$ containing the ordered sequence of query-answer pairs. Let $\mathsf{p}_{\mathbf{S}}(\theta)$ be the probability that if the adversary queries $u_1, \ldots, u_q$ in that order, the answers will be $v_1, \ldots, v_q$ respectively.

In the H-coefficient technique, one wants to bound the distinguishing advantage of a real system $\mathbf{S}_{\text{real}}$ and an ideal one $\mathbf{S}_{\text{ideal}}$. The adversary's interactions with those systems define transcripts $\mathcal{T}_{\text{real}}$ and $\mathcal{T}_{\text{ideal}}$, respectively. The following result bounds the distinguishing advantage of $\mathcal{A}$.

Lemma 13. [25, 12] *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 $1 - \frac{\mathsf{p}_{\mathbf{S}_{\text{real}}}(\theta)}{\mathsf{p}_{\mathbf{S}_{\text{ideal}}}(\theta)} \le \epsilon$ for every good transcript θ. Then, the advantage of $\mathcal{A}$ in distinguishing $\mathbf{S}_{\text{real}}$ and $\mathbf{S}_{\text{ideal}}$ is at most $\epsilon + \Pr[\mathcal{T}_{\text{ideal}} \text{ is bad}]$.*

Application to our setting. Recall that we have two games G_1 and G_2, and we need to bound (1) $\Pr[G_1 \text{ sets } \mathsf{bad}]$ and (2) $\Pr[G_1] - \Pr[G_2]$. To use the H-coefficient technique, we view G_1 as the real system $\mathbf{S}_{\text{real}}$, and G_2 as the ideal system $\mathbf{S}_{\text{ideal}}$, and define what's meant for transcripts to be bad. Then one can

use Lemma 13 to bound (2). Our key idea here is that it's often possible to extend the definition of bad transcripts (say adding some certain conditions, or revealing some extra information when the adversary finishes querying) so that if G_1 sets bad then the transcript must be bad. In that case, the following result shows that the *same* bound of Lemma 13 can be used to bound (1) as well. In fact, the actual bound is even slightly better.

Lemma 14. *Suppose we can partition the set of valid transcripts for the ideal system into good and bad ones such that if G_1 sets bad then the transcript must be bad. Further, suppose that there exists a constant $\epsilon \geq 0$ such that $1 - \frac{\mathsf{ps}_{\text{real}}(\theta)}{\mathsf{ps}_{\text{ideal}}(\theta)} \leq \epsilon$ for every good transcript θ. Then*

$$\Pr[G_1 \textit{ sets } \mathsf{bad}] \leq \epsilon + (1-\epsilon)\Pr[\mathcal{T}_{\text{ideal}} \textit{ is bad}] \ .$$

Proof. Since G_1's setting bad will lead to a bad transcript,

$$\Pr[G_1 \text{ sets } \mathsf{bad}] \leq \Pr[\mathcal{T}_{\text{real}} \text{ is bad}] \leq 1 - \Pr[\mathcal{T}_{\text{real}} \text{ is good}] \ . \tag{1}$$

Recall that for any good transcript θ,

$$\mathsf{ps}_{\text{real}}(\theta) \geq (1-\epsilon)\mathsf{ps}_{\text{ideal}}(\theta) \ .$$

Summing this over all good transcripts,

$$\Pr[\mathcal{T}_{\text{real}} \text{ is good}] \geq (1-\epsilon)\Pr[\mathcal{T}_{\text{ideal}} \text{ is good}] \ . \tag{2}$$

By combining Equation (1) and Equation (2), we obtain

$$\begin{aligned}\Pr[G_1 \text{ sets } \mathsf{bad}] &\leq 1 - (1-\epsilon)\Pr[\mathcal{T}_{\text{ideal}} \text{ is good}] \\ &= 1 - (1-\epsilon)\big(1 - \Pr[\mathcal{T}_{\text{ideal}} \text{ is bad}]\big) \\ &= \epsilon + (1-\epsilon)\Pr[\mathcal{T}_{\text{ideal}} \text{ is bad}] \ .\end{aligned}$$

This concludes the proof. □

DISCUSSION. Let Bad_{real} be the event that G_1 sets bad. Our approach requires extending the definition of bad transcripts so that if Bad_{real} happens, the resulting transcript will be bad. Effectively, this requires bounding the probability of an extra event $\text{Bad}_{\text{ideal}}$ in the *ideal* system when we deal with $\Pr[\mathcal{T}_{\text{ideal}} \text{ is bad}]$. This doesn't mean we gain nothing, because events in the ideal system are often much easier to analyze.

8.2 Proof of Theorem 12

Without loss of generality, assume that the adversary doesn't repeat prior queries. Moreover, if it queries $C \leftarrow \text{ENC}(i, T, M)$ then it won't later query $\text{DEC}(i, T, C)$, and vice versa.

Consider the following sequence of games. Game G_0 coincides to game $\mathbf{G}^{\pm\widetilde{\mathsf{prp}}}_{\overline{\mathsf{TE}}}(\mathcal{A})$ with challenge bit 1. Game G_1 is identical to game G_0, except that each call to $H(K_i, \cdot)$ is replaced by a corresponding call to a truly random function $f_i : \{0,1\}^r \to \{0,1\}^k$. To bound the gap between the two games, we construct an

adversary $\mathcal{D}_1$ attacking the (multi-user) PRF security of H as follows. It runs $\mathcal{A}$ and simulates game G_0, but each call to $H(K_i, \cdot)$ is replaced by a corresponding call to $\textsc{Eval}(i, \cdot)$. Then

$$\mathbf{Adv}^{\mathrm{prf}}_{H}(\mathcal{D}_1) = \Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})] \ .$$

Then in game G_1, the subkeys J_i, J_i^*, L_i, I_i will be uniformly random. In game G_2, instead of using $H(I_i, \cdot)$, we uses truly random functions $g_i : \{0,1\}^r \to \{0,1\}^k$. To bound the gap between G_2 and G_1, we construct an adversary $\mathcal{D}_2$ attacking the (multi-user) security of H as follows. It runs $\mathcal{A}$ and simulates game G_1, but each call to $H(I_i, \cdot)$ is replaced by a corresponding call to $\textsc{Eval}(i, \cdot)$. Then

$$\mathbf{Adv}^{\mathrm{prf}}_{H}(\mathcal{D}_2) = \Pr[G_1(\mathcal{A})] - \Pr[G_2(\mathcal{A})] \ .$$

So far we have two adversaries attacking the PRF security of H. We can unify them to an adversary $\mathcal{B}_1$ as follow: adversary $\mathcal{B}_1$ picks a coin $b \leftarrow\!\!\$\ \{0,1\}$, and runs $\mathcal{D}_b$. Then

$$\mathbf{Adv}^{\mathrm{prf}}_{H}(\mathcal{B}_1) = \frac{1}{2}\mathbf{Adv}^{\mathrm{prf}}_{H}(\mathcal{D}_1) + \frac{1}{2}\mathbf{Adv}^{\mathrm{prf}}_{H}(\mathcal{D}_2) \ ,$$

and thus

$$\Pr[G_0(\mathcal{A})] - \Pr[G_2(\mathcal{A})] = 2 \cdot \mathbf{Adv}^{\mathrm{prf}}_{H}(\mathcal{B}_1) \ .$$

Let $\mathcal{M}$ be the domain of TE, and let $\mathrm{LP}(\mathcal{M})$ denote the set of permutations on $\mathcal{M}$ that are length-preserving, meaning $|\pi(M)| = |M|$ for every $M \in \mathcal{M}$. Game G_3 is identical to game G_2, but calls to $\mathsf{TE.Enc}(L_i, T, \cdot)$ and $\mathsf{TE.Dec}(L_i, T, \cdot)$ are replaced by corresponding calls to $\Pi_{i,T} \leftarrow\!\!\$\ \mathrm{LP}(\mathcal{M})$ and its inverse. To bound the gap between the two games, we construct an adversary $\mathcal{B}_2$ attacking the (multi-user) strong tweakable-PRP security of TE as follows. It runs $\mathcal{A}$ and simulates game G_2, but calls to calls to $\mathsf{TE.Enc}(L_i, T, \cdot)$ and $\mathsf{TE.Dec}(L_i, T, \cdot)$ are replaced by corresponding calls to $\textsc{Enc}(i, T, \cdot)$ and $\textsc{Dec}(i, T, \cdot)$. Then

$$\mathbf{Adv}^{\widetilde{\pm\mathrm{prp}}}_{\mathsf{TE}}(\mathcal{B}_2) = \Pr[G_2(\mathcal{A})] - \Pr[G_3(\mathcal{A})] \ .$$

Game G_4 is specified in Fig. 18. It is the same as G_3 with some bookkeeping, and thus

$$\Pr[G_4(\mathcal{A})] = \Pr[G_3(\mathcal{A})] \ .$$

We are now in the setting of Section 8.1, as the keys for C are uniformly only when G_4 doesn't set bad. Let G_5 be identical to game G_4, except that the answers of C and C^{-1} are replaced by uniformly random strings. The code of game G_5 is specified in Fig. 18. To bound the gap between the two games, we construct an adversary $\mathcal{B}_3$ attacking the hiding security of C as follows. It runs $\mathcal{A}$ and simulates game G_4. For each call to $\mathsf{C}(R, \cdot)$, it creates a new user u and makes a corresponding call to $\textsc{Enc}(u, \cdot)$. Likewise, for each call to $\mathsf{C}^{-1}(R, \cdot)$, it creates a new user u^* and makes a corresponding call to $\textsc{Dec}(u^*, \cdot)$. If the simulated game sets bad then $\mathcal{B}_3$ returns 1, indicating that it's in the real world. Otherwise, it returns the same guess as $\mathcal{A}$.

Let d be the challenge bit of game $\mathbf{G}^{\mathrm{hide}}_{\mathsf{C}}(\mathcal{B}_3)$. Then on the one hand,

$$\Pr[\mathbf{G}^{\mathrm{hide}}_{\mathsf{C}}(\mathcal{B}_3) \Rightarrow \mathsf{true} \mid d = 1] \geq \Pr[G_4(\mathcal{A})] \ .$$

Game $G_4(\mathcal{A}), G_5(\mathcal{A})$	$\textsc{New}()$
$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Dec}}$	$v \leftarrow v+1$; $\mathsf{Dom}_v \leftarrow \emptyset$
Return $(b'=1)$	$J_v, J_v^* \leftarrow\!\!\$\ \{0,1\}^n$; $I_v \leftarrow\!\!\$\ \{0,1\}^k$
	For $T \in \mathcal{T}$ do $\Pi_{v,T} \leftarrow\!\!\$\ \mathrm{LP}(\mathcal{M})$
$\textsc{Enc}(i,T,P\|S)$	$\textsc{Dec}(i,T,C_1\|C_2)$
$U \leftarrow P \oplus G(J_i,S)$; $V \leftarrow \Pi_{i,T}(U)$	$V \leftarrow C_1 \oplus G(J_i^*,C_2)$; $U \leftarrow \Pi_{i,T}^{-1}(V)$
$X \leftarrow U[1:r] \oplus V[1:r]$; $R \leftarrow H(I_i,X)$	$X \leftarrow U[1:r] \oplus V[1:r]$; $R \leftarrow H(I_i,X)$
If $X \in \mathsf{Dom}_i$ then $\mathsf{bad} \leftarrow \mathsf{true}$	If $X \in \mathsf{Dom}_i$ then $\mathsf{bad} \leftarrow \mathsf{true}$
$\mathsf{Dom}_i \leftarrow \mathsf{Dom}_i \cup \{X\}$	$\mathsf{Dom}_i \leftarrow \mathsf{Dom}_i \cup \{X\}$
$C_2 \leftarrow \mathsf{C}(R,S)$; $C_2 \leftarrow\!\!\$\ \{0,1\}^m$	$S \leftarrow \mathsf{C}^{-1}(R,C_2)$; $S \leftarrow\!\!\$\ \{0,1\}^m$
$C_1 \leftarrow V \oplus G(J_i^*,C_2)$	$P \leftarrow U \oplus G(J_i,S)$
Return $C_1\|C_2$	Return $P\|S$

Fig. 18: Games G_4 and G_5 in the proof of Theorem 12. Game G_5 contains the highlighted code but game G_4 does not.

On the other hand,

$$\Pr[\mathbf{G}_{\mathsf{C}}^{\mathrm{hide}}(\mathcal{B}_3) \Rightarrow \mathsf{false} \mid d=0] \le \Pr[G_5(\mathcal{A})] + \Pr[G_5(\mathcal{A}) \text{ sets } \mathsf{bad}] \ .$$

Subtracting, we obtain

$$\Pr[G_4(\mathcal{A})] - \Pr[G_5(\mathcal{A})] \le \mathbf{Adv}_{\mathsf{C}}^{\mathrm{hide}}(\mathcal{B}_3) + \Pr[G_5(\mathcal{A}) \text{ sets } \mathsf{bad}] \ .$$

Let G_6 be game $\mathbf{G}_{\widetilde{\mathsf{TE}}}^{\pm\widetilde{\mathsf{prp}}}(\mathcal{A})$ with challenge bit 0. Using the technique in Section 8.1, we obtain the following result; the proof is is deferred to Section 8.3.

Lemma 15. *Let G_5 and G_6 be as above. Then*

$$\Pr[G_5(\mathcal{A}) \textit{ sets } \mathsf{bad}] + (\Pr[G_5(\mathcal{A})] - \Pr[G_6(\mathcal{A})]) \le \frac{2qB}{2^m} + \frac{6cqB}{2^r} \ .$$

Summing up,

$$\begin{aligned}
\mathbf{Adv}_{\widetilde{\mathsf{TE}}}^{\pm\widetilde{\mathsf{prp}}}(\mathcal{A}) &= \Pr[G_0(\mathcal{A})] - \Pr[G_6(\mathcal{A})] \\
&= \sum_{i=0}^{5} \Pr[G_i(\mathcal{A})] - \Pr[G_{i+1}(\mathcal{A})] \\
&\le 2\mathbf{Adv}_H^{\mathrm{prf}}(\mathcal{B}_1) + \mathbf{Adv}_{\widetilde{\mathsf{TE}}}^{\pm\widetilde{\mathsf{prp}}}(\mathcal{B}_2) + \mathbf{Adv}_{\mathsf{C}}^{\mathrm{hide}}(\mathcal{B}_3) + \frac{6cqB}{2^r} + \frac{2qB}{2^m} \ .
\end{aligned}$$

8.3 Proof of Lemma 15

We will use the technique in Section 8.1 . In particular, we will consider computationally unbounded adversaries, and thus without loss of generality, assume that the adversary is deterministic. The real system corresponds to game G_5, and the ideal system corresponds to game G_6. When the adversary finishes querying, in

the real world, we will grant it the subkeys J_i, J_i^* of the AXU hash G, 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 J_i, J_i^* and the following information:

- For each query $C_1\|C_2 \leftarrow \textsc{Enc}(i, T, P\|S)$, we store a corresponding entry $(i, T, P\|S, C_1\|C_2, U, V, X)$, where $U \leftarrow P \oplus G(J_i, S)$, $V \leftarrow C_1 \oplus G(J_i^*, C_2)$, and $X \leftarrow U[1:r] \oplus V[1:r]$.
- For each query $P\|S \leftarrow \textsc{Dec}(i, T, C_1\|C_2)$, we store a corresponding entry $(i, T, P\|S, C_1\|C_2, U, V, X)$, where $U \leftarrow P \oplus G(J_i, S)$, $V \leftarrow C_1 \oplus G(J_i^*, C_2)$, and $X \leftarrow U[1:r] \oplus V[1:r]$.

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

1. There are entries $(i, T, P\|S, C_1\|C_2, U, V, X)$ and $(i, T^*, P^*\|S^*, C_1^*\|C_2^*, U^*, V^*, X^*)$ such that $X = X^*$. This ensures that if G_4 sets bad then the transcript is bad.
2. There are $(i, T, P\|S, C_1\|C_2, U, V, X)$ and $(i, T, P^*\|S^*, C_1^*\|C_2^*, U^*, V^*, X^*)$ such that $U = U^*$. This forces $V = V^*$ in the real world, but it is unlikely to happen in the ideal world.
3. There are $(i, T, P\|S, C_1\|C_2, U, V, X)$ and $(i, T, P^*\|S^*, C_1^*\|C_2^*, U^*, V^*, X^*)$ such that $V = V^*$. This forces $U = U^*$ in the real world, but it is unlikely to happen in the ideal world.

Entries are stored in the order of the queries. If a transcript is not bad and is valid for the ideal system then we say that it is *good*.

Probability of bad transcripts. Let $\mathcal{T}_{\text{ideal}}$ be the random variable for the transcript in the ideal world. We now bound the probability that $\mathcal{T}_{\text{ideal}}$ is bad. Let Bad_j be the set of transcripts that violate the j-th constraint of badness. Then from the union bound,

$$\Pr[\mathcal{T}_{\text{ideal}} \text{ is bad}] \le \Pr[\mathcal{T}_{\text{ideal}} \in \text{Bad}_1 \cup \text{Bad}_2 \cup \text{Bad}_3] \le \sum_{j=1}^{3} \Pr[\mathcal{T}_{\text{ideal}} \in \text{Bad}_j] \ .$$

We first bound $\Pr[\mathcal{T}_{\text{ideal}} \in \text{Bad}_1]$. Consider two entries $(i, T, P\|S, C_1\|C_2, U, V, X)$ and $(i, T^*, P^*\|S^*, C_1^*\|C_2^*, U^*, V^*, X^*)$ in $\mathcal{T}_{\text{ideal}}$ in that order. Due to symmetry, without loss of generality, we only need to consider the case that second entry is created by a decryption query. In that case, S^* is uniformly random, independent of S, and thus the chance that $S^* = S$ is at most 2^{-m}. Suppose that $S^* \neq S$. Note that if $X = X^*$ then

$$\big(P \oplus P^* \oplus G(J_i^*, C_2) \oplus G(J_i^*, C_2^*)\big)[1:r] = \big(G(J_i, S) \oplus G(J_i, S^*)\big)[1:r] \ . \qquad (3)$$

Since J_i is independent of $(J_i^*, P, P^*, C_2, C_2^*, S, S^*)$, we can consider a fixed choice of $(J_i^*, P, P^*, C_2, C_2^*, S, S^*)$ but still treat J_i as uniformly random, and bound the *conditional* probability that Equation (3) happens. For any $\Delta \in \{0,1\}^{n-r}$, since G is c-AXU and $S \neq S^*$, the chance that $G(J_i, S) \oplus G(J_i, S^*)$ is $\big(P \oplus P^* \oplus G(J_i^*, C_2) \oplus G(J_i^*, C_2^*)\big)[1:r]\|\Delta$ is at most $c/2^n$. Summing this over

all 2^{n-r} choices of Δ, the chance that Equation (3) happens is at most $c/2^r$. Summing over at most all qB pairs of entries of the same user in $\mathcal{T}_{\text{ideal}}$,

$$\Pr[\mathcal{T}_{\text{ideal}} \in \text{Bad}_1] \leq \frac{qB}{2^m} + \frac{cqB}{2^r} \ .$$

Next, we bound $\Pr[\mathcal{T}_{\text{ideal}} \in \text{Bad}_2]$. Consider two entries $(i, T, P\|S, C_1\|C_2, U, V, X)$ and $(i, T, P^*\|S^*, C_1^*\|C_2^*, U^*, V^*, X^*)$ in $\mathcal{T}_{\text{ideal}}$. Due to the assumption on the way the adversary makes queries, we must have $(P, S) \neq (P^*, S^*)$. We consider the following cases.

Case 1: $S = S^*$, meaning $P \neq P^*$. If $U = U^*$ then

$$U \oplus U^* = (P \oplus G(J_i, S)) \oplus (P^* \oplus G(J_i, S^*)) = P \oplus P^*$$

which is a contradiction because $U \oplus U^*$ is the all-zero string, whereas $P \oplus P^*$ can't be the all-zero string.

Case 2: $S \neq S^*$. If $U = U^*$ then we must have

$$G(J_i, S) \oplus G(J_i, S^*) = (P \oplus P^*)[1 : n] \ .$$

Since G is c-AXU, and J_i is independent of (S, S^*, P, P^*), this happens with probability at most $c/2^n$.

Summing both cases over at most qB pairs of entries of the same user,

$$\Pr[\mathcal{T}_{\text{ideal}} \in \text{Bad}_2] \leq \frac{cqB}{2^n} \leq \frac{cqB}{2^r} \ .$$

Similarly, we can bound $\Pr[\mathcal{T}_{\text{ideal}} \in \text{Bad}_3] \leq cqB/2^r$. Hence

$$\Pr[\mathcal{T}_{\text{ideal}} \text{ is bad}] \leq \frac{qB}{2^m} + \frac{3cqB}{2^r} \ . \tag{4}$$

Transcript ratio. Fix a good transcript θ. For each system $\mathbf{S} \in \{\mathbf{S}_{\text{real}}, \mathbf{S}_{\text{ideal}}\}$, the event that $\mathbf{S}$ produces θ is the sequence of the following events:

- For each user i, its revealed hash keys J_i, J_i^* are as indicated by θ. Given prior events, the conditional probability that this happens is 2^{-2n} for both the real and ideal systems.
- For each entry $(i, T, P\|S, C_1\|C_2, U, V, X)$ in θ, if we query $(i, T, P\|S)$ then we obtain $C_1\|C_2$. Given prior events, the condition probability that this happens in the ideal system is

  $$\frac{1}{2^{|P|+m} - Q} \ .$$

  where Q is the number of prior entries of the same (i, T) and $|P|$ in θ. In contrast, the conditional probability that this happens in the real system is

  $$\frac{1}{(2^{|P|} - Q)2^m} \geq \frac{1}{2^{|P|+m} - Q} \ .$$

Hence $\mathsf{p}_{\mathbf{S}_{\text{real}}}(\theta) \geq \mathsf{p}_{\mathbf{S}_{\text{ideal}}}(\theta)$.

Wrapping up. Applying Lemma 13 and Lemma 14 with $\epsilon = 0$ and $\Pr[\mathcal{T}_{\text{ideal}} \text{ is bad}]$ as indicated from Equation (4), we obtain the claimed bound.

Acknowledgement. We thank the Asiacrypt 2024 reviewers for their valuable comments. Viet Tung Hoang was supported in part by NSF grant CNS-2046540 (CAREER).

References

1. A. Albertini, T. Duong, S. Gueron, S. Kölbl, A. Luykx, and S. Schmieg. How to abuse and fix authenticated encryption without key commitment. In K. R. B. Butler and K. Thomas, editors, *USENIX Security 2022*, pages 3291–3308. USENIX Association, Aug. 2022. 2
2. E. Andreeva, A. Bogdanov, A. Luykx, B. Mennink, N. Mouha, and K. Yasuda. How to securely release unverified plaintext in authenticated encryption. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 105–125. Springer, Heidelberg, Dec. 2014. 2
3. E. Andreeva, A. Bogdanov, A. Luykx, B. Mennink, E. Tischhauser, and K. Yasuda. Parallelizable and authenticated online ciphers. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 424–443. Springer, Heidelberg, Dec. 2013. 2
4. M. Bellare and V. T. Hoang. Efficient schemes for committing authenticated encryption. In O. Dunkelman and S. Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 845–875. Springer, Heidelberg, May / June 2022. 2, 3, 10, 11, 12, 14
5. M. Bellare and V. T. Hoang. Succinctly-committing authenticated encryption. In *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA*, 2024. 2, 3, 5, 12, 18, 19, 21
6. M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. Format-preserving encryption. In M. J. Jacobson Jr., V. Rijmen, and R. Safavi-Naini, editors, *SAC 2009*, volume 5867 of *LNCS*, pages 295–312. Springer, Heidelberg, Aug. 2009. 12
7. M. Bellare and P. Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In T. Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 317–330. Springer, Heidelberg, Dec. 2000. 2, 9
8. 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. 27
9. P. Campbell. GLEVIAN and VIGORNIAN: Robust beyond-birthday aead modes. Cryptology ePrint Archive, Paper 2023/1379, 2023. `https://eprint.iacr.org/2023/1379`. 2, 6
10. D. Chakraborty and P. Sarkar. HCH: A new tweakable enciphering scheme using the hash-encrypt-hash approach. In R. Barua and T. Lange, editors, *INDOCRYPT 2006*, volume 4329 of *LNCS*, pages 287–302. Springer, Heidelberg, Dec. 2006. 23
11. J. Chan and P. Rogaway. On committing authenticated-encryption. In V. Atluri, R. Di Pietro, C. D. Jensen, and W. Meng, editors, *ESORICS 2022, Part II*, volume 13555 of *LNCS*, pages 275–294. Springer, Heidelberg, Sept. 2022. 2
12. 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. 27, 28

13. Y. L. Chen, A. Flórez-Gutiérrez, A. Inoue, R. Ito, T. Iwata, K. Minematsu, N. Mouha, Y. Naito, F. Sibleyras, and Y. Todo. Key committing security of AEZ. The Third NIST Workshop on Block Cipher Modes of Operation, 2023. 3
14. P. Crowley, N. Huckleberry, and E. Biggers. Length-preserving encryption with HCTR2. Cryptology ePrint Archive, Paper 2021/1441, 2021. `https://eprint.iacr.org/2021/1441`. 2, 23, 27
15. 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. 28
16. 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. 2
17. S. Gueron, A. Langley, and Y. Lindell. AES-GCM-SIV: specification and analysis. *IACR Cryptol. ePrint Arch.*, 2017. 23, 27
18. V. T. Hoang, T. Krovetz, and P. Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 15–44. Springer, Heidelberg, Apr. 2015. 2, 3, 9, 23
19. 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. 28
20. J. Len, P. Grubbs, and T. Ristenpart. Partitioning oracle attacks. In M. Bailey and R. Greenstadt, editors, *USENIX Security 2021*, pages 195–212. USENIX Association, Aug. 2021. 2
21. M. Luby and C. Rackoff. How to construct pseudo-random permutations from pseudo-random functions (abstract). In H. C. Williams, editor, *CRYPTO'85*, volume 218 of *LNCS*, page 447. Springer, Heidelberg, Aug. 1986. 17
22. Z. Luo, Y. Jia, Y. Shen, and A. Kate. Proxying is enough: Security of proxying in TLS oracles and AEAD context unforgeability. Cryptology ePrint Archive, Paper 2024/733, 2024. `https://eprint.iacr.org/2024/733`. 2
23. 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. 8
24. M. Naor and O. Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. *Journal of Cryptology*, 12(1):29–66, Jan. 1999. 4, 22
25. 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. 27, 28
26. 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. 2, 5, 18