A preliminary version of this paper appears in CRYPTO 2024. This is the full version.

Succinctly-Committing Authenticated Encryption

Mihir Bellare[1] Viet Tung Hoang[2]

June 2024

Abstract

Recent attacks and applications have led to the need for symmetric encryption schemes that, in addition to providing the usual authenticity and privacy, are also committing. In response, many committing authenticated encryption schemes have been proposed. However, all known schemes, in order to provide s bits of committing security, suffer an expansion—this is the length of the ciphertext minus the length of the plaintext—of $2s$ bits. This incurs a cost in bandwidth or storage. (We typically want $s = 128$, leading to 256-bit expansion.) However, it has been considered unavoidable due to birthday attacks. We show how to bypass this limitation. We give authenticated encryption (AE) schemes that provide s bits of committing security, yet suffer expansion only around s as long as messages are long enough, namely more than s bits. We call such schemes succinct. We do this via a generic, ciphertext-shortening transform called SC: given an AE scheme with $2s$-bit expansion, SC returns an AE scheme with s-bit expansion while preserving committing security. SC is very efficient; an AES-based instantiation has overhead just two AES calls. As a tool, SC uses a collision-resistant invertible PRF called HtM, that we design, and whose analysis is technically difficult. To add the committing security that SC assumes to a base scheme, we also give a transform CTY that improves Chan and Rogaway's CTX. Our results hold in a general framework for authenticated encryption, called AE3, that includes both AE1 (also called AEAD) and AE2 (also called nonce-hiding AE) as special cases, so that we in particular obtain succinctly-committing AE schemes for both these settings.

[1]Department of Computer Science and Engineering, University of California San Diego. Supported in part by NSF grant CNS-2154272 and KACST.

[2]Department of Computer Science, Florida State University. Supported in part by NSF grant CNS-2046540 (CAREER).

Contents

1 Introduction

Authenticated Encryption refers to symmetric encryption that provides both data privacy and authenticity. It is widely used in practice. Recent attacks and applications, however, indicate that AE schemes should also be committing. This paper asks how succinctly, meaning with how little a growth in ciphertext size, one can provide committing security, and gives an essentially optimal solution. We start with some background.

Authenticated encryption. Modern symmetric encryption is nonce-based. It comes in a few forms. We'll start by focusing on the form called AE1 or AEAD [39]. (AE1 is not without its defects [7], but it is the currently most popular form of symmetric encryption. We'll get to other forms later.)

In an AE1 scheme SE, the encryption algorithm $\mathsf{SE.Enc}$ takes key K, nonce N, associated data A and message M to deterministically return a ciphertext $C \leftarrow \mathsf{SE.Enc}(K, N, A, M)$. Decryption recovers via $M \leftarrow \mathsf{SE.Dec}(K, N, A, C)$. AE1 security asks for privacy of the message and authenticity of both the message and the associated data, under the condition that encryption never repeats (reuses) a nonce. A central AE1 scheme is GCM [35]. It is a NIST standard [24]. AE1 (and GCM in particular) is widely used, in particular for securing communicated data in TLS [42] and for securing stored data in Amazon Web Services (AWS).

Committing security. Different names and formalizations of this, given over the years in the literature [26, 1, 27, 25, 2, 5, 16, 36], have converged to one optimally strong and simple one from [5, 16], namely that the function $\mathsf{SE.Enc}$ is collision resistant. That is, it is hard to find distinct tuples $(K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2)$ such that $\mathsf{SE.Enc}(K_1, N_1, A_1, M_1) = \mathsf{SE.Enc}(K_2, N_2, A_2, M_2)$. We will refer to this simply as committing, abbreviated CMT.

CMT-security is motivated by recent attacks and applications. One such is password-based encryption (PBE), which Len, Grubbs, and Ristenpart [33] show how to break, in some settings, if the underlying AE1 scheme is not CMT-secure. The presence of CMT-security in the scheme is then shown by Bellare and Shea [11] to restore security for PBE. Albertini *et al.* [2] describe security failures in key rotation, envelope encryption and subscribe-with-Google that arise from lack of CMT-security of the AE1 scheme. Committing security is needed for unambiguous decryption in anonymous encryption [1, 15]. Grubbs, Lu, and Ristenpart [27] show that CMT-security is needed for an AE1 scheme to provide message franking, a capability in messaging systems that allows a receiver to report the receipt of abusive content. Further examples are in [25, 32].

Towards obtaining CMT-secure schemes, it is natural to first ask if existing AE1 schemes happen to already be committing. The answer is broadly no. Attacks from [33, 2, 27, 36, 16] show that GCM, $\mathsf{ChaCha20/Poly1305}$, OCB [40], CCM [43], EAX [10] and SIV [41] are all CMT-insecure. Chen *et al.* [17] give attacks on Encode-then-Encipher (EtE) schemes [8] if the enciphering is done via AEZ [30], Adiantum [18], or HCTR2 [19].

This has led to the design of new AE1 schemes that are proven CMT-secure. The most attractive solutions are generic transforms. One such is CTX [16], which, with the aid of a hash function, adds CMT-security to any tag-based AE1 scheme. Another is $\mathsf{HtE} \circ \mathsf{UtC}$ [5] that adds CMT-security to any AE1 scheme. (As the notation indicates, this composes two individual transforms.) There are also dedicated schemes: $\mathsf{CAU\text{-}C4}$ [5] is a simple modification of GCM that is CMT-secure, and $\mathsf{SpongeWrap}$ [13] has been shown to be CMT-secure [22].

CTY. Before getting to our core contributions we pause to discuss an auxiliary one, namely a transform we call CTY. It is an improvement of CTX that offers the same security but with strictly greater efficiency. Recall that CTX takes a (tag-based) AE1 scheme SE_1 and hash function H to

return a CMT-secure AE1 scheme $\mathsf{SE}_2 \leftarrow \mathsf{CTX}[\mathsf{SE}_1, \mathsf{H}]$ built as follows: $\mathsf{SE}_2.\mathsf{Enc}(K, N, A, M)$ lets $C' \| R \leftarrow \mathsf{SE}_1.\mathsf{Enc}(K, N, A, M)$ —here R is the tag— and returns $C' \| T$ where $T = \mathsf{H}(K, N, A, R)$. For CTY, the change is simply that A is dropped (more precisely, replaced by the empty string ε) as input to $\mathsf{SE}_1.\mathsf{Enc}$ in the first step. The speed increase is perceptible for long A. Theorem 3.3 implies that CTY, like CTX, provides s bits of CMT security if the hash function outputs $2s$ bits, and Theorem 3.4 implies it also retains the AE1-security of SE_1. We will use CTY as a tool but it is also of independent interest.

EXPANSION. An AE1 scheme SE has expansion e if $|\mathsf{SE}.\mathsf{Enc}(K, N, A, M)| = |M| + e$ for all K, N, A, M. (That is, ciphertexts are e bits longer than plaintexts.) Now say we want SE to provide s bits of CMT-security, meaning it takes 2^s time to violate CMT. It turns out that *all existing CMT-secure schemes* incur expansion $e \geq 2s$ to provide s bits of CMT security.

In this paper we ask whether $2s$-bit expansion is necessary for s bits of CMT-security, and whether it can instead be provided with expansion (about) s.

This question is of both theoretical and practical interest. To explain the latter, first note that CMT-security, unlike AE1-security, is subject (like collision-resistance for hash functions) to offline attack, so while 64-bits of AE1 security is considered adequate (and is what GCM provides), one would like 128 bits of CMT security. With known schemes or transforms, this incurs 256 bits of expansion. This is 128 bits more than current (non-committing) AE1 schemes like GCM. So CMT-security is coming at a price in bandwidth. For IoT devices, this translates to increased power consumption that they can ill afford. It also means increased storage cost for cloud storage systems. Moreover, increasing the ciphertext length can create backward compatibility issues in legacy systems.

The quest to minimize expansion is not new. Indeed, robust AE and AEZ [30] were motivated by the need to provide best-possible AE security with a user-defined expansion. The area of lightweight cryptography also aims (amongst other things) to minimize expansion. We are following in these steps by addressing the same concerns for CMT security.

Due to these practical considerations, we not only want to reduce expansion but want to do so with minimal performance penalty.

THE BIRTHDAY ATTACK. Recall that current CMT-secure AE1 schemes incur expansion $e \geq 2s$ to provide s bits of CMT security and our question is whether there exist AE1 schemes that reduce this expansion to (around) s. Folklore says that the answer is "no." Indeed, the $2s$-bit expansion is seen as necessary, due to the following birthday attack. Let SE be any AE1 scheme and let e be its expansion. Pick distinct $(K_1, N_1, A_1), \ldots, (K_q, N_q, A_q)$ and encrypt the empty message ε under them to obtain ciphertexts $C_i \leftarrow \mathsf{SE}.\mathsf{Enc}(K_i, N_i, A_i, \varepsilon)$ for $i = 1, \ldots, q$. Since the message has length 0, these ciphertexts all have length $0 + e = e$. A collision, meaning distinct a, b with $C_a = C_b$, is thus expected when $q = 2^{e/2}$, and constitutes a violation of CMT-security. Requiring $q \geq 2^s$ (for s-bit security) now forces $e \geq 2s$.

EVENTUAL EXPANSION. We suggest a way out. This is to allow variable expansion. The expansion now is not a constant but rather a function $\mathsf{SE.ce} : \mathbb{N} \to \mathbb{N}$ of the message length, meaning $|\mathsf{SE}.\mathsf{Enc}(K, N, A, M)| = |M| + \mathsf{SE.ce}(|M|)$. In particular let us say a scheme has *eventual expansion* e if $\mathsf{SE.ce}(m) = e$ for all $m \geq \mathsf{c}$ where c is a small value called the *cutoff*. We claim (with justification below) that, while attacks mandate expansion $2s$ to provide s-bits of CMT security, there is nothing precluding schemes with *eventual* expansion much smaller, even as small as s, as long as $\mathsf{c} \geq s$.

Let us refer to a scheme providing (about) s-bits of CMT security with eventual expansion s as *succinctly committing*. A succinct scheme (if it exists!) would "morally" be providing s bits of CMT-security with the optimal s bits of expansion. Concretely, for $s = 128$, it would reduce

expansion from 256 to (the optimal) 128 for all long enough (more than 128-bit if $\mathsf{c} = 128$) messages. This means it will (for messages above the cutoff length) provide 128-bit CMT-security with the same expansion as the non-CMT-secure GCM scheme, so that no extra bandwidth costs are incurred to add CMT-security in this case.

The challenge will be to actually give succinct schemes, but first let us revisit the attack picture to justify the above claim. For a scheme with variable expansion $\mathsf{SE.ce}$, the above birthday attack says that $\mathsf{SE.ce}(0) \geq 2s$, which imposes no restriction on *eventual* expansion being as low as s. The attack can be generalized to use an m-bit message, but in this case takes $q = 2^{(m+\mathsf{SE.ce}(m))/2}$ time, leading to the constraint $m + \mathsf{SE.ce}(m) \geq 2s$ or $\mathsf{SE.ce}(m) \geq 2s - m$. Simple attacks show that $\mathsf{SE.ce}(m) \geq s$ is always needed, so that overall the constraint that emerges is that $\mathsf{SE.ce}(m) \geq \max(2s - m, s)$ for all $m \geq 0$. This means that eventual expansion as low as s, with cutoff also s, is not excluded by these attacks. This is the opening we want to exploit.

Our solution. Above we have opened the door to succinct CMT-secure AE1 schemes by showing they would not contradict known attacks. But this is a far cry from showing such schemes actually exist. We would like not only to show that they do exist but also to give schemes that, in light of the above practical considerations, have minimal performance overhead compared to existing schemes with $2s$-bit expansion. To achieve both of these goals, we have a two-step solution.

▷ **Step 1: SC.** Suppose that we have a committing (tag-based) AE1 scheme SE_1 of s-bit committing security but with expansion of $\lambda \geq 2s$ bits. We aim to *shorten* the ciphertext of SE_1 with very little security degradation. For this, we give a transform SC that, with the aid of an auxiliary primitive F, constructs an AE1 scheme $\mathsf{SE}_2 \leftarrow \mathsf{SC}[\mathsf{SE}_1, \mathsf{F}]$ as shown in Figure 9. The auxiliary primitive F is a collision-resistant invertible PRF (CR IPF) $\mathsf{F} : \{0,1\}^\lambda \times \{0,1\}^{\leq s} \to \{0,1\}^{2s}$ that takes a λ-bit key T and an input P of length at most s to return a $2s$-bit output $R \leftarrow \mathsf{F}(T, P)$. Scheme SE_2 has eventual expansion only s, with cutoff around s. With regard to security, we give a construction HtM for F such that the AE1 scheme SE_2 provides (about) s bits of CMT-security. Moreover, SC is very efficient. Indeed, if we use HtM for F then the overhead of SC is just two AES calls.

Unfortunately, SC does not *generically* preserve AE1 security. Still, we can show that if SE_1 is obtained by applying our above-mentioned CTY transform to a standard tag-based AE1 scheme SE_0, then SC *does* preserve AE1 security. Furthermore, CTY is currently the fastest way to build a CMT-secure AE1 scheme. In other words, by composing SC and CTY, we have a fast transform that turns a standard tag-based AE1 scheme SE_0 (such as GCM) into a succinctly committing AE1 scheme SE_2.

▷ **Step 2: HtM.** For SC to work as claimed above, we need a CR IPF $\mathsf{F} : \{0,1\}^\lambda \times \{0,1\}^{\leq s} \to \{0,1\}^{2s}$ that provides (nearly) s bits of CR-security and at least $s/2$-bits of PRF security. Furthermore we want F to be as efficient as possible. We give a construction called HtM that we prove achieves this. The construction itself is quite simple, but the analysis to show the tight bounds we need is difficult.

HtM can be seen as a 2-round Feistel in which the right-half of the Feistel input is the (padded) F input and the left half is the all-zero string. Alternatively, HtM can also be viewed as following the SIV paradigm [41]. To minimize the cost of HtM, we build the round functions from the Davies-Meyer function of a blockcipher E; this blockcipher can be instantiated from AES if $s \leq 128$. See Figure 7 for a full description.

It's trivial to show that HtM provides $s/2$ bits of collision resistance using the well-known collision resistance of Davies-Meyer. Furthermore, at first glance, given that HtM uses a CR hash of s-bit output, one doesn't expect anything beyond $s/2$ bits of collision-resistance security. Surprisingly, by exploiting a circularity in the SIV paradigm, we can prove that HtM provides

(nearly) s bits of collision resistance in the ideal-cipher model.

Our CR proof for HtM however is technically involved. The main obstacle is that Feistel decryption is *non-atomic*. For example, the adversary may just specify the left half of a ciphertext (that can match several prior ciphertexts) and run a Feistel round on it, and then *adaptively* pick the right half of the ciphertext to continue the decryption. Even worse, since we build the Feistel rounds on top of a blockcipher E, the complexity blows up, because one has to consider both forward and backward queries to E. (Had we instead built the Feistel rounds from a random oracle then the analyses would be a lot more tractable.) To deal with this problem, we (1) use concentration bounds via (non-standard) balls-into-bins analyses to show that the adaptivity can at best amplify the adversary's advantage by a factor of s, and (2) use a detailed case analysis on the types of the four ideal-cipher queries that correspond to the adversary's output.

Alternative IPFs. Invertible PRF (IPFs) arises naturally in the context of deterministic AE, a solution for the key-wrapping problem [41]. An IPF is a special case of a pseudorandom injection (one of the characterizations of deterministic AE) in which the header is just the empty string. The new element for us is that we also want collision-resistance (CR). HtM is not the only possible approach. A natural approach to use the encode-then-encipher paradigm (EtE) [8], building a CR IPF $\mathsf{F}\colon \{0,1\}^\lambda \times \{0,1\}^{\leq n-s} \to \{0,1\}^n$ via $\mathsf{F}(T,P) = E(T, P\|1\|0^{n-|P|-1})$ for a block cipher $E\colon \{0,1\}^\lambda \times \{0,1\}^n \to \{0,1\}^n$. In Proposition 5.3 we verify its IPF security, and in Proposition 5.4 we show that it provides s bits of CR security if E is an ideal cipher and $n \geq 2s$. However, one can't directly instantiate E from AES because we want $s \geq 128$. Instantiating E using a Feistel network via indifferentiability theory [34] leads to several obstacles. Namely, on the one hand, if we want the Feistel network itself to be indifferentiable from an ideal cipher, then the state of the art [21] demands 8 Feistel rounds, and the concrete bound is poor, so we do not in the end get s bits of CR from F. On the other hand, if we want the EtE construction to be indifferentiable from an ideal injective function, then we only need 3 Feistel rounds [3], but we only get $s/2$ bits of CR from F. Compared to these approaches, HtM does not aim for indifferentiability but instead directly achieves CR-security and PRF-security. It does this very efficiently (just 2 Feistel rounds, and each round from a single AES call) and moreover the concrete bounds proven are very good.

Generalization and extensions. Above we have focused on the form of symmetric encryption called AE1 or AEAD [39] but in fact our results are more general and apply to a broader class of schemes, as we now explain.

AE1 claims to provide security for any (non-repeating) choice of nonce, but in fact fails to actually do so [7]. This may sound surprising but in hindsight is obvious. Since AE1 decryption needs the nonce as input, it must be sent in the clear. But there are valid and useful choices of nonces, like hashes of the message, that if used with low-entropy messages, allow message recovery given the ciphertext and nonce. A remedy is AE2 [7], which truly provides security for all nonce choices. (We forbear to refer to it by the sometimes-used term "nonce-hiding" because the latter misleads one into seeing hiding the nonce as the primary goal. In fact, hiding the nonce is primarily a means to truly hide the message, and secondarily important because nonces too can be sensitive.) Now all the reasons to desire committing-security for AE1 hold also for AE2 so it is natural to ask if our results extend to AE2.

They do, but in fact more is true; our results hold, and are established in, a broader framework from [11] that includes both AE1 and AE2 as special cases. We call it AE3. We develop the AE3 framework in Section 3. Encryption, as before, takes K, N, A, M to return a ciphertext, but decryption takes K, N, D, C where $D = \mathsf{SE.df}(N)$ is determined by a scheme-associated function SE.df called the decryption-nonce function. Defining $\mathsf{SE.df}(N) = N$ recovers AE1 and defining $\mathsf{SE.df}(N) = \varepsilon$ recovers AE2. But consider a nonce N consisting of a sensitive device-id, plus a

random number that can be sent in the clear. Neither AE1 nor AE2 capture this well, but AE3 does, by letting $\mathsf{SE.df}(N)$ return the non-sensitive part of the nonce N.

In Section 3 we define AE3 (syntax and AE security) following [11] but also, for the first time, define its committing security (CMT). We then give our constructions and results in this general framework. Specifically, both the CTY transform and the SC transform are given for general (tag-based) AE3 schemes. Results for AE1 and AE2 are recovered as special cases via the particular choices of SE.df given above.

Further related work and approaches. CAU-C4 [5] provides 64 bits of CMT-security with 128-bit expansion and is attractive due to being only a slight modification of GCM. However, as argued above, due to offline attacks, one might prefer 128 bits of CMT security. For sponge-based schemes like Ascon [23] or SpongeWrap [13], if we want 128-bit CMT security, we still need 256-bit expansion. Moreover, those schemes are much slower than AES-based ones in platforms where AES-NI is available. (Even if the underlying permutation is designed based on the AES round function, like Simpira [29], the sequential nature of the sponge construction will disrupt the AES-NI pipelining.)

The Encode-then-encipher (EtE) construction of [8] is shown in [27] to provide a form of commitment called receiver-binding (r-BIND) security. This is not the same as CMT but their result does show s bits of r-BIND security with s bits of expansion if the underlying cipher is modeled as ideal. However, it is unclear how to build this ideal enciphering from common primitives such as a blockcipher. In fact, if we use off-the-shelf enciphering schemes like AEZ [30] or Adiantum [18] then the committing security of EtE breaks down [17]. Moreover, EtE constructions are expensive and need at least two passes over the data.

In concurrent work, Naito, Sasaki, and Sugawara [37] also explore how to achieve s-bit CMT security with s-bit ciphertext expansion. Their work assume that the message format offers some redundancy that is known to the encryption scheme. For example, in the HTTP protocol, a message starts with "HTTP/1.1", providing an eight-byte redundancy. They give a transform KIVR to add committing security to a base AE scheme SE without increasing the ciphertext expansion. If SE is GCM then KIVR requires $2s$ bits of redundancy for s-bit committing security. The use of KIVR is rather limited because it requires that the message format provides enough redundancy. In the example of the HTTP protocol, messages offer only 64-bit redundancy, meaning that KIVR provides merely 32-bit committing security. Moreover, in practice it's unrealistic to assume that message redundancy is known to the encryption algorithm.

2 Preliminaries

Notation and Terminology. Let ε denote the empty string. For a string $x \in \{0,1\}^*$ we let $|x|$ denote its length, and let $x[i:j]$ denote the bits in positions i through j (inclusive), for $1 \le i \le j \le |x|$. For an integer $n \ge 1$, we let $\{0,1\}^{\le n} = \{\, x \in \{0,1\}^* : |x| \le n \,\}$ be the set of strings of length at most n. By $\mathsf{Func}(\mathsf{Dom},\mathsf{Rng})$ we denote the set of all functions $f : \mathsf{Dom} \to \mathsf{Rng}$. By $\mathsf{Perm}(\mathsf{Dom})$ we denote the set of all permutations $\pi : \mathsf{Dom} \to \mathsf{Dom}$. We write $\mathsf{Perm}(n)$ for $\mathsf{Perm}(\{0,1\}^n)$. 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. If A is a randomized algorithm, we let $y \leftarrow\!\!\$\, A(x_1,\ldots)$ denote running A on inputs $x_1,\ldots$ and assigning the output to y. If A is deterministic, we may use $\leftarrow$ in place of $\leftarrow\!\!\$$. If S is a finite set, then $|S|$ denotes it size. We say that a set S is *full* if it is either $\{0,1\}^*$ or $\{0,1\}^{\le n}$ for some n. (This will be a requirement for message, associated-data, and nonce spaces.)

Game $\mathbf{G}^{\mathsf{prf}}_{\mathsf{F}}(\mathcal{A})$	Game $\mathbf{G}^{\pm\mathsf{prp}}_{E}(\mathcal{A})$
$b \leftarrow\!\!\$\ \{0,1\};\ \ b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Eval}}$	$b \leftarrow\!\!\$\ \{0,1\};\ \ b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Dec}}$
Return $(b' = b)$	Return $(b' = b)$
$\underline{\textsc{New}()}$	$\underline{\textsc{New}()}$
$v \leftarrow v + 1;\ \ K_v \leftarrow\!\!\$\ \mathcal{K}$	$v \leftarrow v + 1;\ \ K_v \leftarrow\!\!\$\ \{0,1\}^k$
$f_v \leftarrow\!\!\$\ \mathsf{Func}(\mathsf{Dom}, \mathsf{Rng})$	$\Pi_v \leftarrow\!\!\$\ \mathsf{Perm}(n)$
$\underline{\textsc{Eval}(i, M)}$	$\underline{\textsc{Enc}(i, M)}$
$C_1 \leftarrow \mathsf{F}(K_i, M)$	$C_1 \leftarrow E(K_i, M);\ \ C_0 \leftarrow \Pi_i(M)$
$C_0 \leftarrow f_i(M)$	Return C_b
Return C_b	$\underline{\textsc{Dec}(i, C)}$
	$M_1 \leftarrow E^{-1}(K_i, C);\ \ M_0 \leftarrow \Pi_i^{-1}(C)$
	Return M_b

Figure 1: Left: Game defining (multi-user) PRF security of F. Right: Game defining (multi-user) strong PRP security of E.

For two adversaries that are given the same set of oracles, we say that they have the same *query statistics* if they have the same number of queries, the same number of queries per user, and the same total length of queries, for each oracle that they are given.

GAMES. We use the code-based game-playing framework of [9] with some modifications. A game $\mathrm{G}(\mathcal{A})$ runs adversary $\mathcal{A}$ with oracles that the game specifies as procedures, and then returns an output. By $\Pr[\mathrm{G}(\mathcal{A}) \Rightarrow y]$ we denote the probability that the execution of $\mathrm{G}(\mathcal{A})$ results in the game output being y, and write just $\Pr[\mathrm{G}(\mathcal{A})]$ for $\Pr[\mathrm{G}(\mathcal{A}) \Rightarrow \mathsf{true}]$. In game pseudocode, integer variables, set variables, boolean variables and string variables are assumed initialized, respectively, to 0, the empty set $\emptyset$, the boolean false and $\perp$. Adversaries in games are always assumed to be domain-respecting, meaning if a query they provide is expected to fall in some scheme-associated set, then it does.

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 $(X_1, X_2) \leftarrow\!\!\$\ \mathcal{A}$.

PRFS AND PRPS. 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.

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. 1.

3 The AE3 Symmetric Encryption Framework

Our results are strengthened by using a general, unified definitional framework from [11]. We call it AE3. It includes AE1 (also called AEAD) and AE2 (also called nonce-hiding AE) as special cases. It starts with a general syntax that we give and then explain.

AE3 SYNTAX. A symmetric encryption scheme SE in the AE3 framework specifies a deterministic encryption algorithm $\mathsf{SE.Enc} : \{0,1\}^{\mathsf{SE.kl}} \times \{0,1\}^{\mathsf{SE.nl}} \times \mathsf{SE.AS} \times \mathsf{SE.MS} \to \{0,1\}^*$, a deterministic decryption algorithm $\mathsf{SE.Dec} : \{0,1\}^{\mathsf{SE.kl}} \times \{0,1\}^{\mathsf{SE.dl}} \times \mathsf{SE.AS} \times \{0,1\}^* \to (\mathsf{SE.MS} \times \{0,1\}^{\mathsf{SE.nl}}) \cup \{\perp\}$, and a decryption-nonce derivation function $\mathsf{SE.df} : \{0,1\}^{\mathsf{SE.nl}} \to \{0,1\}^{\mathsf{SE.dl}}$. Here $\mathsf{kl}, \mathsf{nl}, \mathsf{dl}$ are the key, nonce and decryption-nonce lengths, respectively. The associated-data space $\mathsf{SE.AS}$ and message space $\mathsf{SE.MS}$ are required to be full sets as defined above. We require that there is a *ciphertext-length function* $\mathsf{SE.cl} : \mathbb{N} \to \mathbb{N}$ such that $|\mathsf{SE.Enc}(K, N, A, M)| = \mathsf{SE.cl}(|M|)$ for all $(K, N, A, M) \in \{0,1\}^{\mathsf{SE.kl}} \times \{0,1\}^{\mathsf{SE.nl}} \times \mathsf{SE.AS} \times \mathsf{SE.MS}$. The *ciphertext-expansion function* $\mathsf{SE.ce} : \mathbb{N} \to \mathbb{N}$ is then defined by $\mathsf{SE.ce}(m) = \mathsf{SE.cl}(m) - m$ for all $m \in \mathbb{N}$. The correctness requirement is that if $C \leftarrow \mathsf{SE.Enc}(K, N, A, M)$ then $\mathsf{SE.Dec}(K, \mathsf{SE.df}(N), A, C)$ returns (M, N) for all $(K, N, A, M) \in \{0,1\}^{\mathsf{SE.kl}} \times \{0,1\}^{\mathsf{SE.nl}} \times \mathsf{SE.AS} \times \mathsf{SE.MS}$.

Now let us explain. The sender is as usual expected to communicate the ciphertext $C \leftarrow \mathsf{SE.Enc}(K, N, A, M)$ to the receiver. However, what information the receiver gets about the nonce varies across definitions in the literature. In AE1 (also called AEAD [39]), the nonce is sent in the clear and is an explicit input to $\mathsf{SE.Dec}$, captured in AE3 by setting $\mathsf{SE.df}$ to the identity function, namely $\mathsf{SE.df}(N) = N$ for all $N \in \{0,1\}^{\mathsf{SE.nl}}$, and correspondingly setting $\mathsf{SE.dl} = \mathsf{SE.nl}$. The AE1 definition however fails to provide security for some choices of nonces, for example when they are hashes of the messages [7]. This leads [7] to introduce AE2 (also called nonce-hiding AE), where the decryptor gets no information about the nonce, captured in AE3 by setting $\mathsf{SE.df}(N) = \varepsilon$ for all $N \in \{0,1\}^{\mathsf{SE.nl}}$ and correspondingly $\mathsf{SE.dl} = 0$.

The AE3 syntax of [11] not only unifies AE1 and AE2, but generalizes them to capture realistic usage in between the two extremes. In applications, the nonce can consist of a sensitive part, meaning one whose privacy we want to protect, like a device-id, and a non-sensitive part, meaning one whose privacy is not a concern, like a random number. The sensitive part would be encrypted, and the receiver is expected to recover it through the ciphertext. The non-sensitive part is sent in the clear. This setting is captured in AE3 by letting $\mathsf{SE.df}(N)$ return the non-sensitive part of the nonce.

The decryption-nonce derivation function, unlike the encryption and decryption algorithms, is not necessarily implemented with the scheme. Its purpose is instead conceptual. It does show up in the definitions of security.

Note we have required decryption to recover not only the message but also the nonce. In AE1 this is redundant but we maintain it for consistency. However it will be useful to write $\mathsf{SE.Dec}^*(K, D, A, C)$ for just the message, meaning the first component of $\mathsf{SE.Dec}(K, D, A, C)$.

We define the *decryption-nonce density* $\mathsf{SE.dnd} \in \mathbb{N}$ of SE to be the smallest integer $d \geq 0$ such that: $|\{\ N \in \{0,1\}^{\mathsf{SE.nl}} \ :\ \mathsf{SE.df}(N) = D\ \}| \leq 2^d$ for all $D \in \{0,1\}^{\mathsf{SE.dl}}$. For AE1 schemes, $\mathsf{SE.dnd} = 0$. For AE2 schemes, $\mathsf{SE.dnd} = \mathsf{SE.nl}$ is the nonce length.

TIDINESS. We extend the definition of AE1 tidiness [38] to AE3 as follows. We say that SE is *tidy* if for all $(K, D, A, C) \in \{0,1\}^{\mathsf{SE.kl}} \times \{0,1\}^{\mathsf{SE.dl}} \times \mathsf{SE.AS} \times \{0,1\}^*$ the following is true: If $(M, N) \leftarrow \mathsf{SE.Dec}(K, D, A, C)$ and $(M, N) \neq \perp$, then it must be that $\mathsf{SE.df}(N) = D$ and $\mathsf{SE.Enc}(K, N, A, M) = C$. The schemes we consider will be tidy.

Suppose $(K, N, A) \in \{0,1\}^{\mathsf{SE.kl}} \times \{0,1\}^{\mathsf{SE.nl}} \times \mathsf{SE.AS}$ and we let $I_{K,N,A} = \{\ \mathsf{SE.Enc}(K, N, A, M)\ :\ M \in \mathsf{SE.MS}\ \}$. Combining correctness and tidiness implies that the functions $\mathsf{SE.Enc}(K, N, A, \cdot) : \mathsf{SE.MS} \to I_{K,N,A}$ and $\mathsf{SE.Dec}^*(K, \mathsf{SE.df}(N), A, \cdot) : I_{K,N,A} \to \mathsf{SE.MS}$ are the inverse of each other.

TAG-BASED SCHEMES. We say that symmetric encryption scheme SE is *tag-based* if its ciphertext consists of a $\mathsf{SE.tl}$-bit tag and a ciphertext core C^*, and decryption verifies the tag. Proceeding formally, SE is tag-based if it specifies an additional deterministic tagging algorithm $\mathsf{SE.Tag} : \{0,1\}^{\mathsf{SE.kl}}$

Game $\mathbf{G}^{\mathrm{real}}_{\mathsf{SE}}(\mathcal{A})$	Game $\mathbf{G}^{\mathrm{rand}}_{\mathsf{SE}}(\mathcal{A})$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Vf}}$	$b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Vf}}$
Return b'	Return b'
$\textsc{New}()$	$\textsc{New}()$
$v \leftarrow v+1$; $K_v \leftarrow\!\!\$\ \mathcal{K}$	$v \leftarrow v+1$
$\textsc{Enc}(i,N,A,M)$	$\textsc{Enc}(i,N,A,M)$
$C \leftarrow \mathsf{SE.Enc}(K_i,N,A,M)$	$C \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.cl}(\lvert M\rvert)}$
$S \leftarrow S \cup \{(i,\mathsf{SE.df}(N),A,C)\}$	$S \leftarrow S \cup \{(i,\mathsf{SE.df}(N),A,C)\}$
Return C	Return C
$\textsc{Vf}(i,D,A,C)$	$\textsc{Vf}(i,D,A,C)$
If $(i,D,A,C) \in S$ then return $\perp$	If $(i,D,A,C) \in S$ then return $\perp$
$(M,N) \leftarrow \mathsf{SE.Dec}(K_i,D,A,C)$	Return false
Return $((M,N) \neq \perp)$	

Figure 2: Games defining the AE security of a symmetric encryption scheme SE.

$\times \{0,1\}^{\mathsf{SE.dl}} \times \mathsf{SE.AS} \times \{0,1\}^* \to \mathsf{SE.MS} \times \{0,1\}^{\mathsf{SE.nl}} \times \{0,1\}^{\mathsf{SE.tl}}$ in terms of which the decryption algorithm is specified as follows:

$\mathsf{SE.Dec}(K,D,A,C)$
If $|C| < \mathsf{SE.tl}$ then return $\perp$
$\lambda \leftarrow \mathsf{SE.tl}$; $C^* \leftarrow C[1:|C|-\lambda]$; $T \leftarrow C[\lambda+1:|C|]$
$(M,N,T^*) \leftarrow \mathsf{SE.Tag}(K,D,A,C^*)$
If $(T^* = T)$ then return (M,N) else return $\perp$

We refer to $\mathsf{SE.tl}$ as the tag length. Not every scheme is tag-based, but those built via the Encrypt-then-MAC paradigm [6, 38], including GCM and CCM, are tag-based. The following result shows the relationship between the tag length, the expansion, and the decryption-nonce density.

Lemma 3.1 *Let* SE *be a tag-based symmetric encryption scheme. Then* $\mathsf{SE.tl} \leq \mathsf{SE.ce}(m) - \mathsf{SE.dnd}$ *for every* m.

Proof: Let D be a decryption nonce with the maximum number of preimages N under $\mathsf{SE.df}$, and let S be the set of those preimages. From the definition of decryption-nonce density, we must have $|S| > 2^{\mathsf{SE.dnd}-1}$. Fix a message length m and a key K. Consider the function $f : S \times \{0,1\}^m \to \{0,1\}^{\mathsf{SE.cl}(m)-\mathsf{SE.tl}}$ defined via $f(N,M) = \mathsf{SE.Enc}(K,N,\varepsilon,M)[1:\mathsf{SE.cl}(m)-\mathsf{SE.tl}]$. That is, $f(N,M)$ returns the ciphertext core of the message M that is encrypted under key K, nonce N and the empty AD. The function f is injective, because we can recover (N,M) from $C^* = f(N,M)$ via $\mathsf{SE.Tag}$. Hence $\log_2(|S|) + m \leq \mathsf{SE.cl}(m) - \mathsf{SE.tl}$, because the former is the input length of f and the latter is the output length of f. Hence

$$\mathsf{SE.tl} \leq \mathsf{SE.cl}(m) - m - \log_2(|S|) = \mathsf{SE.ce}(m) - \log_2(|S|) < \mathsf{SE.ce}(m) - \mathsf{SE.dnd} + 1 \ .$$

Since both sides are integers, we have $\mathsf{SE.tl} \leq \mathsf{SE.ce}(m) - \mathsf{SE.dnd}$. ▌

AE SECURITY. Let SE be a symmetric encryption scheme as per the AE3 syntax above. Following the formalizations of [11], we define AE security of SE in the multi-user setting of [12]. Restricting

Game $\mathbf{G}^{\mathrm{cmt}}_{\mathsf{SE}}(\mathcal{A})$

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

$C_1 \leftarrow \mathsf{SE.Enc}(K_1, N_1, A_1, M_1)$; $C_2 \leftarrow \mathsf{SE.Enc}(K_2, N_2, A_2, M_2)$

If $(C_1 \neq C_2)$ then return false

Return $((K_1, N_1, A_1, M_1) \neq (K_2, N_2, A_2, M_2))$

Game $\mathbf{G}^{\text{cmt-d}}_{\mathsf{SE}}(\mathcal{A})$

$\big(C, (K_1, D_1, A_1), (K_2, D_2, A_2)\big) \leftarrow\!\!\$\ \mathcal{A}$

$(M_1, N_1) \leftarrow \mathsf{SE.Dec}(K_1, D_1, A_1, C)$; $(M_2, N_2) \leftarrow \mathsf{SE.Dec}(K_2, D_2, A_2, C)$

If $((M_1, N_1) = \perp$ or $(M_2, N_2) = \perp)$ then return false

Return $((K_1, D_1, A_1) \neq (K_2, D_2, A_2))$

Figure 3: Games defining committing security of a symmetric encryption scheme SE.

attention to AE1 schemes recovers the usual AEAD notion of [39] while restricting attention to AE2 schemes recovers the nonce-hiding notion of [7], but the AE3 formulation here is more general, and useful, beyond unifying prior notions, for the reasons explained above.

Consider games $\mathbf{G}^{\mathrm{real}}_{\mathsf{SE}}(\mathcal{A})$ and $\mathbf{G}^{\mathrm{rand}}_{\mathsf{SE}}(\mathcal{A})$ in Fig. 2. We define the AE advantage of an adversary $\mathcal{A}$ as $\mathbf{Adv}^{\mathrm{ae}}_{\mathsf{SE}}(A) = \Pr[\mathbf{G}^{\mathrm{real}}_{\mathsf{SE}}(\mathcal{A})] - \Pr[\mathbf{G}^{\mathrm{rand}}_{\mathsf{SE}}(\mathcal{A})]$. We require that the adversary be *unique-nonce*, meaning it never repeats an (i, N) pair across its ENC queries. (That is, a nonce is never reused for a given user. Note this implies no entire query is repeated, which is needed for non-triviality.) We stress that there is no such restriction on verification queries. We ask of course that queries are restricted to their domains, for example that $i \in \{1, \ldots, v\}$ and $(N, A, M) \in \{0,1\}^{\mathsf{SE.nl}} \times \mathsf{SE.AS} \times \mathsf{SE.MS}$ in an $\textsc{Enc}(i, N, A, M)$ query.

In the definition above, an adversary can adaptively interleave encryption and verification queries. An adversary is *orderly* if (i) its verification queries are made at the very end, after all its encryption queries are over, and (ii) these queries are non-adaptive, meaning a verification query does not depend on the answers to prior verification queries. (It may still depend on the answers to prior encryption queries.) The following result shows that one can restrict to orderly adversaries with just a small loss in the advantage; it generalizes an AE1 result of [5]. The proof is in Appendix A.

Lemma 3.2 *Let* SE *be a symmetric encryption scheme. Let* $\tau \in \mathbb{N}$ *be such that* $\mathsf{SE.ce}(m) \geq \tau + \mathsf{SE.dnd}$ *for all* $m \in \mathbb{N}$*. (For example, if* SE *is tag-based then* $\tau = \mathsf{SE.tl}$ *can be the tag length.) For any adversary* $\mathcal{A}$ *that makes* q_v *verification queries, we can construct an orderly adversary* $\mathcal{B}$*, of about the same running time and the same query statistics, such that*

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

If $\mathsf{SE.ce}(m) = \tau + \mathsf{SE.dnd}$ *for all* $m \in \mathbb{N}$ *then the term* $2q_v/2^\tau$ *above can be improved to* $q_v/2^\tau$*.*

COMMITTING SECURITY. Two types of committing security are defined in [5]: CMT-1, where the committal is just to the key K, and CMT-4, where the committal is to the entire input (K, N, A, M) to the encryption algorithm. Our paper focuses on the stronger CMT-4 notion, and we just write CMT for it.

The definitions of [5] were for the AE1 setting. Here we extend them to AE3. Specifically, for an adversary $\mathcal{A}$, we define its advantage in breaking the committing security of a symmetric encryption 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. 3.

$\overline{\mathsf{SE}}.\mathsf{Enc}(K, N, A, M)$	$\overline{\mathsf{SE}}.\mathsf{Dec}(K, D, A, C\|T)$
$C\|R \leftarrow \mathsf{SE}.\mathsf{Enc}(K, N, A, M)$	$(M, N, R) \leftarrow \mathsf{SE}.\mathsf{Tag}(K, D, A, C)$
$T \leftarrow H(K, N, A\|R)$	$T^* \leftarrow H(K, N, A\|R)$
Return $C\|T$	If $(T \neq T^*)$ then return $\perp$ else return (M, N)

Figure 4: The AE scheme $\overline{\mathsf{SE}} = \mathsf{CTX}[H, \mathsf{SE}]$.

$\overline{\mathsf{SE}}.\mathsf{Enc}(K, N, A, M)$	$\overline{\mathsf{SE}}.\mathsf{Dec}(K, D, A, C\|T)$
$C\|R \leftarrow \mathsf{SE}.\mathsf{Enc}(K, N, \varepsilon, M)$	$(M, N, R) \leftarrow \mathsf{SE}.\mathsf{Tag}(K, D, \varepsilon, C)$
$T \leftarrow H(K, N, A\|R)$	$T^* \leftarrow H(K, N, A\|R)$
Return $C\|T$	If $(T \neq T^*)$ then return $\perp$ else return (M, N)

Figure 5: The AE scheme $\overline{\mathsf{SE}} = \mathsf{CTY}[H, \mathsf{SE}]$.

Informally, committing security means that an adversary cannot produce a ciphertext collision. We give an alternative, decryption based version, letting $\mathbf{Adv}^{\text{cmt-d}}_{\mathsf{SE}}(\mathcal{A}) = \Pr[\mathbf{G}^{\text{cmt-d}}_{\mathsf{SE}}(\mathcal{A})]$, where game $\mathbf{G}^{\text{cmt-d}}_{\mathsf{SE}}(\mathcal{A})$ is defined in Fig. 3.

To achieve committing security, one only needs to commit to $(K, \mathsf{SE}.\mathsf{df}(N), A)$; the pair (M, N) will be committed via the decryption correctness requirement of SE.

We write $\mathbf{G}^{\text{cmt}}_{\mathsf{SE}}(\mathcal{A}, m)$ for a variant of game $\mathbf{G}^{\text{cmt}}_{\mathsf{SE}}(\mathcal{A})$ in which the adversary is required to output m-bit messages, and let $\mathbf{Adv}^{\text{cmt}}_{\mathsf{SE}}(\mathcal{A}, m)$ be the corresponding advantage.

In Appendix B, we show that CMTD implies CMT in general, and CMT implies CMTD for tidy schemes.

The CTX and CTY transforms. Chan and Rogaway [16] give a transform CTX that adds committing security to a tag-based AE1 scheme. We generalize this to our general setting, continuing to call it CTX. We then give an improvement, CTY, that is more efficient than CTX.

Let SE be a tag-based symmetric encryption scheme with key space $\{0,1\}^k$. Let $H : \{0,1\}^k \times \{0,1\}^{\mathsf{SE}.\mathsf{nl}} \times \{0,1\}^* \to \{0,1\}^\lambda$ be a hash function with output length λ. We assume that SE does not use H, which is often the case in practice. The code of $\overline{\mathsf{SE}} = \mathsf{CTX}[H, \mathsf{SE}]$ is shown in Fig. 4. In the first line of $\overline{\mathsf{SE}}.\mathsf{Enc}$, the output of $\mathsf{SE}.\mathsf{Enc}(K, N, A, M)$ is parsed so that C is the first $\mathsf{SE}.\mathsf{cl}(|M|) - \mathsf{SE}.\mathsf{tl}$ bits and R is the last $\mathsf{SE}.\mathsf{tl}$ bits. It has the same nonce space, AD space, message space, and decryption-nonce derivation function as SE. It is also a tag-based scheme, with tag length λ, the output length of H. The expansion is $\mathsf{CTX}.\mathsf{ce}(m) = \mathsf{SE}.\mathsf{ce}(m) - \mathsf{SE}.\mathsf{tl} + \lambda$. We omit a statement and proof about the security of our general form of CTX because we are going to improve it to CTY.

Note that in CTX, the AD is processed twice, once by SE and another by H. The hashing of AD via H is necessary if one wants CMT security, because if the message is the empty string then the ciphertext is a (constant-sized) commitment of the AD. However, the processing of AD via SE is redundant. We therefore describe an improved transform CTY in Fig. 5. It's the same as CTX but we use the empty AD for SE. The overhead of CTY is the hashing cost of the AD, making CTY optimal in overhead. Again, in the first line of $\overline{\mathsf{SE}}.\mathsf{Enc}$, the output of $\mathsf{SE}.\mathsf{Enc}(K, N, A, M)$ is parsed so that C is the first $\mathsf{SE}.\mathsf{cl}(|M|) - \mathsf{SE}.\mathsf{tl}$ bits and R is the last $\mathsf{SE}.\mathsf{tl}$ bits.

Security of CTY. The following result shows that CTY has CMT security. Intuitively, we commit (K, N, A) via H. Then the message is also committed due to decryption correctness.

Theorem 3.3 *Let* SE *be a tag-based symmetric encryption scheme with key space* $\{0,1\}^k$*. Let* $H : \{0,1\}^k \times \{0,1\}^{\mathsf{SE.nl}} \times \{0,1\}^* \to \{0,1\}^\lambda$ *be a hash function. Then for an adversary* $\mathcal{A}$*, we can construct* $\mathcal{B}$ *such that*

$$\mathbf{Adv}^{\mathrm{cmt}}_{\mathsf{CTY}[H,\mathsf{SE}]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{coll}}_{H}(\mathcal{B}) \ .$$

Adversary $\mathcal{B}$ *runs* $\mathcal{A}$ *and then uses* SE *to encrypt the output of* $\mathcal{A}$*.*

Proof: Adversary $\mathcal{B}$ runs $\mathcal{A}$ to get $\big((K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2)\big)$. It then computes $C_i \| R_i \leftarrow \mathsf{SE}(K_i, N_i, \varepsilon, M_i)$ for each $i \in \{1, 2\}$, and then outputs $\big((K_1, N_1, A_1 \| R_1), (K_2, N_2, A_2 \| R_2)\big)$.

For analysis, without loss of generality, suppose that $\mathcal{A}$ outputs distinct tuples (K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2). Let $T_i \leftarrow H(K_i, N_i, A_i \| R_i)$. Suppose further that $\mathcal{A}$ wins, meaning that $C_1 = C_2$ and $T_1 = T_2$. To show that $\mathcal{B}$ also wins, it suffices to show that the tuples $(K_1, N_1, A_1 \| R_1)$, $(K_2, N_2, A_2 \| R_2)$ are distinct so that $\mathcal{B}$'s output is valid. Assume to the contrary that $(K_1, N_1, A_1, R_1) = (K_2, N_2, A_2, R_2)$. Due to correctness, $M_i \leftarrow \mathsf{SE.Dec}(K_i, N_i, A_i, C_i \| R_i)$, meaning that $M_1 = M_2$. But it means that $(K_1, N_1, A_1, M_1) = (K_2, N_2, A_2, M_2)$, which is a contradiction.

Hence $\mathcal{B}$ also wins, leading the claimed bound. ∎

For the AE security of CTY, at the first glance, it seems to be an easy exercise by modifying the existing proof of CTX. However, Chan and Rogaway [16] only consider a *restricted* setting where the adversary attacks just a single user, and it can only make a single verification query. Translating this result to the general setting via a hybrid argument will lead to a very poor bound. The following result shows that CTY has good AE security in the general setting; the proof is in Appendix C.

Theorem 3.4 *Let* SE *be a tag-based symmetric encryption scheme with key space* $\{0,1\}^k$*. Let* $H : \{0,1\}^k \times \{0,1\}^{\mathsf{SE.nl}} \times \{0,1\}^* \to \{0,1\}^\lambda$ *be a hash function that we will model as a random oracle. Let* $\overline{\mathsf{SE}} = \mathsf{CTY}[H, \mathsf{SE}]$ *as above. Then for an adversary* $\mathcal{A}$ *that attacks* u *users with at most* p *random-oracle queries,* q *encryption queries, and* q_v *verification queries, we can construct* $\mathcal{B}$ *of about the same running time such that*

$$\mathbf{Adv}^{\mathrm{ae}}_{\overline{\mathsf{SE}}}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{ae}}_{\mathsf{SE}}(\mathcal{B}) + \frac{u(u+2p)}{2^k} + \frac{3q_v + q^2}{2^\lambda} \ .$$

It has the same query statistics as $\mathcal{A}$ *plus an encryption query (whose message is just* k*-bit) per user.*

4 Committing Attacks

In this section we give generic committing attacks on any symmetric encryption schemes, assuming that the message length must be exactly m bits. These attacks show that for any symmetric encryption scheme SE, if we want s bits committing security then we must have $\mathsf{SE.ce}(m) \ge \max(s, 2m - s)$. For tag-based schemes (that include CTX and CTY), we give a birthday attack to show that the tag length alone needs to be at least $2s$.

Generic attacks. Let SE be an AE scheme. Fix a number $m \ge 0$ and let $r = \mathsf{SE.ce}(m)$ and $\ell = m + r$.

Consider the following adversary $\mathcal{A}$. It first picks arbitrary distinct tuples $(K_1, N_1, A_1, M_1), \ldots, (K_q, N_q, A_q, M_q)$ with each $|M_i| = m$. It then computes $C_i \leftarrow \mathsf{SE.Enc}(K_i, N_i, A_i, M_i)$ for each $i \le q$. If there are $s \ne t$ such that $C_s = C_t$ then the adversary outputs $((K_s, N_s, A_s, M_s), (K_t, N_t, A_t, M_t))$. Note that each $|C_i| = m + \mathsf{SE.ce}(m) = \ell$. If we model the ciphertexts as uniformly random strings

then from the birthday paradox, $\mathbf{Adv}^{\mathrm{cmt}}_{\mathsf{SE}}(\mathcal{A}, m) = \Omega(q^2)/2^{\ell}$. That is, if we want s bits of committing security then we must have $\ell = \mathsf{SE.ce}(m) + m \geq 2s$, meaning that $\mathsf{SE.ce}(m) \geq 2s - m$.

Next, consider the following adversary $\mathcal{B}$. It first picks arbitrary distinct triples $(K_1, N_1, A_1), \ldots, (K_q, N_q, A_q)$. It then pick an arbitrary m-bit message M_1 and runs $C \leftarrow \mathsf{SE.Enc}(K_1, N_1, A_1, M_1)$. Then for every $i = 2, \ldots, q$, it runs $\mathsf{SE.Dec}(K_i, \mathsf{SE.df}(N_i), A_i, C)$. If there is some s such that the decryption gives a valid message M_s then $\mathcal{B}$ outputs $((K_1, N_1, A_1, M_1), (K_s, N_s, A_s, M_s))$. Let $f_i : \{0,1\}^m \to \{0,1\}^{m+r}$ be the function that

$$f_i(M) = \mathsf{SE.Enc}(K_i, N_i, A_i, M) \ .$$

If we model f_i as a truly random injective function then each decryption independently succeeds with probability $1/2^r$, and thus the chance that at least one of them succeeds is

$$1 - (1 - 1/2^r)^{q-1} \ ,$$

which is $\Omega(q)/2^r$ if $q \leq 2^r$. Hence $\mathbf{Adv}^{\mathrm{cmt}}_{\mathsf{SE}}(\mathcal{B}, m) = \Omega(q)/2^r$. That is, if we want s bits of committing security then we must have $r = \mathsf{SE.ce}(m) \geq s$.

Birthday attack on tag-based schemes. Let SE be a tag-based symmetric encryption scheme. The adversary $\mathcal{A}$ first picks a ciphertext core C^* of length $\mathsf{SE.cl}(m) - \mathsf{SE.tl}$. It then chooses arbitrary distinct $(K_1, N_1), \ldots, (K_q, N_q)$. Then for every $i \leq q$, it runs $(M_i, T_i) \leftarrow \mathsf{SE.Tag}(K_i, \mathsf{SE.df}(N_i), \varepsilon, C^*)$. If there are $s \neq t$ such that $T_s = T_t$ then the adversary outputs $((K_s, N_s, \varepsilon, M_s), (K_t, N_t, \varepsilon, M_t))$. If we model the tentative tags as uniformly randoms strings then due to the birthday paradox, $\mathbf{Adv}^{\mathrm{cmt}}_{\overline{\mathsf{SE}}}(\mathcal{A}, m) = \Omega(q^2)/2^{\mathsf{SE.tl}}$. That is, if we want s bits of committing security then the tag length alone must be at least $2s$.

5 Collision-Resistant IPF

Recall that we want to build a transform to shorten the ciphertext expansion of a committing AE scheme with little security degradation. As a stepping stone for our transform, in this section, we build a collision-resistant invertible PRF. We begin by reviewing the definition of invertible PRF.

Invertible PRF. An *invertible PRF* (IPF) is an injective function $\mathsf{F} : \mathcal{K} \times \mathsf{Dom} \to \mathsf{Rng}$ with an inverse $\mathsf{F}^{-1} : \mathcal{K} \times \mathsf{Rng} \to \mathsf{Dom} \cup \{\perp\}$ such that (i) $\mathsf{F}^{-1}(K, \mathsf{F}(K, P)) = P$ for every $P \in \mathsf{Dom}$ and $K \in \mathcal{K}$, and (ii) $\mathsf{F}^{-1}(K, C) = \perp$ if there is no $P \in \mathsf{Dom}$ such that $\mathsf{F}(K, P) = C$. This primitive arises in the context of deterministic AE (or equivalently, misuse-resistant AE), a solution for the key-wrapping problem [41]. An IPF is a special case of a *pseudorandom injection*, the characterization of deterministic AE, where the nonce and AD are empty.

For an adversary $\mathcal{A}$ and an IPF F, define the advantage of $\mathcal{A}$ breaking the (multi-user) IPF security of F as

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

where games $\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{A})$ is defined in Fig. 6. To avoid trivial wins, we require that the adversary does not repeat prior queries to Enc, and we forbid the adversary from receiving $C \leftarrow \textsc{Enc}(i, P)$ and then later querying $\textsc{Vf}(i, C)$. Informally, the IPF security is the misuse-resistant security [41] of an AE scheme with empty nonce and AD.

Collision-resistant IPF. For our transform, we need an IPF F on domain $\{0,1\}^{\leq m}$ (the set of bit strings of at most m bits) and range $\{0,1\}^{m+s}$. Moreover, we want F to be a collision-resistant hash with (nearly) s-bit security. This requirement is non-trivial. For example, recall that a common approach to build IPF is via the SIV paradigm [41]: on a message M and key K, run $T \leftarrow H(K, M)$, where H is a PRF, and then run a privacy-only encryption (such as CTR) with IV

Game $\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{A})$	NEW()
$v \leftarrow 0$; $b \leftarrow \{0,1\}$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\text{NEW},\text{ENC},\text{VF}}$	$v \leftarrow v+1$; $K_v \leftarrow\!\!\$\ \mathcal{K}$
Return $(b' = b)$	
ENC(i, P)	VF(i, C)
If $i \notin \{1, \ldots, v\}$ then return $\perp$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$C_1 \leftarrow \mathsf{F}(K_i, P)$; $C_0 \leftarrow\!\!\$\ \{0,1\}^{\lvert C_0\rvert}$	$P \leftarrow \mathsf{F}^{-1}(K_i, C)$
Return C_b	If $(b = 1)$ then return $(P \neq \perp)$
	Else return false

Figure 6: Game defining the ipf security of an IPF F.

T to encrypt M. A natural approach to add collision resistance to this IPF is to require that H is collision-resistant. Still, since the IPF accepts m-bit messages and has output length $m + s$ bits, the output length of H is at most s bis, meaning that we only get $s/2$ bits of collision resistance.

Rogaway and Shrimpton [41] suggest two ways to build IPF: (i) the SIV paradigm (which is commonly used in practice), and (ii) the Encode-then-Encipher (EtE) method [8]. Below, we will explore how to build a collision-resistant IPF via both directions.

5.1 Collision-resistant IPF via SIV

THE HtM CONSTRUCTION. Let $n, \tau, d \geq 2$ be integers such that $32 \leq \tau \leq n - d$, and let $\mathcal{M} = \{0,1\}^{\leq n-d}$ (that is, the set of binary strings of at most $n-d$ bits). Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Let $\mathsf{pad} : \mathcal{M} \times \{0,1\} \to \{0,1\}^n$ be a padding mechanism such that: (1) $\mathsf{pad}(X, 1) = X \| 1^{n-|X|}$ for every $X \in \mathcal{M}$, (2) pad is a domain separation, meaning that $\mathsf{pad}(X, 0) \neq \mathsf{pad}(Y, 1)$ for every $X, Y \in \mathcal{M}$, and (3) there is an associated unpad such that $\mathsf{unpad}(\cdot)$ and $\mathsf{pad}(\cdot, 0)$ are the inverse of each other.[1]

Our IPF Hash-then-Mask (HtM) with domain $\mathcal{M}$ and range $\{0,1\}^{n+\tau}$ is shown in Fig. 7. Informally, given a key K and plaintext P, we pad $M \leftarrow \mathsf{pad}(P, 0)$, and then hash (K, M) to derive a τ-bit tag T. Then, hash $(K, \mathsf{pad}(T, 1))$ to obtain a one-time pad to mask M, resulting in a string C^*. The output is $C^* \| T$. The hash function H in both cases is the Davies-Meyer construction on E, meaning $H(K, M) = E_K(M) \oplus M$.

Our construction HtM follows the SIV paradigm of Rogaway and Shrimpton [41] in building misuse-resistant AE schemes: first use a PRF on the message to derive a synthetic IV, and then use a privacy-only encryption scheme on that IV to encrypt the message. Due to the committing security requirement, here both the PRF and the privacy-only encryption scheme are built on top of the Davies-Meyer construction. Alternatively, one can view HtM as a two-round (unbalanced) Feistel network, where the left half is 0^τ and the right half is the padded message, and the round functions are implemented via Davies-Meyer.

INSTANTIATION. We instantiate E via AES, meaning $n = 128$. For the ciphertext of HtM to be a byte string, we pick $d = 8$. For an m-bit string M with $m \leq 120$, let

$$\mathsf{pad}(M, 0) = M \| 0^{120-m} \| [m]_8 \ ,$$

where $[m]_8$ denotes the 8-bit encoding of the number m. For $X \in \{0,1\}^n$, let $\mathsf{unpad}(X)$ be the

[1] For example, we can let $\mathsf{pad}(M, 0) = M \| 10^{n-|M|-1}$. For $\mathsf{unpad}(Y)$, we first obtain a string X by removing the trailing 0's and the last 1 of Y. If $Y \neq 0^n$ and $|X| \leq n - d$ then we return X; otherwise we return $\perp$. Later we will give a more efficient instantiation of pad and unpad where $n = 128$ and $d = 8$.

$\underline{\mathsf{F}(K,P)}$	$\underline{\mathsf{F}^{-1}(K,C^*\|T)}$
$M \leftarrow \mathsf{pad}(P,0)$	$Z \leftarrow \mathsf{pad}(T,1)$
$T \leftarrow \big(E_K(M)\oplus M\big)[1:\tau]$	$M \leftarrow \big(E_K(Z)\oplus Z\big)\oplus C^*$
$Z \leftarrow \mathsf{pad}(T,1)$	$V \leftarrow (E_K(M)\oplus M)[1:\tau]$
$C^* \leftarrow \big(E_K(Z)\oplus Z\big)\oplus M$	If $(T \neq V)$ then return $\perp$
Return $C^*\|T$	Return $\mathsf{unpad}(M)$

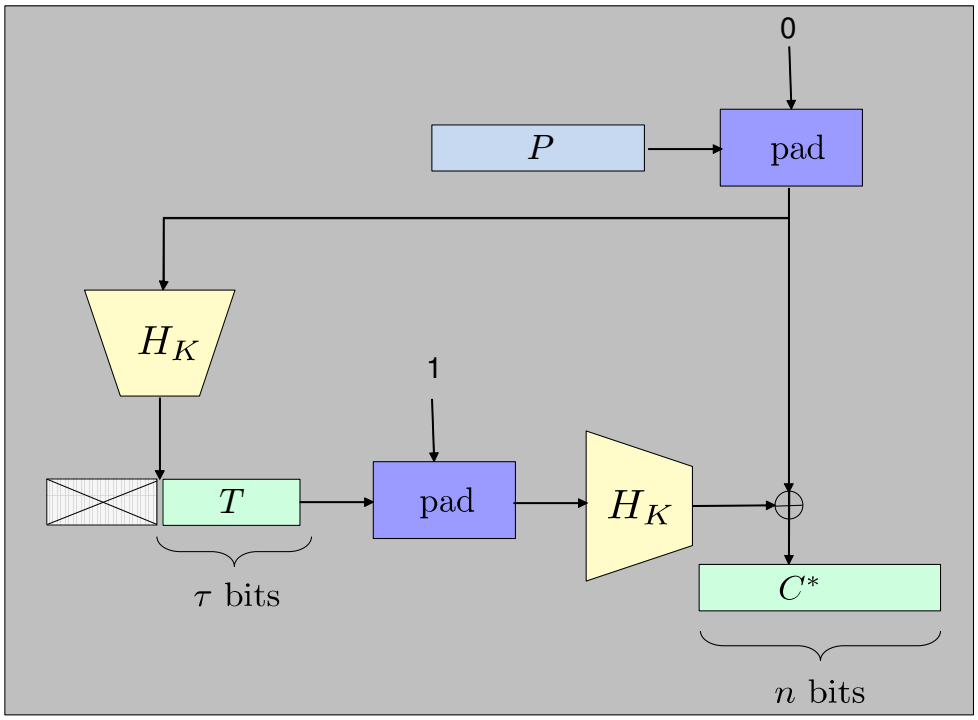

Figure 7: The IPF $\mathsf{F} = \mathsf{HtM}[E,\tau]$. In the illustration, the function H denotes the Davies-Meyer construction on E, meaning $H(K,M) = E_K(M)\oplus M$.

---

string obtained by (i) parsing the last byte of X as a number m, and (ii) returning $X[1:m]$ if $m \leq 120$ and $X[m+1:120] = 0^{120-m}$, and returning $\perp$ otherwise.

Collision resistance of HtM. At the first glance, since HtM produces a τ-bit tag via a collision-resistant hash function, it seems that it only has $\tau/2$ bits of collision resistance. Surprisingly, the following Proposition 5.1 shows that HtM achieves around $\tau - \lceil \log_2(\tau) \rceil$ bits of collision resistance in the ideal-cipher model; the proof is deferred to Section 5.3. The root of the strong security of HtM lies in the circularity of first deriving T from hashing P, and then hashing T to mask P. Without this circularity, the collision resistance would drop to $\tau/2$ bits.

To see that the bound in Proposition 5.1 is essentially tight, note that from the generic attacks in Section 4, one can build an adversary $\mathcal{B}$ that makes q calls to HtM (namely $2q$ calls to E), and $\mathbf{Adv}^{\mathrm{cmt}}_{\mathsf{HtM}[E,\tau]}(\mathcal{B},m) = \Omega(q)/2^{\tau+d}$. Moreover, recall that d is small; for example, $d = 8$ in our instantiation, or $d = 2$ if one wants to maximize HtM security. Hence the bound in Proposition 5.1 nearly matches the generic attacks.

The proof of Proposition 5.1 also contains terms like $q^2/2^{n+\tau}$ that correspond to actual attacks. Still, those are dominated by the term $q/2^{\tau}$, and thus we upper-bound them by this dominant term so that the final bound is simple.

Proposition 5.1 *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 $\mathsf{HtM}[E,\tau]$ be as above. Consider an adversary $\mathcal{A}$ that makes at most q ideal-cipher queries. Then*

$$\mathbf{Adv}^{\mathrm{coll}}_{\mathsf{HtM}[E,\tau]}(\mathcal{A}) \le \frac{4(n+\tau)q+5}{2^\tau} \ .$$

REMARK. Here we need HtM to commit both the input and the key. But in applications, we actually need to commit just the key. In that case, Proposition 5.1 doesn't need $\mathsf{pad}(\cdot,0)$ to be invertible, and thus for the parameter d in HtM, we only need $d \ge 1$ (instead of $d \ge 2$).

IPF SECURITY OF HtM. The following Proposition 5.2 analyzes the IPF security of $\mathsf{HtM}[E,\tau]$. The proof is in Appendix E.

Proposition 5.2 *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher and let $\mathsf{HtM}[E,\tau]$ be as above. For an adversary $\mathcal{A}$ that makes at most q_e encryption queries and q_v verification queries, we can construct an adversary $\mathcal{B}$ of about the same time and $2(q_e+q_v)$ queries such that*

$$\mathbf{Adv}^{\mathrm{ipf}}_{\mathsf{HtM}[E,\tau]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{B}) + \frac{1.5q_v}{2^\tau} + \frac{q_e(B-1)}{2^\tau} \ .$$

where B is the maximum number of encryption queries per user of $\mathcal{A}$.

DISCUSSION. At the first glance, the term $q_e(B-1)/2^\tau$ looks inferior for small choices of τ, such as $\tau = 88$. However, for the SC transform in Section 6, we only have one encryption query per user, meaning $B = 1$ and the term $q_e(B-1)/2^\tau$ vanishes.

One the other hand, in the decryption of HtM, an invalid ciphertext can be rejected for bad tag or improper padding, meaning there is a potential timing issue. However, note that an invalid ciphertext is very likely rejected because of its bad tag. Thus as long as one checks the tag before unpadding, as in the code of Fig. 7, then one can avoid the timing issue. The proof in Proposition 5.2 rigorously confirms this intuition, showing that HtM is a good IPF even in the presence of timing leakage. In particular, in the games, verification queries only consider tag checking. That is, if an adversary can launch a verification query that passes the tag checking but still ends up with a bad-padding rejection, it will get a true answer (instead of false) and win the game, knowing that it's in the real world.

ON HIGHER SECURITY LEVEL. Our work targets (nearly) 128-bit committing security. If one instead wants 256-bit security, one can instantiate the blockcipher in HtM via Rijndael-256 [20] but this is not a standardized primitive. Alternatively, instead of using the Davies-Meyer construction for HtM, one can use a cryptographic hash function like SHA-3 or (truncated) SHA-512, and model it as a random oracle. The downside of this approach is that the cost will be a lot more expensive.

5.2 Collision-resistant IPF via Encode-then-Encipher

The SIV approach is not the only way to build collision-resistant IPF. We now consider another direction via the Encode-then-Encipher paradigm [8].

THE PtE CONSTRUCTION. Let $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher, and let $s \ge 1$ be an integer. Let $\mathcal{M} = \{0,1\}^{\le n-s}$ (meaning the set of bit strings at most $n-s$ bits). Let $\mathsf{pad} : \mathcal{M} \to \{0,1\}^n$ be a padding mechanism with an associated inverse unpad.[2] Our construction Pad-then-Encipher (PtE) is given in Fig. 8; it has key space $\{0,1\}^k$, domain $\mathcal{M}$, and range $\{0,1\}^n$.

[2]For example, we can let $\mathsf{pad}(M) = M \| 10^{n-1-|M|}$. Conversely, $\mathsf{unpad}(Y)$ returns $\perp$ if $Y = 0^n$ or if Y doesn't end with 0^{s-1}. Otherwise, return the string X obtained by removing the trailing 0's and the last bit 1 of Y.

$\underline{\mathsf{F}(K,P)}$	$\underline{\mathsf{F}^{-1}(K,C)}$
$M \leftarrow \mathsf{pad}(P)$; $C \leftarrow E(K,M)$	$M \leftarrow E^{-1}(K,C)$
Return C	Return $\mathsf{unpad}(M)$

Figure 8: The IPF $\mathsf{F} = \mathsf{PtE}[E]$.

IPF SECURITY OF PtE. The following result shows that PtE is a good IPF; the proof is in Appendix 5.3. This is expected, following standard results of the Encode-then-Encipher paradigm [30].

Proposition 5.3 *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Let $\mathsf{F} = \mathsf{PtE}[E]$. Then for any adversary $\mathcal{A}$ making at most q queries (with at most B queries per user), we can construct an adversary $\mathcal{B}$ of about the same running time and the same query statistics such that*

$$\mathbf{Adv}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{A}) \leq \mathbf{Adv}^{\pm\mathrm{prp}}_{E}(\mathcal{B}) + \frac{4q}{2^s} + \frac{qB}{2^n} \quad .$$

COLLISION RESISTANCE OF PtE. The following result shows that PtE has s bits of collision resistance in the ideal-cipher model, assuming that $n \geq 2s$. The proof is in Appendix G. The bound is tight, matching the generic attacks in Section 4.

Here we need PtE to commit both the input and the key. But in applications, we actually need to commit just the key. In that case, Proposition 5.4 doesn't need pad to be invertible.

Proposition 5.4 *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 $\mathsf{F} = \mathsf{PtE}[E]$. Then for any adversary $\mathcal{A}$ making at most q ideal-cipher queries,*

$$\mathbf{Adv}^{\mathrm{coll}}_{\mathsf{F}}(\mathcal{A}) \leq \frac{4q}{4^s} + \frac{2q^2}{2^n} \quad .$$

DISCUSSION. Since we want $s \geq 128$ bits of collision resistance, we need $n \geq 256$. As such, one can't instantiate E directly from AES. One may consider instantiating E using a Feistel network via indifferentiability theory [34] but this runs into several obstacles as mentioned in the Introduction. Another approach is to instantiate E from Rijndael-256 [20] but this is not a standardized primitive.

5.3 Proof of Proposition 5.1

SETUP. 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 $C \leftarrow E_K(M)$ then later it will not query $E_K^{-1}(C)$, and if it queries $M \leftarrow E_K^{-1}(C)$ then it will not later query $E_K(M)$. Without loss of generality, assume that the right-hand side of the claimed bound is smaller than 1; otherwise the bound is moot.

For a query $C \leftarrow E_K(\mathsf{pad}(P,0))$, we store a log entry $\big(+, K, \mathsf{pad}(P,0), (C \oplus \mathsf{pad}(P,0))[1:\tau]\big)$. For a query $C \leftarrow E_K(\mathsf{pad}(T,1))$ with $|T| = \tau$, we store a log entry $\big(+, \mathsf{pad}(T,1), C \oplus \mathsf{pad}(T,1)\big)$. Analogously, for a query $M \leftarrow E_K^{-1}(C)$, if M can be parsed as $\mathsf{pad}(P,0)$ then we also store $\big(-, K, M, (C \oplus M)[1:\tau]\big)$. Otherwise, if M can be parsed as $\mathsf{pad}(T,1)$ with $|T| = \tau$ then we also store a log entry $\big(-, K, M, C \oplus M\big)$.

If a query results in a log entry $(\cdot, K, \mathsf{pad}(P,0), T)$, and there is no prior entry $(\cdot, K, \mathsf{pad}(T,1), \cdot)$, then we immediately grant it a free query $E_K(\mathsf{pad}(T,1))$ and store the corresponding log entry.

These free queries can only help $\mathcal{A}$. Note that for each log entry $(\cdot, K, \mathsf{pad}(T,1), X)$, conditioning on the prior logs, the string X is uniformly distributed over a set of at least $2^n - 2q \geq (16 \cdot 2^n)/15$ members. Likewise, for each log entry $(\cdot, K, \mathsf{pad}(P,0), T)$, the full Davies-Meyer output V of T (meaning $T = V[1:\tau]$) is uniformly distributed over a set of at least $(16 \cdot 2^n)/15$ members.

Two entries $(\cdot, K, \mathsf{pad}(P,0), T)$ and $(\cdot, K, \mathsf{pad}(T,1), \cdot)$ are called a *match*, where $(\cdot, K, \mathsf{pad}(P,0), T)$ is the *first entry* and $(\cdot, K, \mathsf{pad}(T,1), \cdot)$ is the *second entry*. A match is *even* if it contains a granted query; otherwise we call it *odd*. From the way we grant free queries, in an odd match, the second query is made *before* its first query, but in an even match, its second query is made *right after* its first query.

For a match Q, let $\mathsf{sign1}(Q)$ denote the sign of its first entry, and let $\mathsf{sign2}(Q)$ denote the sign of the second entry. Note that if Q is even then $\mathsf{sign2}(Q) = +$. Note that there are at most q matches, as there are at most q log entries $(\cdot, K, \mathsf{pad}(P,0), T)$, and each such entry has at most one partner $(\cdot, K, \mathsf{pad}(T,1), \cdot)$.

We say that a match Q is a *prior* match of another match Q^* if the last query of Q^* is made after both queries in Q. Moreover, if both queries of Q^* are made after the queries of Q then we say that Q^* *succeeds* Q. Two matches $Q = \{(\cdot, K, \mathsf{pad}(P,0), T), (\cdot, K, \mathsf{pad}(T,1), X)\}$ and $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*,0), T^*), (\cdot, K^*, \mathsf{pad}(T^*,1), X^*)\}$ are *compatible* if $K \neq K^*$, $T = T^*$, and $X \oplus X^* = \mathsf{pad}(P,0) \oplus \mathsf{pad}(P^*,0)$. The adversary's goal is to find two compatible matches among its queries.

A BALLS-INTO-BINS RESULT. Our proof heavily relies on many non-standard balls-into-bins games whose analyses are based on the following lemma. The proof in in Appendix D.1.

Lemma 5.5 *Let $q \geq 1$ and $m \geq 32$ be integers. Consider a q-round game in which for each round, we probabilistically pick some bins out of 2^m ones and put a ball in each bin. Assume that for each fixed bin, in each round, given the throws of prior rounds, the conditional probability that we place a ball on this bin is at most $c/2^m$. If $3qc \leq 2^m$ then with probability at least $1 - 2^{-m}$, every bin has at most $m/2$ balls.*

A USEFUL INEQUALITY. Our analyses rely on the following result from Bellare and Hoang [4]. It resembles the Bernoulli's inequality $(1-a)^p \geq 1 - ap$ but reverses the inequality sign.

Lemma 5.6 [4] *Let $p \geq 1$ be an integer and $a \geq 0$ be a real number. Assume $ap \leq 1$. Then $(1-a)^p \leq 1 - ap/2$.*

COLLISION RESISTANCE OF HtM. Suppose that $\mathcal{A}$ outputs (K_1, P_1, K_2, P_2). Let $C_1^* \| T_1$ and $C_2^* \| T_2$ be the resulting ciphertexts. If $T_1 = T_2$ and $C_1^* = C_2^*$ then we must have $K_1 \neq K_2$, otherwise P_1 and P_2 are the same due to the perfect correctness of HtM, meaning that this output is invalid. Hence without loss of generality, we can assume that $K_1 \neq K_2$.

We first consider the case that there is no log entry $(\cdot, K_1, \mathsf{pad}(P_1,0), \cdot)$ or $(\cdot, K_2, \mathsf{pad}(P_2,0), \cdot)$. Without loss of generality, assume that there is no log $(\cdot, K_1, \mathsf{pad}(P_1,0), \cdot)$. Then given T_2, the full Davies-Meyer output V of T_1 is uniformly chosen within a set of at least $15 \cdot 2^n/16$ elements. Among them, there are $2^{n-\tau}$ strings R such that $R[1:\tau] = T_2$. Since $T_1 \leftarrow V[1:\tau]$, the event $T_1 = T_2$ happens if and only if the random variable V lands among the $2^{n-\tau}$ strings above, and this happens with probability at most $2^{n-\tau} \cdot 16/(15 \cdot 2^n) = 16/(15 \cdot 2^\tau)$.

From now on, we will assume that the adversary's queries create log entries $(\cdot, K_1, \mathsf{pad}(P_1,0), \cdot)$ and $(\cdot, K_2, \mathsf{pad}(P_2,0), \cdot)$. From the way that we grant free queries, these logs will lead to the corresponding matches Q_1 and Q_2. Moreover, if the adversary creates a collision then Q_1 and Q_2

must be compatible. Without loss of generality, assume that Q_1 is a prior match of Q_2. We consider the following cases; summing the bounds over those cases lead to the claimed result.

Case 1: Q_2 is even. From the way we grant queries, this means that Q_2 succeeds Q_1. Consider two matches $Q = \{(\cdot, K, \mathsf{pad}(P, 0), T), (+, K, \mathsf{pad}(T, 1), X)\}$ and $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*, 0), T^*), (+, K^*, \mathsf{pad}(T^*, 1), X^*)\}$ such that Q is even and Q succeeds Q^*. Then given T^*, the full Davies-Meyer output V of T is uniformly distributed over a set of at least $15 \cdot 2^n/16$ elements, and thus the probability that $T \leftarrow V[1:\tau]$ hits T^* is at most $16/(15 \cdot 2^\tau)$. Furthermore, given prior queries, the conditional probability that $X = X^* \oplus \mathsf{pad}(P, 0) \oplus \mathsf{pad}(P^*, 0)$ is at most $16/(15 \cdot 2^n)$. Hence the chance that Q and Q^* are compatible is at most $256/(225 \cdot 2^{\tau+n}) \leq 2/2^{\tau+n}$. Summing this over at most $\binom{q}{2} \leq q^2/2$ pairs of matches, the chance that this case happens is at most $q^2/2^{\tau+n} \leq q/2^n$.

Case 2: Q_2 is odd, and Q_1 is even. We consider the following sub-cases.

Case 2.1: $\mathsf{sign1}(Q_2) = +$. Each query $V \leftarrow E(K, \mathsf{pad}(P, 0))$ aims at a prior entry $(\cdot, K, \mathsf{pad}(T, 1), X)$ and a prior even match $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*, 0), T), (+, K^*, \mathsf{pad}(T, 1), X^*)\}$ such that

$$V[1:\tau] = T \oplus \mathsf{pad}(P, 0)[1:\tau] \ , \text{ and} \tag{1}$$

$$\mathsf{pad}(P, 0) = \mathsf{pad}(P^*, 0) \oplus X^* \oplus X \ . \tag{2}$$

The difficulty here is that $\mathcal{A}$ may *adaptively* choose $(K, \mathsf{pad}(P, 0))$ to maximize the number of entry-match pairs that satisfy Equation (2). Still, we claim that with probability at least $1 - 4q/2^n$, any query $E(K, \mathsf{pad}(P, 0))$ will have at most $3n/2$ pairs that satisfy Equation (2). Then, the chance that the answer V of this query will satisfy Equation (1) with one of those $3n/2$ pairs is at most

$$\frac{3n}{2} \cdot \frac{16}{15 \cdot 2^n} \cdot \frac{1}{2^{n-\tau}} = \frac{24n}{15 \cdot 2^\tau} \ .$$

Summing over all queries, the chance this case happens is at most

$$\frac{24qn}{15 \cdot 2^\tau} + \frac{4q}{2^n} \ .$$

To justify the claim above, view each log entry $(\cdot, K, \mathsf{pad}(T, 1), X)$ as a round in a ball-throwing game. In each round, for any string $\Delta \in \{0,1\}^n$, we throw a ball to bin (K, Δ) if there exists an even match $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*, 0), T), (+, K^*, \mathsf{pad}(T, 1), X^*)\}$ such that $K \neq K^*$ and $X^* \oplus X \oplus \mathsf{pad}(P^*, 0) = \Delta$. (If there are multiple such matches then we still throw a single ball to bin (K, Δ).) Lemma 5.7 below bounds the number of balls in the heaviest bin; the proof is in Appendix D.2.

Lemma 5.7 *For the game of ball throwing above, with probability at least* $1 - 4q/2^n$*, every bin has at most* $3n/2$ *balls.*

Now, for each query $E(K, \mathsf{pad}(P, 0))$, its number of entry-match pairs that satisfy Equation (2) is exactly the number of balls in bin $(K, \mathsf{pad}(P, 0))$. Thus the claim above directly follows Lemma 5.7.

Case 2.2: $\mathsf{sign1}(Q_2) = -$. Each query $\mathsf{pad}(P, 0) \leftarrow E^{-1}(K, V)$ aims at a prior entry $(\cdot, K, \mathsf{pad}(T, 1), X)$ and a prior even match $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*, 0), T), (+, K^*, \mathsf{pad}(T, 1), X^*)\}$ such that

$$\mathsf{pad}(P, 0)[1:\tau] = T \oplus V[1:\tau] \ , \text{ and} \tag{3}$$

$$\mathsf{pad}(P, 0) = \mathsf{pad}(P^*, 0) \oplus X^* \oplus X \ . \tag{4}$$

From Equation (3), we can replace Equation (4) with

$$V[1:\tau] = (\mathsf{pad}(P^*, 0) \oplus X^* \oplus X)[1:\tau] \oplus T \ . \tag{5}$$

We claim that with probability at least $1 - 3q/2^\tau - q/2^n$, any query $E^{-1}(K, V)$ will have at most $(\tau + n/2)$ entry-match pairs that satisfy Equation (5). Then, the chance that the answer $\mathsf{pad}(P, 0)$

of this query will satisfy Equation (3) with one of those $(\tau + n/2)$ pairs is at most

$$(\tau + n/2) \cdot \frac{16}{15 \cdot 2^n} \cdot \frac{1}{2^{n-\tau}} = \frac{16\tau + 8n}{15 \cdot 2^\tau} \ .$$

Summing over all queries, the chance this case happens is at most

$$\frac{(16\tau + 8n)q}{15 \cdot 2^\tau} + \frac{3q}{2^\tau} + \frac{q}{2^n} \ .$$

To justify the claim above, consider the following balls-into-bins game. For each log entry $(\cdot, K, \mathsf{pad}(T, 1), X)$ and an even match $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*, 0), T), (+, K^*, \mathsf{pad}(T, 1), X^*)\}$, with $K \neq K^*$, throw a ball to bin

$$(K, (X \oplus X^* \oplus \mathsf{pad}(P^*, 0))[1 : \tau] \oplus T) \ .$$

Lemma 5.8 below bounds the number of balls in the heaviest bin; the proof is in Appendix D.3.

Lemma 5.8 *For the game of ball throwing above, with probability at least $1 - 3q/2^\tau - q/2^n$, every bin has at most $\tau + n/2$ balls.*

Now, for each query $E^{-1}(K, V)$, its number of entry-match pairs that satisfy Equation (2) is exactly the number of balls in bin $(K, V[1 : \tau])$. Thus the claim above directly follows Lemma 5.8.

Case 3: Both Q_1 and Q_2 are odd, and Q_2 succeeds Q_1. For each log entry $(\cdot, K, \mathsf{pad}(T, 1), X)$, its *target* is a prior odd match $Q^* = \{(\cdot, K^*, \mathsf{pad}(T, 1), X^*), (\cdot, K^*, \mathsf{pad}(P^*, 0), T)\}$, with $K^* \neq K$. While a log entry may have many targets, we claim that with probability at least $1 - q/2^\tau$, each log entry has at most $\tau/2$ targets. Indeed, view each $(\cdot, K^*, \mathsf{pad}(P^*, 0), T)$ as throwing a ball into bin T. For each throw, given the prior throws, the chance that it lands into any particular bin is at most $16/(15 \cdot 2^\tau)$. The claim then directly follows Lemma 5.5 with $c = 16/15$ and $m = \tau$.

Suppose that each entry has at most $\tau/2$ targets. We consider the following sub-cases.

Case 3.1: $\mathsf{sign1}(Q_2) = +$ and $\mathsf{sign2}(Q_2) = +$. Each query $V \leftarrow E(K, \mathsf{pad}(P, 0))$ aims at a prior entry $(+, K, \mathsf{pad}(T, 1), X)$ with a target $Q^* = \{(\cdot, K^*, \mathsf{pad}(T), X^*), (\cdot, K^*, \mathsf{pad}(P^*, 0), T)\}$ such that

$$V[1 : \tau] = T \oplus \mathsf{pad}(P, 0)[1 : \tau] \ , \text{ and} \tag{6}$$

$$\mathsf{pad}(P, 0) = \mathsf{pad}(P^*, 0) \oplus X^* \oplus X \ . \tag{7}$$

We claim that with probability at least $1 - q/2^n$, any query $E(K, \mathsf{pad}(P, 0))$ has at most $n/2$ entry-target pairs that satisfy Equation (7). Then the chance that the answer V of this query satisfies Equation (6) with one of those $n/2$ pairs is at most

$$\frac{n}{2} \cdot \frac{16}{15 \cdot 2^n} \cdot \frac{1}{2^{n-\tau}} = \frac{8n}{15 \cdot 2^\tau} \ .$$

Summing over all queries $E(K, \mathsf{pad}(P, 0))$, the chance that this case happens is at most

$$\frac{8qn}{15 \cdot 2^\tau} + \frac{q}{2^n} \ .$$

To justify the claim above, fix a key K. View each log entry $(+, K, \mathsf{pad}(T, 1), X)$ as a round in a ball-throwing game. In each round, for each $\Delta \in \{0, 1\}^n$, we throw a ball to bin $(K, X \oplus \Delta)$ if this log entry has a target $Q^* = \{(\cdot, K^*, \mathsf{pad}(T, 1), X^*), (\cdot, K^*, \mathsf{pad}(P^*, 0), T)\}$ such that $X^* \oplus \mathsf{pad}(P^*, 0) = \Delta$. (If there are multiple such targets then we still throw a ball to bin $(K, X \oplus \Delta)$.) Since each log entry has at most $\tau/2$ targets, for each particular bin, the chance that this round puts a ball to this bin is at most $8\tau/(15 \cdot 2^n)$. From Lemma 5.5 with $c = 8\tau/15$ and $m = n$, for any fixed key K, with probability at least $1 - 1/2^n$, each bin associated with key K has at most $n/2$ balls. Summing over q possible keys, with probability at least $1 - q/2^n$, every bin has at most $n/2$ balls.

Now, for any query $E(K, \mathsf{pad}(P, 0))$, its number of entry-target pairs that satisfy Equation (7) is exactly the number of balls in bin $(K, \mathsf{pad}(P, 0))$. Thus our claim directly follows the balls-into-bins analyses above.

Case 3.2: $\mathsf{sign1}(Q_2) = -$ and $\mathsf{sign2}(Q_2) = +$. Each query $\mathsf{pad}(P, 0) \leftarrow E^{-1}(K, V)$ aims at a prior entry $(+, K, \mathsf{pad}(T, 1), X)$ with a target $Q^* = \{(\cdot, K^*, \mathsf{pad}(T), X^*), (\cdot, K^*, \mathsf{pad}(P^*, 0), T)\}$ such that

$$\mathsf{pad}(P, 0)[1 : \tau] = V[1 : \tau] \oplus T \ , \text{ and} \tag{8}$$

$$\mathsf{pad}(P, 0) \oplus X = \mathsf{pad}(P^*, 0) \oplus X^* \ . \tag{9}$$

From Equation (8), we can replace Equation (9) with

$$V[1 : \tau] = X[1 : \tau] \oplus T \oplus (X^* \oplus \mathsf{pad}(P^*, 0))[1 : \tau] \ . \tag{10}$$

We claim that with probability at least $1 - q/2^\tau$, for each fixed query $E^{-1}(K, V)$, there are at most $\tau/2$ entry-target pairs that satisfy Equation (10). Then the chance that the answer $\mathsf{pad}(P, 0)$ of this query satisfies Equation (8) with one of those $\tau/2$ pairs is at most $8\tau/(15 \cdot 2^\tau)$. Summing over all queries $E^{-1}(K, V)$, the chance that this case happens is at most

$$\frac{8q\tau}{15 \cdot 2^\tau} + \frac{q}{2^\tau} \ .$$

To justify the claim above, fix a key K. View each log entry $(+, K, \mathsf{pad}(T, 1), X)$ as a round of a ball-throwing game. In this round, for each $\Delta \in \{0, 1\}^\tau$, we throw a ball to bin $(K, X[1 : \tau] \oplus \Delta)$ if this log entry has a target $Q^* = \{(\cdot, K^*, \mathsf{pad}(T), X^*), (\cdot, K^*, \mathsf{pad}(P^*, 0), T)\}$ such that

$$(X^* \oplus \mathsf{pad}(P^*, 0))[1 : \tau] \oplus T = \Delta \ .$$

Since each log entry has at most $\tau/2$ targets, for each particular bin, the chance that this round puts a ball to this bin is at most $8\tau/(15 \cdot 2^\tau)$. From Lemma 5.5 with $c = 8\tau/15$ and $m = \tau$, for each fixed key K, with probability at least $1 - 1/2^\tau$, each bin associated with key K has at most $\tau/2$ balls. Summing over q possible keys, with probability at least $1 - q/2^\tau$, every bin has at most $\tau/2$ balls.

Now, for any query $E^{-1}(K, V)$, its number of entry-target pairs that satisfy Equation (10) is exactly the number of balls in bin $(K, V[1 : \tau])$. Thus our claim directly follows the balls-into-bins analyses above.

Case 3.3: $\mathsf{sign1}(Q_2) = -$ and $\mathsf{sign2}(Q_2) = -$. We need the following technical lemma Lemma 5.9. The proof is in Appendix D.4.

Lemma 5.9 *With probability at least* $1-(q+3)/2^\tau$*, for every* $\Delta \in \{0, 1\}^\tau$*, there are at most* $\tau/2$ *odd matches* $Q^* = \{(\cdot, K^*, \mathsf{pad}(T, 1), X^*), (\cdot, K^*, \mathsf{pad}(P^*, 0), T)\}$ *such that* $(X^* \oplus \mathsf{pad}(P^*, 0))[1 : \tau] = \Delta$*.*

Below, we will assume that the event in Lemma 5.9 happens. Now, each query $\mathsf{pad}(P, 0) \leftarrow E^{-1}(K, V)$ aims at a prior query $\mathsf{pad}(T, 1) \leftarrow E^{-1}(K, U)$ with a target $Q^* = \{(\cdot, K^*, \mathsf{pad}(T), X^*), (\cdot, K^*, \mathsf{pad}(P^*, 0), T)\}$ such that

$$\mathsf{pad}(P, 0)[1 : \tau] = V[1 : \tau] \oplus T \ , \text{ and} \tag{11}$$

$$\mathsf{pad}(P, 0) \oplus (U \oplus \mathsf{pad}(T, 1)) = \mathsf{pad}(P^*, 0) \oplus X^* \ . \tag{12}$$

From Equation (11) and the fact that $T = \mathsf{pad}(T, 1)[1 : \tau]$, we can replace Equation (12) with

$$V[1 : \tau] = (X^* \oplus \mathsf{pad}(P^*, 0) \oplus U)[1 : \tau] \ . \tag{13}$$

We claim that with probability at least $1 - q/2^\tau$, for each query $E^{-1}(K, V)$, there are at most $\tau/2$ query-target pairs that satisfy Equation (13). Then the chance that the answer $\mathsf{pad}(P, 0)$ of this

query satisfies Equation (11) with one of those $\tau/2$ pairs is at most $8\tau/(15 \cdot 2^\tau)$. Summing over all queries $E^{-1}(K, V)$, the chance that this case happens is at most

$$\frac{8q\tau}{15 \cdot 2^\tau} + \frac{q}{2^\tau} \ .$$

Still, we need to add another term $(q+3)/2^\tau$ to account for the assumption that the event of Lemma 5.9 happens. Thus this case actually happens with probability at most

$$\frac{8q\tau}{15 \cdot 2^\tau} + \frac{2q+3}{2^\tau}$$

To justify the claim above, fix a key K and view each query $E^{-1}(K, U)$ as a round of a ball-throwing game. From Lemma 5.9, for any $\Delta \in \{0,1\}^\tau$, there are at most $\tau/2$ prior odd matches $Q^* = \{(\cdot, K^*, \mathsf{pad}(T,1), X^*), (\cdot, K^*, \mathsf{pad}(P^*,0), T)\}$ such that $(X^* \oplus \mathsf{pad}(P^*,0))[1:\tau] = \Delta \oplus U[1:\tau]$. In this round, we will put a ball to bin Δ if the answer of this query hits the $\mathsf{pad}(T,1)$ of one of those matches, which happens with probability at most $8\tau/(15 \cdot 2^n) \leq 8\tau/(15 \cdot 2^\tau)$. From Lemma 5.5 with $c = 8\tau/15$ and $m = \tau$, for each fixed key K, with probability at least $1 - 2^{-\tau}$, every bin associated with K has at most $\tau/2$ balls. Summing over all q possible keys, with probability at least $1 - q/2^\tau$, every bin has at most $\tau/2$ balls.

Now, for any query $E^{-1}(K, V)$, its number of entry-target pairs that satisfy Equation (13) is exactly the number of balls in bin $(K, V[1:\tau])$. Thus our claim directly follows the balls-into-bins analyses above.

Case 3.4: $\mathsf{sign1}(Q_2) = +$ and $\mathsf{sign2}(Q_2) = -$. We need the following technical lemma Lemma 5.10. The proof is in Appendix D.5.

Lemma 5.10 *With probability at least $1 - 2^{-\tau} - 2^{-n}$, for every $\Delta \in \{0,1\}^n$, there are at most $n/2$ odd matches $Q^* = \{(\cdot, K^*, \mathsf{pad}(T,1), X^*), (\cdot, K^*, \mathsf{pad}(P^*,0), T)\}$ such that $X^* \oplus \mathsf{pad}(P^*,0) \oplus \mathsf{pad}(T,1) = \Delta$.*

Below, we will assume that the event of Lemma 5.10 happens. Now, each query $V \leftarrow E(K, \mathsf{pad}(P,0))$ aims at a prior query $\mathsf{pad}(T,1) \leftarrow E^{-1}(K,U)$ with a target $Q^* = \{(\cdot, K^*, \mathsf{pad}(T), X^*), (\cdot, K^*, \mathsf{pad}(P^*,0), T)\}$ such that

$$V[1:\tau] \quad = \quad \mathsf{pad}(P,0)[1:\tau] \oplus T \ , \tag{14}$$

$$\mathsf{pad}(P,0) \oplus (U \oplus \mathsf{pad}(T,1)) \quad = \quad \mathsf{pad}(P^*,0) \oplus X^* \ . \tag{15}$$

We claim that with probability at least $1 - q/2^n$, each query $E(K, \mathsf{pad}(P,0))$ has at most $n/2$ query-target pairs that satisfy Equation (15). Then the chance that the answer V of this query satisfies Equation (14) with one of those $n/2$ pairs is at most $8n/(15 \cdot 2^\tau)$. Summing over all queries $E(K, \mathsf{pad}(P,0))$, the chance that this case happens is at most

$$\frac{8qn}{15 \cdot 2^\tau} + \frac{q}{2^n} \ .$$

Still, we need to add another term $2^{-n} + 2^{-\tau}$ to account for the assumption that the event of Lemma 5.10 happens. Thus the chance that this case happens is actually at most

$$\frac{8qn}{15 \cdot 2^\tau} + \frac{1}{2^\tau} + \frac{q+1}{2^n}$$

To justify the claim above, fix a key K and view each query $E^{-1}(K, U)$ as a round of a ball-throwing game. From Lemma 5.10, for any $\Delta \in \{0,1\}^n$, there are at most $n/2$ prior odd matches $Q^* = \{(\cdot, K^*, \mathsf{pad}(T,1), X^*), (\cdot, K^*, \mathsf{pad}(P^*,0), T)\}$ such that $X^* \oplus \mathsf{pad}(P^*,0) \oplus \mathsf{pad}(T,1) = \Delta \oplus U$. In this round, we will put a ball to bin Δ if the answer of this query hits the $\mathsf{pad}(T,1)$ of one of those matches, which happens with probability at most $8n/(15 \cdot 2^n)$. From Lemma 5.5 with

$c = 8\tau/15$ and $m = n$, for each fixed key K, with probability at least $1 - 2^{-n}$, every bin associated with K has at most $n/2$ balls. Summing over q possible keys, with probability at least $1 - q/2^n$, every bin has at most $n/2$ balls.

Now, for any query $E(K, \mathsf{pad}(P, 0))$, its number of entry-target pairs that satisfy Equation (15) is exactly the number of balls in bin $(K, \mathsf{pad}(P, 0))$. Thus our claim directly follows the balls-into-bins analyses above.

Case 4: Both Q_1 and Q_2 are odd, and Q_2 does not succeed Q_1. We need the following technical lemmas Lemma 5.11 and Lemma 5.12. The proofs are in Appendix D.6 and Appendix D.7 respectively.

Lemma 5.11 *With probability at least $1 - 3q/2^n - 2q/2^\tau$, for every key K and every $\Delta \in \{0,1\}^\tau$, there are at most $n+\tau$ entries $(\cdot, K, \mathsf{pad}(T, 1), X)$ such that there is some entry $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ with $K^* \neq K$ and $(X^* \oplus X)[1 : \tau] = \Delta$.*

Lemma 5.12 *With probability at least $1 - 5q/2^\tau$, for every key K and every $\Delta \in \{0,1\}^\tau$, there are at most 2τ entries $(\cdot, K, \mathsf{pad}(T, 1), X)$ such that there is some other entry $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ with $K^* \neq K$ and $(X^* \oplus X)[1 : \tau] \oplus T = \Delta$.*

Assume that the events of Lemma 5.11 and Lemma 5.12 happen, meaning we need to add a term $3q/2^n + 7q/2^\tau$ into our bound to account for this. We consider the following sub-cases.

Case 4.1: $\mathsf{sign1}(Q_1) = +$ and $\mathsf{sign1}(Q_2) = +$. In this case each query $V \leftarrow E(K, \mathsf{pad}(P, 0))$ aims at a prior query $U \leftarrow E(K^*, \mathsf{pad}(P^*, 0))$ that has prior entries $(\cdot, K, \mathsf{pad}(T, 1), X)$ and $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ such that

$$V[1 : \tau] = \mathsf{pad}(P, 0)[1 : \tau] \oplus T \ , \tag{16}$$

$$\mathsf{pad}(P, 0) = \mathsf{pad}(P^*, 0) \oplus X^* \oplus X \tag{17}$$

$$T = (U \oplus \mathsf{pad}(P^*, 0))[1 : \tau] \ . \tag{18}$$

We claim that with probability at least $1 - q/2^\tau$, for each query $E(K, \mathsf{pad}(P, 0))$, there are at most $\tau/2$ triples that satisfy Equation (17) and Equation (18). Then the chance that the answer V of this query satisfies Equation (16) with one of those $\tau/2$ pairs is at most $8\tau/(15 \cdot 2^\tau)$. Summing over all queries, the chance that this case happens is at most

$$\frac{8q\tau}{15 \cdot 2^\tau} + \frac{q}{2^\tau} \ .$$

To justify the claim above, fix a key K and consider the following balls-into-bins game. For each (adaptive) query $U \leftarrow E(K^*, \mathsf{pad}(P^*, 0))$ and each $\Delta \in \{0,1\}^\tau$, from Lemma 5.11, there are at most $n + \tau$ prior entries $(\cdot, K, \mathsf{pad}(T, 1), X)$ and $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ such that $(X \oplus X^* \mathsf{pad}(P^*, 0))[1 : \tau]) = \Delta$. This query will put a ball to bin (K, Δ) if its $(U \oplus \mathsf{pad}(P^*, 0))[1 : \tau]$ hits the T of one of those $n + \tau$ entries, which happens with probability at most $16(n + \tau)/(15 \cdot 2^\tau)$. Using Lemma 5.5 with $c = 16(n + \tau)/15$ and $m = \tau$, for each fixed key K, with probability at least $1 - 1/2^\tau$, each bin associated with K has at most $\tau/2$ balls. Summing over all q possible keys, with probability at least $1 - q/2^\tau$, each bin has at most $\tau/2$ balls.

Now, for each query $E(K, \mathsf{pad}(P, 0))$, the number of query-match pairs satisfying Equation (17) is exactly the number of balls in bin $(K, \mathsf{pad}(P, 0)[1 : \tau])$. Thus our claim directly follows the balls-into-bins analyses above.

Case 4.2: $\mathsf{sign1}(Q_1) = -$ and $\mathsf{sign1}(Q_2) = +$. In this case each entry $V \leftarrow E(K, \mathsf{pad}(P, 0))$ aims at a prior query $\mathsf{pad}(P^*, 0) \leftarrow E^{-1}(K^*, U)$ that has prior entries $(\cdot, K, \mathsf{pad}(T, 1), X)$ and

$(\cdot, K^*, \mathsf{pad}(T,1), X^*)$ such that

$$V[1:\tau] = \mathsf{pad}(P,0)[1:\tau] \oplus T \ , \tag{19}$$

$$\mathsf{pad}(P,0) = \mathsf{pad}(P^*,0) \oplus X^* \oplus X \ , \tag{20}$$

$$U[1:\tau] = \mathsf{pad}(P^*,0)[1:\tau] \oplus T \ . \tag{21}$$

From Equation (21), we can replace Equation (20) with

$$\mathsf{pad}(P,0)[1:\tau] = (X \oplus X^* \oplus U)[1:\tau] \oplus T \ . \tag{22}$$

We claim that with probability at least $1 - q/2^\tau$, for each query $E(K, \mathsf{pad}(P,0))$, there are at most $\tau/2$ triples that satisfy Equation (22) and Equation (21). Then the chance that the answer V of this query satisfies Equation (16) with one of those $\tau/2$ triples is at most $8\tau/(15 \cdot 2^\tau)$. Summing over all queries, the chance that this case happens is at most

$$\frac{8q\tau}{15 \cdot 2^\tau} + \frac{q}{2^\tau} \ .$$

To justify the claim above, consider the following balls-into-bins game. For each query $E^{-1}(K^*, U)$ and each $\Delta \in \{0,1\}^\tau$, from Lemma 5.12, there are at most 2τ prior entries $(\cdot, K, \mathsf{pad}(T,1), X)$ and $(\cdot, K^*, \mathsf{pad}(T,1), X^*)$ such that

$$(X \oplus X)[1:\tau] \oplus T = \Delta \ .$$

This query will put a ball to bin (K, Δ) if the τ-bit prefix of its answer hits $T \oplus U[1:\tau]$ of one of the 2τ prior entries. In other words, for each query, we will put a ball to a particular bin with probability at most $32\tau/(15 \cdot 2^\tau)$. Using Lemma 5.5 with $c = 32\tau/15$ and $m = \tau$, for each fixed key K, with probability at least $1 - 1/2^n$, each bin associated with K has at most $\tau/2$ balls. Summing over all q possible keys, with probability at least $1 - q/2^\tau$, each bin has at most $\tau/2$ balls.

For each query $E(K, \mathsf{pad}(P,0))$, the number of triples satisfying Equation (22) is exactly the number of balls in bin $(K, \mathsf{pad}(P,0)[1:\tau])$, and thus our claim directly follows the balls-into-bins analyses above.

Case 4.3: $\mathsf{sign1}(Q_1) = +$ and $\mathsf{sign1}(Q_2) = -$. In this case each query $\mathsf{pad}(P,0) \leftarrow E^{-1}(K,V)$ aims at a prior query $U \leftarrow E(K^*, \mathsf{pad}(P^*,0))$ that has prior entries $(\cdot, K, \mathsf{pad}(T,1), X)$ and $(\cdot, K^*, \mathsf{pad}(T,1), X^*)$ such that

$$\mathsf{pad}(P,0)[1:\tau] = V[1:\tau] \oplus T \ , \text{ and} \tag{23}$$

$$\mathsf{pad}(P,0) = \mathsf{pad}(P^*,0) \oplus X^* \oplus X, \tag{24}$$

$$T = (U \oplus \mathsf{pad}(P^*,0))[1:\tau] \ . \tag{25}$$

From Equation (23), we can replace Equation (24) with

$$V[1:\tau] = (\mathsf{pad}(P^*,0) \oplus X^* \oplus X)[1:\tau] \oplus T \ . \tag{26}$$

We claim that with probability at least $1 - q/2^\tau$, each query $E^{-1}(K,V)$ has at most $\tau/2$ triples that satisfy Equation (25) and Equation (26). Then the answer $\mathsf{pad}(P,0)$ of this query satisfies Equation (23) with one of these triples is at most $8\tau/(15 \cdot 2^\tau)$. Summing over all queries, the chance that this case happens is at most

$$\frac{8q\tau}{15 \cdot 2^\tau} + \frac{q}{2^\tau} \ .$$

We now justify the claim above. Fix a key K and consider the following balls-into-bins game. For each (adaptive) query $U \leftarrow E(K^*, \mathsf{pad}(P^*,0))$ and each $\Delta \in \{0,1\}^\tau$, from Lemma 5.12, there are at most 2τ prior log entries $(\cdot, K, \mathsf{pad}(T,1), X)$ and $(\cdot, K^*, \mathsf{pad}(T,1), X^*)$ such that $(X \oplus X^*)[1:\tau] \oplus T = \mathsf{pad}(P^*,0)[1:\tau] \oplus \Delta$. This query will put a ball to bin (K, Δ) if its $(U \oplus \mathsf{pad}(P^*,0))[1:\tau]$

hits the T of one of those 2τ entries, which happens with probability at most $32\tau/(15 \cdot 2^\tau)$. Using Lemma 5.5 with $c = 32\tau/(15 \cdot 2^\tau)$ and $m = \tau$, with probability at least $1 - 1/2^\tau$, each bin associated with key K has at most $\tau/2$ balls. Summing over all possible keys, with probability at least $1 - q/2^\tau$, every bin has most $\tau/2$ balls.

For each query $E^{-1}(K, V)$, the number of triples satisfying Equation (25) and Equation (26) is exactly the number of balls in bin $(K, V[1:\tau])$, and thus our claim directly follows the balls-into-bins analyses above.

Case 4.4: $\mathsf{sign1}(Q_1) = -$ and $\mathsf{sign1}(Q_2) = -$. In this case each query $\mathsf{pad}(P, 0) \leftarrow E^{-1}(K, V)$ aims at a prior query $\mathsf{pad}(P^*, 0) \leftarrow E^{-1}(K^*, U)$ that has prior entries $(\cdot, K, \mathsf{pad}(T, 1), X)$ and $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ such that

$$\mathsf{pad}(P, 0)[1:\tau] = V[1:\tau] \oplus T \ , \text{ and} \tag{27}$$

$$\mathsf{pad}(P, 0) = \mathsf{pad}(P^*, 0) \oplus X^* \oplus X, \tag{28}$$

$$\mathsf{pad}(P^*, 0)[1:\tau] = T \oplus U[1:\tau] \ . \tag{29}$$

We then obtain

$$V[1:\tau] = (U \oplus X^* \oplus X)[1:\tau] \ . \tag{30}$$

We claim that with probability at least $1 - q/2^\tau$, each query $E^{-1}(K, V)$ has at most $\tau/2$ triples that satisfy Equation (29) and Equation (30). Then the answer $\mathsf{pad}(P, 0)$ of this query satisfies Equation (27) with one of these triples is at most $8\tau/(15 \cdot 2^\tau)$. Summing over all queries, the chance that this case happens is at most

$$\frac{8q\tau}{15 \cdot 2^\tau} + \frac{q}{2^\tau} \ .$$

We now justify the claim above. Fix a key K and consider the following balls-into-bins game. For each (adaptive) query $\mathsf{pad}(P^*, 0) \leftarrow E^{-1}(K^*, U)$ and each $\Delta \in \{0,1\}^\tau$, from Lemma 5.11, there are at most $n + \tau$ prior log entries $(\cdot, K, \mathsf{pad}(T, 1), X)$ and $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ such that $(X \oplus X^*)[1:\tau] = U \oplus \Delta$. This query will put a ball to bin (K, Δ) if its $(U \oplus \mathsf{pad}(P^*, 0))[1:\tau]$ hits the T of one of those $n + \tau$ entries, which happens with probability at most $16(n + \tau)/(15 \cdot 2^\tau)$. Using Lemma 5.5 with $c = 16(n + \tau)/(15 \cdot 2^\tau)$ and $m = \tau$, with probability at least $1 - 1/2^\tau$, each bin associated with key K has at most $\tau/2$ balls. Summing over all possible keys, with probability at least $1 - q/2^\tau$, every bin has most $\tau/2$ balls.

For each query $E^{-1}(K, V)$, the number of triples satisfying Equation (29) and Equation (30) is exactly the number of balls in bin $(K, V[1:\tau])$, and thus our claim directly follows the balls-into-bins analyses above.

6 Succinctly-Committing AE

Suppose that we have a tag-based symmetric encryption scheme SE of s-bit committing security but the tag is long, say SE is CTY. Recall that our generic attack shows that the tag length $\mathsf{SE.tl}$ alone must be at least $2s$ bis. For simplicity, assume that SE has constant expansion. Let ℓ be the length of the ciphertext core of the empty message under SE. For a standard AE1 scheme, ℓ is usually 0. For an AE2 (nonce-hiding) scheme, ℓ is often the nonce length. Note that for a ciphertext core C^* of a message M under SE, we have $|C^*| = |M| + \ell$, meaning that SE has expansion $\ell + \mathsf{SE.tl} \geq \ell + 2s$. We now give a Shorten Ciphertext (SC) transform that turns SE into a succinctly-committing symmetric encryption scheme of $(s + \ell)$-bit (eventual) expansion and (nearly) s bits of committing security.

$\overline{\mathsf{SE}}.\mathsf{Enc}(K, N, A, M)$	$\overline{\mathsf{SE}}.\mathsf{Dec}(K, D, A, C^*\|R)$
$S\|P \leftarrow M$ // $\|P\| = \min\{\|M\|, m\}$	$(S, N, T^*) \leftarrow \mathsf{SE}.\mathsf{Tag}(K, D, A, C^*)$
$C^*\|T \leftarrow \mathsf{SE}.\mathsf{Enc}(K, N, A, S)$	$P \leftarrow \mathsf{F}^{-1}(T^*, R)$
$R \leftarrow \mathsf{F}(T, P)$	If $(P = \perp)$ then return $\perp$
Return $C^*\|R$	If $(\|C^*\| \neq \ell) \wedge (\|P\| \neq m)$ then
	Return $\perp$
	Return $(S\|P, N)$

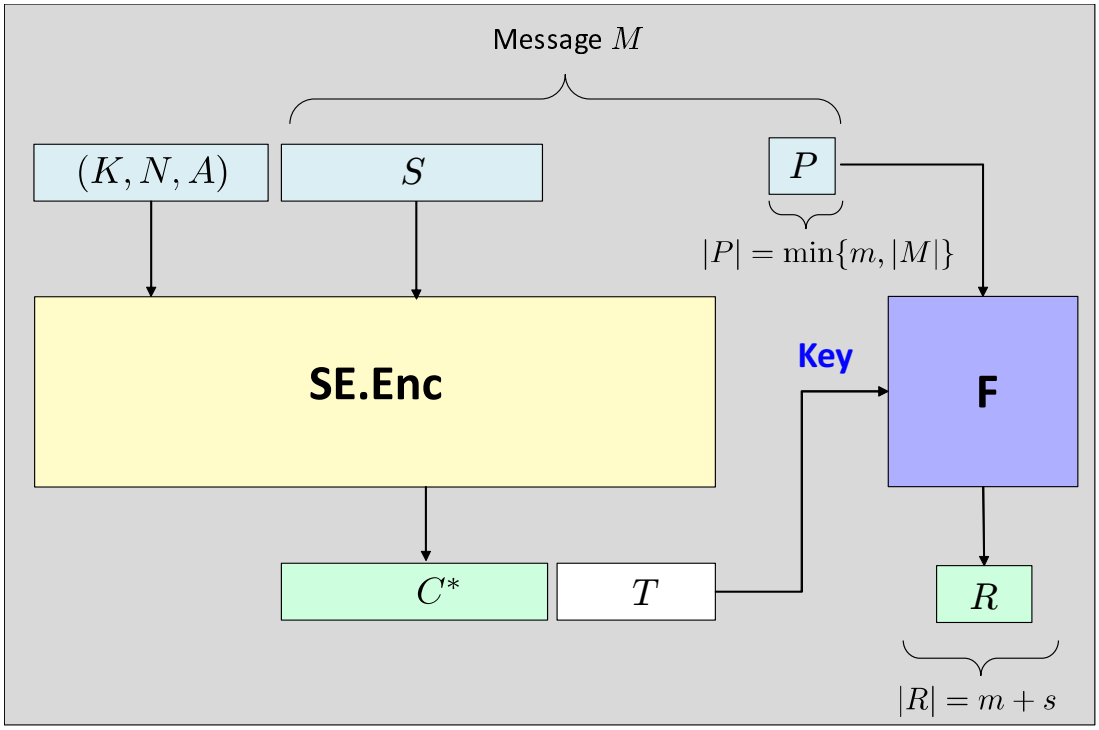

Figure 9: The $\mathsf{SC}[\mathsf{SE}, \mathsf{F}]$ transform.

THE SC TRANSFORM. Let SE be a tag-based committing symmetric encryption scheme of λ-bit tag, and assume that it has constant expansion. Let F be an IPF of message space $\{0,1\}^{\leq m}$, key length λ, and ciphertext length $m + s$. (Note that the key length of F is also the tag length of SE.) Our scheme will have eventual pansion $\ell + s$ after the cutoff m.

The scheme $\overline{\mathsf{SE}} = \mathsf{SC}[\mathsf{SE}, \mathsf{F}]$ is described in Fig. 9. It has the same message space, AD space, nonce space, and decryption-nonce derivation function as SE. It is however *not* a tag-based scheme.

In particular, to encrypt a message M, we first parse it as $S\|P$, where $|P| = \min\{m, |M|\}$. (This means if $|M| \leq m$ then S is the empty string.) We then use SE to encrypt just S to derive $C^*\|T$. Finally, we use F to encrypt P with key T to produce a tag R, and output $C^*\|R$. Note that if $|M| < m$ then $|C| = m + \ell + s$; otherwise if $M| \geq m$ then

$$|C| = (|S| + \ell) + m + s = |M| + \ell + s \ .$$

That is, the eventual expansion of SC is $\ell + s$ bits after the cutoff m.

For decryption, given $(K, D, A, C^*\|R)$, we first use $\mathsf{SE}.\mathsf{Tag}$ on (K, D, A, C^*) to recover the suffix S, the nonce N, and the tentative tag T^*. We then run $\mathsf{F}^{-1}(T^*, R)$ to get the prefix P. However, there's a subtlety here. If $|C^*| \neq \ell$ then P must be exactly m bits, so we need to reject the ciphertext even if $P \neq \perp$ but $|P| \neq m$.

DISCUSSION. If we instantiate F via the $\mathsf{HtM}[E,\tau]$ construction in Section 5 (with $d = 8$) then $m = 120$ and $s = \tau + 8$. That is, for messages at least 120 bits, SC has $\tau + 8$ bits of expansion, and $\tau - \lceil \log_2(\tau) \rceil$ bits of committing security. If we also instantiate SE via CTY then $\lambda = 256$, meaning that the key length of HtM must be 256-bit. Thus the cost of HtM is two sequential AES-256 calls.

COMMITTING SECURITY OF SC. Proposition 6.1 below confirms that SC tightly preserves committing security. Intuitively, thanks to the collision-resistance of F, the outer tag R is a commitment of the plaintext P and the inner tag T, and thus $C^* \| R$ is a commitment of $(P, C^* \| T)$, reducing the committing security of SC to that of SE. The proof is in Appendix H.

Proposition 6.1 *Let SE be a tag-based symmetric encryption scheme and F be an IPF as above. Let $\overline{\mathsf{SE}} = \mathsf{SC}[\mathsf{SE}, \mathsf{F}]$. Then given an adversary $\mathcal{A}$, we can construct $\mathcal{B}$ and $\mathcal{D}$ of about the same resource such that*

$$\mathbf{Adv}^{\mathrm{cmt}}_{\overline{\mathsf{SE}}}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{cmt}}_{\mathsf{SE}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{coll}}_{\mathsf{F}}(\mathcal{D}) \ .$$

Adversary $\mathcal{B}$ has the same running time as $\mathcal{A}$. The running time of $\mathcal{D}$ is that of $\mathcal{A}$ plus the time to run SE on $\mathcal{A}$'s output.

Corollary 6.2 bellow gives a concrete bound on the committing security of $\overline{\mathsf{SE}} = \mathsf{SC}[\mathsf{SE}, \mathsf{F}]$ if we instantiate F by the HtM construction in Section 5. The bound is obtained by substituting the term $\mathbf{Adv}^{\mathrm{coll}}_{\mathsf{F}}(\mathcal{D})$ in Proposition 6.1 with the bound in Proposition 5.1.

Corollary 6.2 *Let E be a blockcipher on $\{0,1\}^n$ that we model as an ideal cipher. Let SE be a tag-based symmetric encryption scheme, and let $\mathsf{F} = \mathsf{HtM}[E,\tau]$ be the IPF in Section 5. Let $\overline{\mathsf{SE}} = \mathsf{SC}[\mathsf{SE}, \mathsf{F}]$. Then given an adversary $\mathcal{A}$ of q ideal-cipher queries, we can construct $\mathcal{B}$ of about the same running time such that*

$$\mathbf{Adv}^{\mathrm{cmt}}_{\overline{\mathsf{SE}}}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{cmt}}_{\mathsf{SE}}(\mathcal{B}) + \frac{4(n+\tau)q + 5}{2^\tau} \ .$$

Our work focuses on the CMT-4 notion of Bellare and Hoang [5] that commits (K, N, A, M). If one only needs CMT-1 security (that commits just the key K), we note that the SC transform also preserves CMT-1 security.

AE SECURITY OF SC. Unfortunately SC doesn't *generically* preserve AE security. In particular, using SC requires that $\mathsf{SE.Tag}$ generates pseudorandom tags on ciphertext cores chosen by the adversary, because those tags will be used as the keys for F. The AE security of SE however only guarantees that such tags are unpredictable. Still, SC *does* preserve AE security if the base AE scheme is built on top of CTY, as confirmed via Theorem 6.3 below.

The term $u^2/2^k$ in the bound of Theorem 6.3 is *inherent* in (multi-user) AE security of *any* symmetric encryption scheme if there is no additional assumption on how nonces are implemented. Indeed, it corresponds to a generic key-collision attack; see, for example, [14, Section 3.1] for details. Likewise, if the adversary makes $\Omega(p)$ computations, the term $up/2^k$ is also inherent. This corresponds to a generic key-recovery attack; see [31, Appendix B.2] for details. (We note that these two attacks were actually given for AE1 schemes, but they can be easily translated to symmetric encryption schemes.)

Theorem 6.3 *Let F be an IPF with plaintext space $\{0,1\}^{\le m}$ and ciphertext length $m + s$. Let SE be a symmetric encryption scheme of k-bit key. Let H be a hash function that we will model as a random oracle. Let $\overline{\mathsf{SE}} = \mathsf{SC}[\mathsf{CTY}[H, \mathsf{SE}], \mathsf{F}]$. Then for an adversary $\mathcal{A}$ that attacks u users with at*

most p random-oracle queries, q encryption queries, and q_v verification queries, we can construct adversaries $\mathcal{B}$ and $\mathcal{D}$ such that

$$\mathbf{Adv}^{\mathrm{ae}}_{\overline{\mathsf{SE}}}(\mathcal{A}) \;\;\leq\;\; \mathbf{Adv}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{ae}}_{\mathsf{SE}}(\mathcal{D}) + \frac{q^2}{2^{m+s}} + \frac{u(2u+3p)}{2^k} + \frac{2q_v}{2^s} \;\;.$$

Compared to the running time of $\mathcal{A}$, the overhead of $\mathcal{B}$ is the time to use $\overline{\mathsf{SE}}$ to encrypt/decrypt $\mathcal{A}$'s queries. It has the same number of verification queries as $\mathcal{A}$, with a single encryption query per user. The overhead of $\mathcal{D}$ is the time to use SE for p times on k-bit data. Its query statistics is that of $\mathcal{A}$ plus an encryption query per user for k-bit data.

Corollary 6.4 below gives a concrete bound on the AE security of the composition of SC and CTY, if we instantiate F via the HtM construction in Section 5. The proof is a straightforward combination of Theorem 6.3 and Proposition 5.2.

Corollary 6.4 *Let E be a blockcipher on $\{0,1\}^n$, and let F be $\mathsf{HtM}[E,\tau]$. Let SE be a symmetric encryption scheme of k-bit key. Let H be a hash function that we will model as a random oracle. Let $\overline{\mathsf{SE}} = \mathsf{SC}[\mathsf{CTY}[H,\mathsf{SE}],\mathsf{F}]$. Then for an adversary $\mathcal{A}$ that attacks u users with at most p random-oracle queries, q encryption queries, and q_v verification queries, we can construct $\mathcal{B}$ and $\mathcal{D}$ such that*

$$\mathbf{Adv}^{\mathrm{ae}}_{\overline{\mathsf{SE}}}(\mathcal{A}) \;\;\leq\;\; \mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{ae}}_{\mathsf{SE}}(\mathcal{D}) + \frac{q^2}{2^{n+\tau}} + \frac{u(2u+3p)}{2^k} + \frac{2q_v}{2^\tau} \;\;.$$

Compared to the running time of $\mathcal{A}$, the overhead of $\mathcal{B}$ is the time to use $\overline{\mathsf{SE}}$ to encrypt/decrypt $\mathcal{A}$'s queries. It has twice the number of queries as $\mathcal{A}$. The overhead of $\mathcal{D}$ is the time to use SE for p times on k-bit data. Its query statistics is that of $\mathcal{A}$ plus an encryption query (whose message is just k-bit) per user.

Discussion. From Theorem 6.3 and Proposition 6.1, the composition of SC and CTY is a generic transform that turns a standard tag-based symmetric encryption scheme like GCM to a succinctly committing one. The overhead consists of (i) the hashing cost of the AD, which is necessary for committing security, and (ii) the cost of an IPF, which is very cheap. (Recall that if we instantiate the IPF via HtM then the cost is just two sequential AES-256 calls.)[3] This hash function needs to be modeled as a random oracle, and thus can be instantiated via SHA-3 or (truncated) SHA-512. On small data, the hashing overhead is expensive for small data, but unfortunately unavoidable if we want to commit the AD. Moreover, this transform allows one to have strong committing security with smaller expansion, say 80-bit committing security for 96-bit expansion. In contrast, none of the existing schemes can offer meaningful committing security when the expansion gets below 128 bits.

On the other hand, there is a potential timing leakage in SC because there are two checks $(P \neq \perp)$ and $(|C^*| \neq \ell) \wedge (|P| \neq m)$ in the decryption of SC. However, if one checks $(P \neq \perp)$ first, as written in the code in Fig. 9, then it's unlikely that an invalid ciphertext can pass the test $(P \neq \perp)$, and thus one can avoid the timing issue. The proof in Theorem 6.3 rigorously confirms this intuition, showing that the composition of SC and CTY has good AE security even in the presence of timing leakage. In particular, in the games, verification queries only consider the test $(P \neq \perp)$. That is, if an adversary can launch a verification query that passes $(P \neq \perp)$ but fails the check $(|C^*| \neq \ell) \wedge (|P| \neq m)$, it will get a true answer (instead of false) and win the game.

[3] There is also an AES key-setup cost since we have to rekey HtM for every encryption, but a good implementation can hide this latency. For example, AES-GCM-SIV [28] also derives new subkeys for every encryption, but manages to hide the key-setup cost of AES.

7 Acknowledgments

Many thanks to Cong Wu who implemented our transforms and provided extensive benchmarks. We thank the CRYPTO 2024 reviewers for their careful reading and valuable comments.

References

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

[2] 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.

[3] M. Barbosa and P. Farshim. Indifferentiable authenticated encryption. In H. Shacham and A. Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 187–220. Springer, Heidelberg, Aug. 2018.

[4] M. Bellare and V. T. Hoang. Identity-based format-preserving encryption. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 2017*, pages 1515–1532. ACM Press, Oct. / Nov. 2017.

[5] 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.

[6] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 531–545. Springer, Heidelberg, Dec. 2000.

[7] M. Bellare, R. Ng, and B. Tackmann. Nonces are noticed: AEAD revisited. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 235–265. Springer, Heidelberg, Aug. 2019.

[8] 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.

[9] 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.

[10] M. Bellare, P. Rogaway, and D. Wagner. The EAX mode of operation. In B. K. Roy and W. Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 389–407. Springer, Heidelberg, Feb. 2004.

[11] M. Bellare and L. Shea. Flexible password-based encryption: Securing cloud storage and provably resisting partitioning-oracle attacks. In M. Rosulek, editor, *CT-RSA 2023*, volume 13871 of *LNCS*, pages 594–621. Springer, Heidelberg, Apr. 2023.

[12] M. Bellare and B. Tackmann. The multi-user security of authenticated encryption: AES-GCM in TLS 1.3. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 247–276. Springer, Heidelberg, Aug. 2016.

[13] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In A. Miri and S. Vaudenay, editors, *SAC 2011*, volume 7118 of *LNCS*, pages 320–337. Springer, Heidelberg, Aug. 2012.

[14] P. Bose, V. T. Hoang, and S. Tessaro. Revisiting AES-GCM-SIV: Multi-user security, faster key derivation, and better bounds. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 468–499. Springer, Heidelberg, Apr. / May 2018.

[15] J. Chan and P. Rogaway. Anonymous AE. In S. D. Galbraith and S. Moriai, editors, *ASIACRYPT 2019, Part II*, volume 11922 of *LNCS*, pages 183–208. Springer, Heidelberg, Dec. 2019.

[16] 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.

[17] 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 and more. *IACR Transactions on Symmetric Cryptology*, 2023(4):452–488, Dec. 2023.

[18] P. Crowley and E. Biggers. Adiantum: length-preserving encryption for entry-level processors. *IACR Trans. Symm. Cryptol.*, 2018(4):39–61, 2018.

[19] P. Crowley, N. Huckleberry, and E. Biggers. Length-preserving encryption with HCTR2. Technical report, Cryptology ePrint Archive, Report 2021/11441, 2021. http://eprint. iacr. org, 2021.

[20] J. Daemen and V. Rijmen. AES proposal: Rijndael. NIST AES proposal, 1998.

[21] Y. Dai and J. P. Steinberger. Indifferentiability of 8-round Feistel networks. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 95–120. Springer, Heidelberg, Aug. 2016.

[22] J. P. Degabriele, M. Fischlin, and J. Govinden. The indifferentiability of the duplex and its practical applications. In J. Guo and R. Steinfeld, editors, *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part VIII*, volume 14445 of *Lecture Notes in Computer Science*, pages 237–269. Springer, 2023.

[23] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer. Ascon v1.2: Lightweight authenticated encryption and hashing. *Journal of Cryptology*, 34(3):33, July 2021.

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

[25] P. Farshim, C. Orlandi, and R. Roşie. Security of symmetric primitives under incorrect usage of keys. *IACR Trans. Symm. Cryptol.*, 2017(1):449–473, 2017.

[26] Y. Gertner and A. Herzberg. Committing encryption and publicly-verifiable signcryption. Cryptology ePrint Archive, Report 2003/254, 2003. https://eprint.iacr.org/2003/254.

[27] P. Grubbs, J. Lu, and T. Ristenpart. Message franking via committing authenticated encryption. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 66–97. Springer, Heidelberg, Aug. 2017.

[28] S. Gueron. AES-GCM-SIV. https://github.com/Shay-Gueron/AES-GCM-SIV, Jan. 2018.

[29] S. Gueron and N. Mouha. Simpira v2: A family of efficient permutations using the AES round function. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 95–125. Springer, Heidelberg, Dec. 2016.

[30] 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.

[31] V. T. Hoang and Y. Shen. Security of streaming encryption in google's tink library. In J. Ligatti, X. Ou, J. Katz, and G. Vigna, editors, *ACM CCS 2020*, pages 243–262. ACM Press, Nov. 2020.

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

[33] 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.

[34] U. M. Maurer, R. Renner, and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In M. Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer, Heidelberg, Feb. 2004.

[35] D. A. McGrew and J. Viega. The security and performance of the Galois/counter mode (GCM) of operation. In A. Canteaut and K. Viswanathan, editors, *INDOCRYPT 2004*, volume 3348 of *LNCS*, pages 343–355. Springer, Heidelberg, Dec. 2004.

[36] S. Menda, J. Len, P. Grubbs, and T. Ristenpart. Context discovery and commitment attacks - how to break CCM, EAX, SIV, and more. In C. Hazay and M. Stam, editors, *EUROCRYPT 2023, Part IV*, volume 14007 of *LNCS*, pages 379–407. Springer, Heidelberg, Apr. 2023.

[37] Y. Naito, Y. Sasaki, and T. Sugawara. KIVR: Committing authenticated encryption using redundancy and application to GCM, CCM, and more. In *22nd International Conference on Applied Cryptography and Network Security (ACNS 2024)*, pages 318–347, 2024.

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

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

[40] P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In M. K. Reiter and P. Samarati, editors, *ACM CCS 2001*, pages 196–205. ACM Press, Nov. 2001.

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

[42] J. Salowey, A. Choudhury, and D. McGrew. AES Galois Counter Mode (GCM) Cipher Suites for TLS. RFC 5288, Aug. 2008. `https://datatracker.ietf.org/doc/html/rfc5288`.

[43] D. Whiting, R. Housely, and N. Ferguson. Counter with CBC-MAC (CCM). IETF Network Working Group, RFC 3610, September 2003.

A Proof of Lemma 3.2

Adversary $\mathcal{B}$ runs $\mathcal{A}$, and lets $\mathcal{A}$ use its NEW and ENC oracles. For each verification query of $\mathcal{A}$, however, $\mathcal{B}$ stores that in a list L, and simply returns false to $\mathcal{A}$. When $\mathcal{A}$ terminates with its guess b', for each query (i, D, A, C) in L, if there is no prior encryption query $C \leftarrow \textsc{Enc}(i, N, A, M)$ such that $\mathsf{SE.df}(N) = D$ then $\mathcal{B}$ will query $\textsc{Vf}(i, D, A, C)$, otherwise it will terminate and return 1. If one of those verification queries results in a true answer then $\mathcal{B}$ will return 1, otherwise it returns b'.

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

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

Let Bad be the event that in the ideal world of $\mathcal{B}$, there is a verification query $\textsc{Vf}(i, D, A, C)$ and then later there is an encryption query $C \leftarrow \textsc{Enc}(i, N, A, M)$ such that $\mathsf{SE.df}(N) = D$. If Bad doesn't happen then $\mathcal{B}$ correctly simulates game $\mathbf{G}^{\mathrm{rand}}_{\mathsf{SE}}(\mathcal{A})$ and has the same answer as $\mathcal{A}$, and thus

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

To bound Pr[Bad], consider a verification query $\textsc{Vf}(i, D, A, C)$. There are at most $2^{\mathsf{SE.dnd}}$ choices for the nonces N, and the message can be anything in the set $\{0,1\}^{\leq |C|-\tau-\mathsf{SE.dnd}}$. Hence this verification query can be targeted by at most

$$\sum_{j=0}^{|C|-\tau-\mathsf{SE.dnd}} 2^j \cdot 2^{\mathsf{SE.dnd}} < 2^{|C|-\tau+1}$$

encryption queries. However, the chance that one of those encryption queries can result in C is at most

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

Summing this over q_v verification queries,

$$\Pr[\mathrm{Bad}] \leq \frac{2q_v}{2^\tau} \ .$$

Hence

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

and thus

$$\mathbf{Adv}^{\mathrm{ae}}_{\mathsf{SE}}(\mathcal{B}) \geq \mathbf{Adv}^{\mathrm{ae}}_{\mathsf{SE}}(\mathcal{A}) - \frac{2q_v}{2^\tau} \ .$$

If $\mathsf{SE.ce}(m) = \mathsf{SE.dnd} + \tau$ for every m then for each verification query $\textsc{Vf}(i, D, A, C)$, the set of messages can be restricted to $\{0,1\}^{|C|-\tau-\mathsf{SE.dnd}}$ (instead of $\{0,1\}^{\leq |C|-\tau-\mathsf{SE.dnd}}$), and thus the term

$2q_v/2^\tau$ can be improved to $q_v/2^\tau$.

B Equivalence of Committing Definitions

$\triangleright$ CMTD $\longrightarrow$ CMT: First we show that CMTD implies CMT. Let SE be a symmetric encryption scheme. 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) and (K_2, N_2, A_2, M_2). It then runs $C \leftarrow \mathsf{SE.Enc}(K_1, N_1, A_1, M_1)$, and outputs $(C, (K_1, \mathsf{SE.df}(N_1), A_1), (K_2, \mathsf{SE.df}(N_2), A_2))$.

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

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

$\triangleright$ CMT $\dashrightarrow$ CMTD: Conversely, we show that for tidy schemes, CMT implies CMTD. Let SE be a tidy symmetric encryption scheme. 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 $(C, (K_1, D_1, A_1), (K_2, D_2, A_2))$, and gets $(M_i, N_i) \leftarrow \mathsf{SE.Dec}(K_i, D_i, A_i, C)$ for each $i \in \{1, 2\}$. It then outputs $((K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2))$.

For analysis, without loss of generality, assume that $\mathcal{A}_d$ outputs distinct tuples (K_1, D_1, A_1), (K_2, D_2, A_2). Suppose that $\mathcal{A}_d$ wins its game, meaning that $(M_i, N_i) \neq \perp$. Since SE is tidy, we have $\mathsf{SE.Enc}(K_i, N_i, A_i, M_i) = C$ for each $i \in \{1, 2\}$ and $\mathsf{SE.df}(N_i) = D_i$, and thus $(K_1, N_1, A_1) \neq (K_2, N_2, A_2)$. Hence $\mathcal{A}_e$ also wins its game, and therefore,

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

C Proof of Theorem 3.4

From Lemma 3.2, without loss of generality, assume that $\mathcal{A}$ is orderly. This assumption creates a difference of at most $2q_v/2^\lambda$, which we will account later. Assume that the adversary doesn't make forbidden queries. Consider games G_1–G_7 in Fig. 10–Fig. 12. Game G_1 coincides to game $\mathbf{G}^{\text{real}}_{\overline{\mathsf{SE}}}(\mathcal{A})$. For clarity, $\mathcal{A}$ is given an interface $\textsc{Ro}$ to the random oracle, and $\overline{\mathsf{SE}}.\mathsf{Enc}$ and $\overline{\mathsf{SE}}.\mathsf{Dec}$ are implemented using another interface $\overline{\textsc{Ro}}$. Game G_7 corresponds to game $\mathbf{G}^{\text{rand}}_{\overline{\mathsf{SE}}}(\mathcal{A})$. Hence

$$\mathbf{Adv}^{\text{ae}}_{\overline{\mathsf{SE}}}(\mathcal{A}) = \Pr[G_1(\mathcal{A})] - \Pr[G_7(\mathcal{A})] \ .$$

We now describe the game chain. In game G_2, we maintain a set Dom that is initialized to $\emptyset$. This set keeps track of the keys $K_1, \ldots, K_v$ so far, and also the components K in random-oracle queries $\textsc{Ro}(K, N, V)$ of $\mathcal{A}$. Thus $|\mathsf{Dom}| \leq p + u$. In game G_2, each time when we initialize a user i and sample its key, if the key falls within Dom then we set bad to true, and re-sample the key uniformly at random from $\{0,1\}^k \backslash \mathsf{Dom}$. Games G_1 and G_2 are identical until bad, and thus from the Fundamental Lemma of Game Playing [9],

$$\Pr[G_1(\mathcal{A})] - \Pr[G_2(\mathcal{A})] \leq \Pr[G_2(\mathcal{A}) \text{ sets } \mathsf{bad}] \leq \frac{u(u+p)}{2^k} \ .$$

Game G_3 is a different way to implement G_2. Instead of using the same underlying table Tbl for both $\textsc{Ro}$ and $\overline{\textsc{Ro}}$, we use two tables Tbl and $\overline{\mathsf{Tbl}}$, where calls to $\overline{\textsc{Ro}}$ are implemented in $\overline{\mathsf{Tbl}}$. For

Games $G_1(\mathcal{A})$, $G_2(\mathcal{A})$

$v \leftarrow 0$; $\mathsf{Dom} \leftarrow \emptyset$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Enc},\text{Vf},\text{Ro}}$
Return $(b' = 1)$

New()
$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k$
If $K_v \in \mathsf{Dom}$ then
  $\mathsf{bad} \leftarrow \mathsf{true}$; $K_v \leftarrow\!\!\$\ \{0,1\}^k \backslash \mathsf{Dom}$
$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{K_v\}$

Enc(i, N, A, M)
$C^* \| R \leftarrow \mathsf{SE.Enc}(K_i, N, \varepsilon, M)$
$T \leftarrow \overline{\text{Ro}}(K_i, N, A \| R)$
Return $C^* \| T$

Vf$(i, D, A, C^* \| T)$
$(M, N, R) \leftarrow \mathsf{SE.Tag}(K_i, D, \varepsilon, C^*)$
$T^* \leftarrow \overline{\text{Ro}}(K_i, N, A \| R)$
Return $(T = T^*)$

Ro(K, N, V)
If $\mathsf{Tbl}[K, N, V] = \bot$ then
  $\mathsf{Tbl}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{K\}$
Return $\mathsf{Tbl}[K, N, V]$

$\overline{\text{Ro}}(K, N, V)$
If $\mathsf{Tbl}[K, N, V] = \bot$ then
  $\mathsf{Tbl}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
Return $\mathsf{Tbl}[K, N, V]$

Games $G_3(\mathcal{A})$, $G_4(\mathcal{A})$

$v \leftarrow 0$; $\mathsf{win} \leftarrow \mathsf{false}$
$\mathsf{Dom} \leftarrow \emptyset$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Enc},\text{Vf},\text{Ro}}$
$\mathsf{win} \leftarrow \mathsf{bad}$; $\mathsf{win} \leftarrow \mathsf{false}$
Return $(b' = 1) \vee \mathsf{win}$

New()
$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k \backslash \mathsf{Dom}$
$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{K_v\}$

Enc(i, N, A, M)
$C^* \| R \leftarrow \mathsf{SE.Enc}(K_i, N, \varepsilon, M)$
$T \leftarrow \overline{\text{Ro}}(K_i, N, A \| R)$
Return $C^* \| T$

Vf$(i, D, A, C^* \| T)$
$(M, N, R) \leftarrow \mathsf{SE.Tag}(K_i, D, \varepsilon, C^*)$
$T^* \leftarrow \overline{\text{Ro}}(K_i, N, A \| R)$
Return $(T^* = T)$

Ro(K, N, V)
If $K \in \{K_1, \ldots, K_v\}$ then
  $\mathsf{bad} \leftarrow \mathsf{true}$; return $\overline{\text{Ro}}(K, N, V)$
If $\mathsf{Tbl}[K, N, V] = \bot$ then
  $\mathsf{Tbl}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{K\}$
Return $\mathsf{Tbl}[K, N, V]$

$\overline{\text{Ro}}(K, N, V)$
If $\overline{\mathsf{Tbl}}[K, N, V] = \bot$ then
  $\overline{\mathsf{Tbl}}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
Return $\overline{\mathsf{Tbl}}[K, N, V]$

Figure 10: Games G_1–G_4 in the proof of Theorem 3.4. Games G_2 and G_3 contain the corresponding boxed statements, but games G_1 and G_4 do not.

calls (K, N, V) to Ro, generally they are implemented via Tbl, but if K somehow falls within the current set of keys $\{K_1, \ldots, K_v\}$ then we implement it by calling $\overline{\text{Ro}}(K, N, V)$. Hence

$$\Pr[G_2(\mathcal{A})] = \Pr[G_3(\mathcal{A})] \ .$$

In game G_4, for each call $\text{Ro}(K, N, V)$, if $K \in \{K_1, \ldots, K_v\}$ then we implement it via Tbl (meaning Ro and $\overline{\text{Ro}}$ are now independent) and set bad to true. The game will return true if bad is set. For $i \in \{3, 4\}$ and $b \in \{\mathsf{true}, \mathsf{false}\}$, let $G_i(\mathcal{A}, b)$ denote the event that $G_i(\mathcal{A})$ returns true and $\mathsf{bad} = b$. Note that G_4 and G_3 are identical until bad is set. Then

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

Games $\boxed{G_5(\mathcal{A})}$, $G_6(\mathcal{A})$

$v \leftarrow 0$; $\mathsf{win} \leftarrow \mathsf{false}$; $\mathsf{Dom} \leftarrow \emptyset$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Vf},\textsc{Ro}}$
Return $(b' = 1) \vee \mathsf{win}$

$\textsc{New}()$

$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k$; $\boxed{K_v \leftarrow\!\!\$\ \{0,1\}^k \backslash \mathsf{Dom}}$; $\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{K_v\}$

$\textsc{Enc}(i, N, A, M)$

$C^* \| R \leftarrow \mathsf{SE.Enc}(K_i, N, \varepsilon, M)$; $T \leftarrow \overline{\textsc{Ro}}(K_i, N, A \| R)$
$\mathsf{Map}[i, N, A \| R, T] \leftarrow \mathsf{true}$; Return $C^* \| T$

$\textsc{Vf}(i, D, A, C^* \| T)$

For (N, R) with $\mathsf{SE.df}(N) = D$ do
  If $(\mathsf{Map}[i, N, A \| R, T] \neq \bot)$ then
    Return $(\mathsf{SE.Dec}(K_i, D, \varepsilon, C^* \| R) \neq \bot)$
Return false

$\textsc{Ro}(K, N, V)$

If $K \in \{K_1, \ldots, K_v\}$ then $\mathsf{win} \leftarrow \mathsf{true}$
If $\mathsf{Tbl}[K, N, V] = \bot$ then $\mathsf{Tbl}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
Return $\mathsf{Tbl}[K, N, V]$

$\overline{\textsc{Ro}}(K, N, V)$

If $\overline{\mathsf{Tbl}}[K, N, V] = \bot$ then $\overline{\mathsf{Tbl}}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
Return $\overline{\mathsf{Tbl}}[K, N, V]$

Figure 11: Games G_5 and G_6 in the proof of Theorem 3.4. Game G_5 contains the corresponding boxed statements, but game G_6 does not.

Game $G_7(\mathcal{A})$

$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Vf},\textsc{Ro}}$
Return $(b' = 1)$

$\textsc{New}()$

$v \leftarrow v + 1$

$\textsc{Enc}(i, N, A, M)$

$C^* \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.cl}(|M|) - \mathsf{SE.tl}}$; $T \leftarrow\!\!\$\ \{0,1\}^\lambda$
Return $C^* \| T$

$\textsc{Vf}(i, D, A, C^* \| T)$

Return false

$\textsc{Ro}(K, N, V)$

If $\mathsf{Tbl}[K, N, V] = \bot$ then
  $\mathsf{Tbl}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
Return $\mathsf{Tbl}[K, N, V]$

Figure 12: Game G_7 in the proof of Theorem 3.4.

Moreover,

$$\begin{aligned} \Pr[G_3(\mathcal{A}, \mathsf{true})] &\leq \Pr[G_3(\mathcal{A}) \text{ sets } \mathsf{bad}] \\ &= \Pr[G_4(\mathcal{A}) \text{ sets } \mathsf{bad}] = \Pr[G_4(\mathcal{A}, \mathsf{true})] \ . \end{aligned}$$

Hence

$$\begin{aligned} \Pr[G_3(\mathcal{A})] &= \Pr[G_3(\mathcal{A}, \mathsf{false})] + \Pr[G_3(\mathcal{A}, \mathsf{true})] \\ &\leq \Pr[G_4(\mathcal{A}, \mathsf{false})] + \Pr[G_4(\mathcal{A}, \mathsf{true})] = \Pr[G_4(\mathcal{A})] \ . \end{aligned}$$

In game G_5, for each verification query $\textsc{Vf}(i, D, A, C^*\|T)$, we check if there is a prior encryption query $\textsc{Enc}(i, N, A, M)$ that produces the same T, and $D = \mathsf{SE.df}(N)$. If yes, we recover the inner tag R of this encryption query, and return whether $\mathsf{SE.Dec}(K_i, D, \varepsilon, C^*\|R) \neq \perp$. Otherwise, return false.

We now bound the gap between G_4 and G_5. Let Bad be the event that in game G_5, there are two encryption queries (i, N, A, M) and (i, N^*, A, M^*) that end with the same tag T. Since each encryption query picks a fresh random tag, the chance that Bad happens is at most $q^2/2^\lambda$. Suppose that Bad doesn't happen. Without loss of generality, assume that $\mathcal{A}$ returns 1 if some verification query returns true. Note that G_4 and G_5 have the same way of implementing encryption queries. Consider a verification query $\textsc{Vf}(i, D, A, C^*\|T)$. Since $\mathcal{A}$ is orderly, without loss of generality, assume that this is the first verification query. Let $(M, N, R) \leftarrow \mathsf{SE.Tag}(K_i, D, \varepsilon, C^*)$. If there is a prior encryption query (i, N, A, M^*) whose internal tag is R and the external tag is T, then G_5 is simply a different implementation of G_4, using $\mathsf{SE.Dec}$ instead of $\mathsf{SE.Tag}$. Suppose that there is no such encryption query. Without loss of generality, assume that in this case G_5 merely returns false for the verification query, this can only decrease $\Pr[G_5(\mathcal{A})]$ and thus increase $\Pr[G_4(\mathcal{A})] - \Pr[G_5(\mathcal{A})]$. Game G_4 instead checks whether T is the same as $T^* \leftarrow \overline{\textsc{Ro}}(K_i, N, A\|R)$. Note that in game G_4, either $T^* \neq T$ (because there is an encryption query (i, N, A, M^*) of the same internal tag R but its external tag, which is also T^*, is different from T), or the tag T^* is a fresh random string (because there is no encryption query (i, N, A, M^*) of the same internal tag R). Hence the chance that $T = T^*$ is at most $2^{-\lambda}$. Summing this over q_v verification queries and taking into account Pr[Bad],

$$\Pr[G_4(\mathcal{A})] - \Pr[G_5(\mathcal{A})] \leq \frac{q_v}{2^\lambda} + \frac{q^2}{2^\lambda} \ .$$

In game G_6, we instead sample the keys K_i uniformly. Then

$$\Pr[G_5(\mathcal{A})] - \Pr[G_6(\mathcal{A})] \leq \frac{u(u+p)}{2^k} \ .$$

In game G_7, for each encryption query, instead of using $\mathsf{SE.Enc}$ we simply pick a random ciphertext. Likewise, for each verification query, instead of using $\mathsf{SE.Dec}$, we simply return false.

To bound the gap between G_6 and G_7, we construct an adversary $\mathcal{B}$ attacking the AE security of SE as follows. It first initializes $\mathsf{Nonces}, \mathsf{Keys} \leftarrow \emptyset$. It then runs $\mathcal{A}$ and simulates game G_6, but with the following differences. For each encryption query (i, N, A, M), instead of using $\mathsf{SE.Enc}(K_i, N, \varepsilon, M)$, it uses $\textsc{Enc}(i, N, \varepsilon, M)$ and adds N to Nonces. For each verification query $(i, D, A, C^*\|T)$, instead of checking $\mathsf{SE.Dec}(K_i, D, \varepsilon, C^*\|R) \neq \perp$, it runs $\textsc{Vf}(i, D, \varepsilon, C^*\|R)$. For each random-oracle query $\textsc{Ro}(K, N, V)$, instead of checking if $K \in \{K_1, \ldots, K_v\}$, it simply adds K to Keys. Finally, when $\mathcal{A}$ terminates, adversary $\mathcal{B}$ picks $N^* \in \mathcal{N} \backslash \mathsf{Nonces}$, and computes $C'_i \leftarrow \textsc{Enc}(i, N^*, \varepsilon, 0^k)$ for every user i. Then, for each $K \in \mathsf{Keys}$, it computes $C \leftarrow \mathsf{SE.Enc}(K, N^*, \varepsilon, 0^k)$, and sets $\mathsf{win} \leftarrow \mathsf{true}$ if there is some $C'_i = C$. Then

$$\Pr[G_6(\mathcal{A})] \leq \Pr[\mathbf{G}^{\mathrm{real}}_{\mathsf{SE}}(\mathcal{B})] \ .$$

Here we have to use an inequality, because there might be false positives in checking $C'_i = C$ where the string K is not the key K_i. On the other hand,

$$\Pr[G_7(\mathcal{A})] \geq \Pr[\mathbf{G}^{\mathrm{rand}}_{\mathsf{SE}}(\mathcal{D})] - \frac{up}{2^{k+\mathsf{SE.tl}}} \ ,$$

because each C'_i is uniformly chosen from $\{0,1\}^{k+\mathsf{SE.tl}}$, independent of what $\mathcal{A}$ receives. Subtracting, we obtain

$$\Pr[G_6(\mathcal{A})] - \Pr[G_7(\mathcal{A})] \ \leq \ \mathbf{Adv}^{\mathrm{ae}}_{\mathsf{SE}}(\mathcal{B}) + \frac{up}{2^{k+\mathsf{SE.tl}}} \leq \mathbf{Adv}^{\mathrm{ae}}_{\mathsf{SE}}(\mathcal{D}) + \frac{up}{2^k} \ .$$

Summing up,

$$\begin{aligned}\mathbf{Adv}^{\mathrm{ae}}_{\overline{\mathsf{SE}}}(\mathcal{A}) &= \Pr[G_1(\mathcal{A})] - \Pr[G_7(\mathcal{A})] \\ &= \sum_{i=1}^{6} \Pr[G_i(\mathcal{A})] - \Pr[G_{i+1}(\mathcal{A})] \leq \mathbf{Adv}^{\mathrm{ae}}_{\mathsf{SE}}(\mathcal{B}) + \frac{u(2u+3p)}{2^k} + \frac{q_v+q^2}{2^\lambda} \ .\end{aligned}$$

By accounting for the loss of $2q_v/2^\lambda$ in the advantage by assuming that $\mathcal{A}$ is orderly,

$$\mathbf{Adv}^{\mathrm{ae}}_{\overline{\mathsf{SE}}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{ae}}_{\mathsf{SE}}(\mathcal{B}) + \frac{u(2u+3p)}{2^k} + \frac{3q_v+q^2}{2^\lambda} \ .$$

D Proofs of the Lemmas in Proposition 5.1

D.1 Proof of Lemma 5.5

Let $t = \lceil m/2 \rceil$. For any particular bin, the chance that it has at least t balls is at most

$$\binom{q}{t}(c/2^m)^t \leq \frac{q^t}{t!} \cdot (c/2^m)^t \leq \frac{(qc/2^m)^t}{(t/e)^t} \leq \frac{1}{t^t} \ ,$$

where the second inequality is due to Stirling's formula, and the last inequality is due to the fact that $3qc \leq 2^m$. By the union bound, the chance that there is some bin among 2^m possible ones that has at least t balls is at most

$$\frac{2^m}{t^t} \leq \frac{2^m}{16^t} \leq \frac{2^m}{16^{m/2}} = 2^{-m} \ ,$$

where the first inequality is due to the fact that $t \geq m/2 \geq 16$ and the second inequality is due to the fact that $t \geq m/2$.

D.2 Proof of Lemma 5.7

We classify the balls into the following three groups.

- **Group 1:** The ball is created by log entry $(\cdot, K, \mathsf{pad}(T,1), \cdot)$ and a subsequent even match Q^*.
- **Group 2:** The ball is created by log entry $(+, K, \mathsf{pad}(T,1), \cdot)$ and a prior even match Q^*.
- **Group 3:** The ball is created by log entry $(-, K, \mathsf{pad}(T,1), \cdot)$ and a prior even match Q^*.

Note that a ball may belong to more than one group, meaning that we double count some balls, but this does not matter, as we only need to find an upper bound on the number of balls. We will show that for each fixed key K, for Group 1 or Group 2, with probability at least $1 - 1/2^n$, no bin contains more than $n/2$ balls from this group. For Group 3, with probability at least $1 - 2/2^n$, no bin contains more than $n/2$ balls from this group. Thus for each fixed key K, with probability at least $1 - 4/2^n$, each bin associated with key K contains at most $3n/2$ balls. Summing this over all keys, with probability at least $1 - 4q/2^n$, every bin has at most $3n/2$ balls.

GROUP 1. Consider the following view of throwing balls in Group 1. For each even match $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*,0), T), (+, K^*, \mathsf{pad}(T,1), X^*)\}$, we throw a ball to bin $(K, X \oplus X^* \oplus \mathsf{pad}(P^*,0))$ if there is a prior entry $(\cdot, K, \mathsf{pad}(T,1), X)$. This view is incorrect because it may double count some balls, but since we want an upper bound on the number of balls, this does not matter. So we have a game of at most q rounds of ball throwing, each corresponding to an even match Q^*. In each round, we probabilistically pick some bins and put a ball in each bin.

For each round, given the prior throws and the associated $(X, \mathsf{pad}(P^*, 0))$, the corresponding X^* is uniformly distributed over a set of at least $15 \cdot 2^n/16$ members. Thus for each fixed bin, the chance this round puts a ball to this bin is at most $16/(15 \cdot 2^n)$. Using Lemma 5.5 with $c = 16/15$ and $m = n$, with probability at least $1 - 1/2^n$, every bin associated with key K has at most $n/2$ balls.

GROUP 2. For each query $E(K, \mathsf{pad}(T, 1))$, its *hitting matches* are the prior even matches $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*, 0), T), (+, K^*, \mathsf{pad}(T, 1), X^*)\}$. Note that two different queries $E(K, \mathsf{pad}(T, 1))$ and $E(K, \mathsf{pad}(T', 1))$ have disjoint sets of hitting matches.

For each query $E(K, \mathsf{pad}(T, 1))$, its ball throwing can be viewed as follows. Recall that the answer of query $E(K, \mathsf{pad}(T, 1))$ is uniformly chosen from a set S^*. Consider a bin (K, Δ). Let S be the set of n-bit strings U such that there is some hitting match $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*, 0), T), (+, K^*, \mathsf{pad}(T, 1), X^*)\}$ of this query with

$$U = X^* \oplus \mathsf{pad}(P^*, 0) \oplus \mathsf{pad}(T, 1) \oplus \Delta \ .$$

Then we put a ball to bin (K, Δ) with (conditional) probability $|S \cap S^*|/|S^*|$. In addition, note that the number B of hitting matches of this query is at least $|S|$.

Now consider an (incorrect) view in which for each even match Q^*, we put a ball into bin (K, Δ) with probability $32/(15 \cdot 2^n)$, independently for every Δ. Thus in the latter view, for a query $E(K, \mathsf{pad}(T, 1))$, the probability that its B hitting matches put balls to bin (K, Δ) is at least

$$1 - \Big(1 - \frac{1}{(15 \cdot 2^n)/32}\Big)^B \ \geq \ \frac{B}{(15 \cdot 2^n)/16} \geq \frac{|S|}{(15 \cdot 2^n)/16} \ \geq \frac{|S|}{|S^*|} \geq \frac{|S \cap S^*|}{|S^*|} \ ,$$

where the first inequality is due to Lemma 5.6. Since we consider an upper bound on the number of balls, we can use the latter view. Using Lemma 5.5 with $c = 32/15$ and $m = n$, with probability at least $1 - 1/2^n$, every bin associated with key K has at most $n/2$ balls.

There is a technical subtlety here. Let p_1 be the probability that in the correct view, some bin associated with key K has at least $n/2$ balls, and define p_2 similarly for the incorrect view. Let ϵ be the probability that a *fixed* bin has at least $n/2$ balls under the incorrect view, and recall that there are 2^n bins associated with key K. Our argument only gives us $p_1 \leq 2^n \cdot \epsilon$, but not $p_1 \leq p_2$. However, we can ignore the distinction between p_2 and $2^n \cdot \epsilon$ because the result in Lemma 5.5 is actually an upper bound for $2^n \cdot \epsilon \geq p_2$.

GROUP 3. An even match $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*, 0), T), (+, K^*, \mathsf{pad}(T, 1), X^*)\}$ is called a Δ-*match* if $\mathsf{pad}(P^*, 0) \oplus X^* \oplus \mathsf{pad}(T, 1) = \Delta$. We claim that with probability at least $1 - 1/2^n$, for every $\Delta \in \{0, 1\}^n$, the number of Δ-matches is at most $n/2$. To justify this, view each even match Q^* as throwing a ball to bin $\mathsf{pad}(P^*, 0) \oplus X^* \oplus \mathsf{pad}(T, 1)$. In this new game, for each particular bin, the chance that each match throws a ball to it is at most $16/(15 \cdot 2^n)$. The claim then follows directly from Lemma 5.5 with $c = 16/15$ and $m = n$.

Now, back to the original balls-into-bins game, and fix a bin (K, Δ). When we consider balls in Group 3, for each query $E^{-1}(K, V)$, we put a ball to this bin if the answer of this query hits one of the $\mathsf{pad}(T, 1)$ of the $(\Delta \oplus V)$-matches. Thus for query $E^{-1}(K, V)$, we put a ball to bin (K, Δ) with probability at most $8n/(15 \cdot 2^n)$. Using Lemma 5.5 with $m = n$ and $c = 8n/15$, with probability at least $1 - 1/2^n$, every bin associated with key K has at most $n/2$ balls.

D.3 Proof of Lemma 5.8

We classify the balls into the following three groups.

- **Group 1:** The ball is created by log entry $(\cdot, K, \mathsf{pad}(T, 1), \cdot)$ and a subsequent even match Q^*.
- **Group 2:** The ball is created by log entry $(+, K, \mathsf{pad}(T, 1), \cdot)$ and a prior even match Q^*.

- **Group 3:** The ball is created by log entry $(-, K, \mathsf{pad}(T, 1), \cdot)$ and a prior even match Q^*.

Note that a ball may belong to more than one group, meaning that we double count some balls, but this does not matter, as we only need to find an upper bound on the number of balls. We will show that for each fixed key K, for Group 1 or Group 2, with probability at least $1 - 1/2^\tau$, no bin associated with key K contains more than $\tau/2$ balls from this group. For Group 3, with probability at least $1 - 1/2^n - 1/2^\tau$, no bin associated with key K contains more than $n/2$ balls from this group. Thus with probability at least $1 - 3/2^\tau - 1/2^n$, each bin associated with key K contains at most $\tau + n/2$ balls. Summing this over all keys, with probability at least $1 - 3q/2^\tau - q/2^n$, every bin has at most $\tau + n/2$ balls.

GROUP 1. Consider the following (incorrect) view of throwing balls in Group 1. For each even match $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*, 0), T), (+, K^*, \mathsf{pad}(T, 1), X^*)\}$, we throw a ball to bin

$$(K, (X \oplus X^* \oplus \mathsf{pad}(P^*, 0))[1 : \tau] \oplus T)$$

if there is a prior entry $(\cdot, K, \mathsf{pad}(T, 1), X)$. This view may double count some balls, but since we want an upper bound on the number of balls, this does not matter. So we have a game of at most q rounds of ball throwing, each corresponding to an even match Q^*. In each round, we probabilistically pick some bins and put a ball in each bin.

For each round, given the prior throws and the associated $(X, \mathsf{pad}(P^*, 0), T)$, the corresponding X^* is uniformly distributed over a set of at least $16 \cdot 2^n/15$ members. Thus for each fixed bin, the chance this round puts a ball to this bin is at most $16/(15 \cdot 2^\tau)$. Using Lemma 5.5 with $c = 16/15$ and $m = \tau$, with probability at least $1 - 1/2^\tau$, every bin associated with key K has at most $\tau/2$ balls.

GROUP 2. For each query $E(K, \mathsf{pad}(T, 1))$, its *hitting matches* are the prior even matches $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*, 0), T), (+, K^*, \mathsf{pad}(T, 1), X^*)\}$. Note that two different queries $E(K, \mathsf{pad}(T, 1))$ and $E(K, \mathsf{pad}(T', 1))$ have disjoint sets of hitting matches.

For each query $E(K, \mathsf{pad}(T, 1))$, its ball throwing can be viewed as follows. Recall that the answer of query $E(K, \mathsf{pad}(T))$ is uniformly chosen from a set S^*. For each bin (K, Δ), let S be the set of n-bit strings U such that there is some hitting match $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*, 0), T), (+, K^*, \mathsf{pad}(T, 1), X^*)\}$ of this query with

$$U[1 : \tau] = (X^* \oplus \mathsf{pad}(P^*, 0))[1 : \tau] \oplus \Delta \ .$$

Then we put a ball to bin (K, Δ) with (conditional) probability $|S \cap S^*|/|S^*|$. Note that if this query has B hitting matches then $B \geq |S|/2^{n-\tau}$.

Now consider another (incorrect) view. For each even match Q^*, we simply put a ball into bin (K, Δ) with probability $32/(15 \cdot 2^\tau)$, independently for every Δ. In in the latter view, for a query $E(K, \mathsf{pad}(T))$, the probability that its B hitting matches put balls to bin (K, Δ) is at least

$$1 - \left(1 - \frac{1}{(15 \cdot 2^\tau)/32}\right)^B \quad \geq \quad \frac{B}{(15 \cdot 2^\tau)/16} \geq \frac{|S|}{(15 \cdot 2^n)/16} \geq \frac{|S|}{|S^*|} \geq \frac{|S \cap S^*|}{|S^*|} \ ,$$

where the first inequality is due to Lemma 5.6. Since we consider an upper bound on the number of balls, we can use the latter view. Using Lemma 5.5 with $c = 32/15$ and $m = \tau$, with probability at least $1 - 1/2^\tau$, every bin associated with key K has at most $\tau/2$ balls.

GROUP 3. An even match $Q^* = \{(\cdot, K^*, \mathsf{pad}(P^*, 0), T), (+, K^*, \mathsf{pad}(T, 1), X^*)\}$ is called a Δ-*match* if $(\mathsf{pad}(P^*, 0) \oplus X^*)[1 : \tau] = \Delta$. We claim that with probability at least $1 - 1/2^\tau$, for every $\Delta \in \{0, 1\}^\tau$, the number of Δ-matches is at most $\tau/2$. To justify this, view each even match Q^* as throwing a ball to bin $(\mathsf{pad}(P^*, 0) \oplus X^*)[1 : \tau]$. In this new game, for each particular bin, the chance that each match throws a ball to it is at most $16/(15 \cdot 2^\tau)$. The claim then follows directly from

Lemma 5.5 with $c = 16/15$ and $m = \tau$.

Now, back to the original balls-into-bins game, and fix a bin (K, Δ). When we consider balls in Group 3, for each query $E^{-1}(K, V)$, we put a ball to this bin if the answer of this query hits one of the $\mathsf{pad}(T, 1)$ of the $(\Delta \oplus V[1:\tau])$-matches. Thus for query $E^{-1}(K, V)$, we put a ball to bin (K, Δ) with probability at most $8\tau/(15 \cdot 2^n)$. Using Lemma 5.5 with $m = n$ and $c = 8\tau/15$, with probability at least $1 - 1/2^n$, every bin associated with key K has at most $n/2$ balls.

D.4 Proof of Lemma 5.9

Consider the following balls-into-bins game. We view each odd match $Q^* = \{(\cdot, K^*, \mathsf{pad}(T, 1), X^*), (\cdot, K^*, \mathsf{pad}(P^*, 0), T)\}$ as throwing a ball to bin

$$(X^* \oplus \mathsf{pad}(P^*, 0))[1:\tau] \ .$$

Our goal is to bound the number of balls in the heaviest bin. As a stepping stone, we define the following events Ω_0, Ω_1, and Ω_2.

- Event Ω_0: For any $R \in \{0,1\}^\tau$, there are at most $\tau/2$ log entries $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ such that $X^*[1:\tau] = R$. To bound $\Pr[\overline{\Omega_0}]$, view each entry $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ as throwing a ball to bin $X^*[1:\tau]$. (Recall that there may be $2q$ such entries, including the granted ones.) By applying Lemma 5.5 with $m = \tau$ and $c = 16/15$ for $2q$ queries, we have $\Pr[\overline{\Omega_0}] \le 1/2^\tau$.
- Event Ω_1: For every $\Delta \in \{0,1\}^\tau$, there are at most $\tau/2$ queries $V \leftarrow E(K^*, \mathsf{pad}(T, 1))$ such that $V[1:\tau] = \Delta$. To bound $\Pr[\overline{\Omega_1}]$, view each $V \leftarrow E(K^*, \mathsf{pad}(T, 1))$ as throwing a ball into bin $V[1:\tau]$. (Recall that there may be $2q$ such entries, including the granted ones.) By applying Lemma 5.5 with $m = \tau$ and $c = 16/15$ for $2q$ queries, we have $\Pr[\overline{\Omega_1}] \le 1/2^\tau$.
- Event Ω_2: For every key K^* and every $\Delta \in \{0,1\}^\tau$, there are at most τ queries $\mathsf{pad}(T, 1) \leftarrow E^{-1}(K^*, V)$ with $T \in \{0,1\}^\tau$ such that $V[1:\tau] = \Delta$. We claim that

  $$\Pr[\overline{\Omega_2}] \le \frac{q}{2^\tau} \ .$$

  To justify this claim, it suffices to show that for each fixed K^* and $\Delta \in \{0,1\}^\tau$, the chance that $\overline{\Omega_2}$ happens is at most $4^{-\tau}$. Let $p = 2^{n-\tau}$. On the one hand, there are at most p queries $E^{-1}(K^*, V)$ with $V[1:\tau] = \Delta$. On the other hand, for each such query, the chance that its answer U ends with $1^{n-\tau}$ so that U can be parsed as $\mathsf{pad}(T, 1)$ is at most $16/(15 \cdot p)$. Thus the chance that there are τ such queries that end with $1^{n-\tau}$ is at most

  $$\binom{p}{\tau} \cdot \left(\frac{16}{15 \cdot p}\right)^\tau \ \le \ \frac{p^\tau}{\tau!} \cdot \left(\frac{16}{15 \cdot p}\right)^\tau \le \frac{p^\tau}{(\tau/e)^\tau} \cdot \left(\frac{16}{15 \cdot p}\right)^\tau = \left(\frac{16}{15\tau/e}\right)^\tau \le 4^{-\tau} \ ,$$

  where the second inequality is due to the Stirling's formula, and the last inequality is due to the fact that $\tau \ge 32$.

Assume that Ω_0, Ω_1, and Ω_2 happen; this has probability at least $1 - (q+2)/2^\tau$. We claim that for any throw, given prior throws, the conditional probability that it lands in any particular bin Δ is at most $8\tau/(5 \cdot 2^\tau)$. Then, using Lemma 5.5 for $m = \tau$ and $c = 8\tau/5$, with (conditional) probability at least $1 - 1/2^\tau$, every bin has at most $\tau/2$ balls. To justify the claim above, we consider whether the first entry of the match Q^* of the ball is created by a forward or backward query.

FORWARD QUERY. Each query $V \leftarrow E(K^*, \mathsf{pad}(P^*, 0))$ targets prior entries $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ such that

$$V[1:\tau] \ = \ \mathsf{pad}(P^*, 0)[1:\tau] \oplus T \ , \text{ and} \tag{31}$$

$$X^*[1:\tau] \ = \ \mathsf{pad}(P^*, 0)[1:\tau] \oplus \Delta \ . \tag{32}$$

Since Ω_0 happens, there are at most $\tau/2$ prior entries satisfying Equation (32). The chance that

one of those $\tau/2$ entries also satisfies Equation (31) is at most

$$\frac{\tau}{2} \cdot \frac{16}{15 \cdot 2^\tau} \leq \frac{8\tau}{5 \cdot 2^\tau} \ .$$

Backward query. For each backward query $\mathsf{pad}(P^*, 0) \leftarrow E^{-1}(K^*, V)$, it targets prior entries $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ such that

$$\mathsf{pad}(P^*, 0)[1:\tau] = V[1:\tau] \oplus T \ , \text{ and} \tag{33}$$

$$(X^* \oplus \mathsf{pad}(P^*, 0))[1:\tau] = \Delta \ . \tag{34}$$

From Equation (33) and the fact that $T = \mathsf{pad}(T, 1)[1:\tau]$, we can replace Equation (34) with

$$(X^* \oplus \mathsf{pad}(T, 1))[1:\tau] = \Delta \oplus V[1:\tau] \ . \tag{35}$$

Note that $X^* \oplus \mathsf{pad}(T, 1) = E(K^*, \mathsf{pad}(T, 1))$. Since Ω_1 happens, there are at most $\tau/2$ prior entries $(+, K^*, \mathsf{pad}(T, 1), X^*)$ satisfying Equation (35). Moreover, since Ω_2 happens, there are at most τ prior entries $(-, K^*, \mathsf{pad}(T, 1), X^*)$ satisfying Equation (35). The chance that one of those $3\tau/2$ entries also satisfies Equation (33) is at most

$$\frac{3\tau}{2} \cdot \frac{16}{15 \cdot 2^\tau} = \frac{8\tau}{5 \cdot 2^\tau} \ .$$

D.5 Proof of Lemma 5.10

Let Ω be the following event: For any $R \in \{0,1\}^\tau$, there are at most $\tau/2$ entries $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ such that $X^*[1:\tau] = R$. To bound $\Pr[\overline{\Omega}]$, view each entry $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ as throwing a ball into bin $X^*[1:\tau]$. (Recall that there may be $2q$ such entries, including the granted ones.) By applying Lemma 5.5 with $m = \tau$ and $c = 16/15$ for $2q$ queries, we have $\Pr[\overline{\Omega}] \leq 1/2^\tau$. From now on, assume that Ω happens.

Consider the following balls-into-bins game. View each odd match $Q^* = \{(\cdot, K^*, \mathsf{pad}(T, 1), X^*), (\cdot, K^*, \mathsf{pad}(P^*, 0), T)\}$ as throwing a ball to bin

$$X^* \oplus \mathsf{pad}(P^*, 0) \oplus \mathsf{pad}(T, 1) \ .$$

Our goal is to bound the number of balls in the heaviest bin. We claim that for any throw, given prior throws, the conditional probability that it lands in any particular bin Δ is at most $8\tau/(15 \cdot 2^\tau)$. Then, using Lemma 5.5 for $m = n$ and $c = 8\tau(2^{n-\tau})/15$, with (conditional) probability at least $1 - 1/2^n$, every bin has at most $n/2$ balls. To justify the claim above, we consider whether the first entry of the match Q^* of the ball is created by a forward or backward query.

Forward query. For each query $V \leftarrow E(K^*, \mathsf{pad}(P^*, 0))$, it aims at prior entries $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ such that

$$V[1:\tau] = \mathsf{pad}(P^*, 0)[1:\tau] \oplus T \ , \text{ and} \tag{36}$$

$$X^* \oplus \mathsf{pad}(T, 1) = \mathsf{pad}(P^*, 0) \oplus \Delta \ . \tag{37}$$

Note that $X^* \oplus \mathsf{pad}(T, 1) = E(K^*, \mathsf{pad}(T, 1))$. There is at most one prior entry $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ satisfying Equation (37). The probability that this entry also satisfies Equation (36) is at most $16/(15 \cdot 2^\tau)$.

Backward query. For each backward query $\mathsf{pad}(P^*, 0) \leftarrow E^{-1}(K^*, V)$, it targets prior entries $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ such that

$$\mathsf{pad}(P^*, 0)[1:\tau] = V[1:\tau] \oplus T \ , \text{ and} \tag{38}$$

$$X^* \oplus \mathsf{pad}(T, 1) = \mathsf{pad}(P^*, 0) \oplus \Delta \ . \tag{39}$$

From Equation (38) and the fact that $T = \mathsf{pad}(T, 1)[1 : \tau]$, we can replace Equation (39) with

$$X^*[1 : \tau] = (\Delta \oplus V)[1 : \tau] \ . \tag{40}$$

Since Ω happens, there are at most $\tau/2$ prior entries satisfying Equation (40). The chance that one of those $\tau/2$ entries also satisfies Equation (38) is at most $8\tau/(15 \cdot 2^\tau)$.

D.6 Proof of Lemma 5.11

For each entry $(\cdot, K, \mathsf{pad}(T, 1), X)$ and each $\Delta \in \{0, 1\}^\tau$, we throw a ball to bin (K, Δ), if there is another entry $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$ with $K^* \neq K$ and $(X^* \oplus X)[1 : \tau] = \Delta$. Our goal is to bound the number of balls in the heaviest bin. We classify the balls into the following groups.

- **Group 1:** The ball is created by a log entry $(\cdot, K, \mathsf{pad}(T, 1), X)$ and a subsequent log entry $(+, K^*, \mathsf{pad}(T, 1), X^*)$.
- **Group 2:** The ball is created by a log entry $(\cdot, K, \mathsf{pad}(T, 1), X)$ and a subsequent log entry $(-, K^*, \mathsf{pad}(T, 1), X^*)$.
- **Group 3:** The ball is created by an entry $(+, K, \mathsf{pad}(T, 1), X)$ and a prior $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$.
- **Group 4:** The ball is created by an entry $(-, K, \mathsf{pad}(T, 1), X)$ and a prior $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$.

Note that a ball may belong to more than one group, and as a result, we may double count some balls. However, as we only consider an upper bound on the number of balls, it does not matter. Fix a key K. Below, we will show that each of Group 1 and Group 3, with probability at least $1 - 1/2^\tau$, gives at most $\tau/2$ balls to each bin associated with key K. Moreover, with probability at least $1 - 2/2^n$, Group 2 gives at most $n/2$ balls to each bin associated with key K. Likewise, with probability at least $1 - 1/2^n$, Group 4 gives at most $n/2$ balls to each bin associated with key K. Summing up for all groups and all q keys, with probability at least $1 - 3q/2^n - 2q/2^\tau$, there are at most $n + \tau$ balls per bin.

Group 1. Consider the following (incorrect) view of throwing balls in Group 1. For each log entry $(+, K^*, \mathsf{pad}(T, 1), X^*)$, if there is a prior log entry $(\cdot, K, \mathsf{pad}(T, 1), X)$ then we throw a ball to bin $(X^* \oplus X)[1 : \tau]$. This view may double count some balls, but since we only consider an upper bound on the number of balls, this does not matter. For each entry $(+, K^*, \mathsf{pad}(T, 1), X^*)$, given prior entries, the string X^* is uniformly distributed over a set of at least $15 \cdot 2^n/16$ members. Thus for each fixed bin, the chance that this entry puts a ball to this bin is at most $16/(15 \cdot 2^\tau)$. Using Lemma 5.5 with $c = 16/15$ and $m = \tau$ for $2q$ rounds, with probability at least $1 - 1/2^\tau$, every bin has at most $\tau/2$ balls.

Group 2. The ball throwing in Group 2 can be viewed as follows. For each query $E^{-1}(K^*, U)$, it puts a ball to bin (K, Δ) if there is a prior entry $(\cdot, K, \mathsf{pad}(T, 1), X)$ such that

$$(X \oplus \mathsf{pad}(T, 1) \oplus U)[1 : \tau] = \Delta \ , \tag{41}$$

and the answer of this query hits the $\mathsf{pad}(T, 1)$ of the prior entry. The difficulty here is that $\mathcal{A}$ may *adaptively* picks (K^*, U) to maximize the number of entries satisfying Equation (41). Still, we claim that with probability at least $1 - 1/2^n$, for any query $E^{-1}(K^*, U)$ there are at most $(2^{n-\tau} + \tau/2)$ prior entries that satisfy Equation (41). Hence the chance that the answer of this query hits the $\mathsf{pad}(T, 1)$ of one of those entries is at most $16(2^{n-\tau} + \tau/2)/(15 \cdot 2^n)$. Using Lemma 5.5 with $c = 16(2^{n-\tau} + \tau/2)/15$ and $m = n$, with probability at least $1 - 1/2^n$, each bin has at most $n/2$ balls.

To justify the claim above, we consider whether entries are created by forward or backward queries.

Case 1: Entries $(\cdot, K, \mathsf{pad}(T,1), X)$ are created from backward queries $E^{-1}(K,V)$. Then $V = X \oplus \mathsf{pad}(T,1)$. For any adaptive query $E^{-1}(K,U)$ and any $\Delta \in \{0,1\}^n$, there are at most $2^{n-\tau}$ prior query $E^{-1}(K,V)$ such that $V = U \oplus \Delta$. That is, there are at most $2^{n-\tau}$ entries $(-, K, \mathsf{pad}(T,1), X)$ satisfying Equation (41).

Case 2: Entries $(\cdot, K, \mathsf{pad}(T,1), X)$ are created from forward queries $V \leftarrow E(K, \mathsf{pad}(T,1))$, and note that $V = X \oplus \mathsf{pad}(T)$. Consider another balls-into-bins game. View each query $V \leftarrow E(K, \mathsf{pad}(T,1))$ as throwing a ball to bin $V[1:\tau]$. By applying Lemma 5.5 with $c = 16/15$ and $m = \tau$ for $2q$ rounds, with probability at least $1 - 1/2^\tau$, there are at most $\tau/2$ balls in each bin. Then for each adaptive query (K^*, U) and any $\Delta \in \{0,1\}^n$, its number of entries $(+, K, \mathsf{pad}(T,1), X)$ satisfying Equation (41) is exactly the number of balls in bin $U \oplus \Delta$ in the new game.

Group 3. For each query $E(K, \mathsf{pad}(T,1))$, its *hitting queries* are prior entries $(\cdot, K^*, \mathsf{pad}(T,1), X^*)$ with $K^* \neq K$. Note that two different queries $E(K, \mathsf{pad}(T,1))$ and $E(K, \mathsf{pad}(T',1))$ have disjoint sets of hitting queries.

For each query $E(K, \mathsf{pad}(T,1))$, its ball throwing can be viewed as follows. Recall that the answer of query $E(K, \mathsf{pad}(T))$ is uniformly chosen from a set S^*. Consider a bin (K, Δ). Let S be the set of n-bit strings U such that there is a hitting query $(\cdot, K^*, \mathsf{pad}(T,1), X^*)$ such that

$$U[1:\tau] = (X^* \oplus \mathsf{pad}(T,1))[1:\tau] \oplus \Delta \ .$$

Then we put a ball to bin (K, Δ) with (conditional) probability $|S \cap S^*|/|S^*|$. Note that the number B of hitting queries of this query is at least $|S|/2^{n-\tau}$.

Now consider another (incorrect) view. For each entry $(\cdot, K^*, \mathsf{pad}(T), X^*)$, we simply put a ball into bin (K, Δ) with probability $32/(15 \cdot 2^\tau)$, independently for every Δ. In the latter view, for a query $E(K, \mathsf{pad}(T,1))$, the chance that its B hitting queries put balls to bin Δ is at least

$$\begin{aligned} 1 - \left(1 - \frac{1}{(15 \cdot 2^\tau)/32}\right)^B &\geq \frac{B}{(15 \cdot 2^\tau)/16} \\ &\geq \frac{|S|}{(15 \cdot 2^n)/16} \geq \frac{|S|}{|S^*|} \geq \frac{|S \cap S^*|}{|S^*|} \ , \end{aligned}$$

where the first inequality is due to Lemma 5.6. Since we consider an upper bound on the number of balls, we can instead use the latter view. Using Lemma 5.5 with $c = 32/15$ and $m = \tau$ for $2q$ rounds, with probability at least $1 - 1/2^\tau$, every bin associated with key K^* has at most $\tau/2$ balls.

Group 4. Fix a bin (K, Δ). The ball throwing in Group 4 can be viewed (incorrectly) as follows. For each entry $(\cdot, K^*, \mathsf{pad}(T,1), X^*)$, if there is a subsequent query $\mathsf{pad}(T,1) \leftarrow E^{-1}(K,V)$ with $V[1:\tau] = (X^* \oplus \mathsf{pad}(T,1))[1:\tau] \oplus \Delta$ then we put a ball to bin (K, Δ). While this view may double count some balls, as we only consider an upper bound on the number of balls, this does not matter. Now, for each entry $(\cdot, K^*, \mathsf{pad}(T,1), X^*)$, there are at most $2^{n-\tau}$ subsequent query $E^{-1}(K,V)$ with $V = (X^* \oplus \mathsf{pad}(T))[1:\tau] \oplus \Delta$, and the chance that the answer of this query hits $\mathsf{pad}(T,1)$ is at most $16/(15 \cdot 2^n)$. Using Lemma 5.5 with $c = 16 \cdot 2^{n-\tau}/15$ and $m = n$ for $2q$ rounds, with probability at least $1 - 1/2^n$, there are at most $n/2$ balls in each bin.

D.7 Proof of Lemma 5.12

For each entry $(\cdot, K, \mathsf{pad}(T,1), X)$ and each $\Delta \in \{0,1\}^\tau$, we throw a ball to bin (K, Δ), if there is another entry $(\cdot, K^*, \mathsf{pad}(T,1), X^*)$ with $K^* \neq K$ and $(X^* \oplus X)[1:\tau] \oplus T = \Delta$. Our goal is to bound the number of balls in the heaviest bin. We classify the balls into the following groups.

- **Group 1:** The ball is created by a log entry $(\cdot, K, \mathsf{pad}(T,1), X)$ and a subsequent log entry $(+, K^*, \mathsf{pad}(T,1), X^*)$.

- **Group 2:** The ball is created by a log entry $(\cdot, K, \mathsf{pad}(T,1), X)$ and a subsequent log entry $(-, K^*, \mathsf{pad}(T,1), X^*)$.
- **Group 3:** The ball is created by an entry $(+, K, \mathsf{pad}(T,1), X)$ and a prior $(\cdot, K^*, \mathsf{pad}(T,1), X^*)$.
- **Group 4:** The ball is created by an entry $(-, K, \mathsf{pad}(T,1), X)$ and a prior $(\cdot, K^*, \mathsf{pad}(T,1), X^*)$.

Note that a ball may belong to more than one group, and as a result, we may double count some balls. However, as we only consider an upper bound on the number of balls, it does not matter. Below, we will show that for each fixed key K, for each of Group 1, Group 3, or Group 4, with probability at least $1 - 1/2^n$, it gives at most $n/2$ balls for each bin associated with key K. For Group 2, with probability at least $1 - 2/2^n$, it gives at most $n/2$ balls for each bin associated with key K. Summing up for all groups and all q keys, with probability at least $1 - 5q/2^n$, there are at most $2n$ balls per bin.

GROUP 1. Consider the following (incorrect) view of throwing balls in Group 1. For each log entry $(+, K^*, \mathsf{pad}(T,1), X^*)$, if there is a prior log entry $(\cdot, K, \mathsf{pad}(T,1), X)$ then we throw a ball to bin $(K, (X^* \oplus X)[1:\tau] \oplus T)$. This view may double count some balls, but since we only consider an upper bound on the number of balls, this does not matter. For each entry $(+, K^*, \mathsf{pad}(T,1), X^*)$, given prior entries, the string X^* is uniformly distributed over a set of at least $15 \cdot 2^n/16$ members. Thus for each fixed bin, the chance that this entry puts a ball to this bin is at most $16/(15 \cdot 2^\tau)$. Using Lemma 5.5 with $c = 16/15$ and $m = \tau$ for $2q$ rounds, with probability at least $1 - 1/2^\tau$, every bin has at most $\tau/2$ balls.

GROUP 2. The ball throwing in Group 2 can be viewed as follows. For each query $E^{-1}(K^*, U)$, it puts a ball to bin (K, Δ) if there is a prior entry $(\cdot, K, \mathsf{pad}(T,1), X)$ such that

$$X[1:\tau] = U[1:\tau] \oplus \Delta \ , \tag{42}$$

and the answer of this query hits the $\mathsf{pad}(T,1)$ of the prior entry. The difficulty here is that $\mathcal{A}$ may *adaptively* chooses (K^*, U) to maximize the number of entries satisfying Equation (42). Still, we claim that with probability at least $1 - 1/2^\tau$, for any query $E^{-1}(K^*, U)$ there are at most $\tau/2$ prior entries that satisfy Equation (42). Hence the chance that the answer of this query hits the $\mathsf{pad}(T,1)$ of one of those entries is at most $8\tau/(15 \cdot 2^n) \le 8\tau/(15 \cdot 2^\tau)$. Using Lemma 5.5 with $c = 8\tau/15$ and $m = \tau$ for $2q$ throws, with probability at least $1 - 1/2^\tau$, each bin has at most $\tau/2$ balls.

To justify the claim above, consider a balls-into-bins game. For each entry $(\cdot, K, \mathsf{pad}(T,1), X)$, view it as throwing a ball to bin $X[1:\tau]$. Using Lemma 5.5 with $c = 16/15$ and $m = \tau$ for $2q$ throws, with probability at least $1 - 1/2^\tau$, each bin has at most $\tau/2$ balls. Then for each adaptive query (K^*, U) and any $\Delta \in \{0,1\}^\tau$, its number of entries satisfying Equation (42) is exactly the number of balls in bin $U[1:\tau] \oplus \Delta$ in the new game. Thus our claim directly follows the balls-into-bins analyses above.

GROUP 3. For each query $E(K, \mathsf{pad}(T,1))$, its *hitting queries* are prior entries $(\cdot, K^*, \mathsf{pad}(T,1), X^*)$ with $K^* \neq K$. Note that two different queries $E(K, \mathsf{pad}(T,1))$ and $E(K, \mathsf{pad}(T',1))$ have disjoint sets of hitting queries.

For each query $E(K, \mathsf{pad}(T,1))$, its ball throwing can be viewed as follows. Recall that the answer of query $E(K, \mathsf{pad}(T))$ is uniformly chosen from a set S^*. Consider a bin (K, Δ). Let S be the set of n-bit strings U such that there is a hitting query $(\cdot, K^*, \mathsf{pad}(T,1), X^*)$ such that

$$U[1:\tau] = X^*[1:\tau] \oplus \Delta \ .$$

Then we put a ball to bin (K, Δ) with (conditional) probability $|S \cap S^*|/|S^*|$. Moreover, note that the number B of hitting queries of this query is at least $|S|/2^{n-\tau}$.

Now consider another (incorrect) view. For each entry $(\cdot, K^*, \mathsf{pad}(T), X^*)$, we simply put a ball into bin (K, Δ) with probability $32/(15 \cdot 2^\tau)$, independently for every Δ. In the latter view, for a query $E(K, \mathsf{pad}(T, 1))$, the chance that its B hitting queries put balls to bin Δ is at least

$$\begin{aligned} 1 - \Big(1 - \frac{1}{(15 \cdot 2^\tau)/32}\Big)^B &\geq \frac{B}{(15 \cdot 2^\tau)/16} \\ &\geq \frac{|S|}{(15 \cdot 2^n)/16} \geq \frac{|S|}{|S^*|} \geq \frac{|S \cap S^*|}{|S^*|} \ , \end{aligned}$$

where the first inequality is due to Lemma 5.6. Since we consider an upper bound on the number of balls, we can instead use the latter view. Using Lemma 5.5 with $c = 32/15$ and $m = \tau$ for $2q$ rounds, with probability at least $1 - 1/2^\tau$, every bin associated with key K^* has at most $\tau/2$ balls.

Group 4. The ball throwing in Group 4 can be viewed (incorrectly) as follows. For each entry $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$, if there is a subsequent query $\mathsf{pad}(T, 1) \leftarrow E^{-1}(K, V)$ with $V = X^*[1 : \tau] \oplus \Delta$ then we put a ball to bin (K, Δ). While this view may double count some balls, as we only consider an upper bound on the number of balls, this does not matter. Now, for each entry $(\cdot, K^*, \mathsf{pad}(T, 1), X^*)$, there are at most $2^{n-\tau}$ subsequent queries $E^{-1}(K, V)$ with $V = X^*[1 : m] \oplus \Delta$, and the chance that the answer of one of these queries hits $\mathsf{pad}(T, 1)$ is at most $2^{n-\tau} \cdot 16/(15 \cdot 2^n) = 16/(15 \cdot 2^\tau)$. Using Lemma 5.5 with $c = 16/15$ and $m = \tau$ for $2q$ rounds, with probability at least $1 - 1/2^\tau$, there are at most $\tau/2$ balls in each bin.

E Proof of Proposition 5.2

Orderly adversaries. We say that an adversary is *orderly* if (i) its verification queries are made at the very end, and (ii) each verification query does not depend on the answers of prior verification queries, but may still depend on the answers of prior encryption queries. We now show that for an IPF F, one can restrict to orderly adversaries with a small loss in the IPF advantage.

Lemma E.1 *Let F be an IPF in which a ciphertext is always at least λ-bit longer than its plaintext. Let $\mathcal{A}$ be an adversary that makes at most q_v verification queries. Then we can construct an orderly adversary $\mathcal{B}$ of the same running time and query statistics such that*

$$\mathbf{Adv}^{\mathrm{ipf}}_{\mathsf{F}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{ipf}}_{\mathsf{F}}(\mathcal{B}) + \frac{2q_v}{2^\lambda} \ .$$

Proof: Adversary $\mathcal{B}$ runs $\mathcal{A}$, and lets $\mathcal{A}$ use its New and Enc oracles. For each verification query of $\mathcal{A}$, however, $\mathcal{B}$ stores that in a list L, and simply returns false to $\mathcal{A}$. When $\mathcal{A}$ terminates with its guess b', for each query (i, C) in L, if there is no prior encryption query $C \leftarrow \textsc{Enc}(i, M)$ then $\mathcal{B}$ will query $\textsc{Vf}(i, C)$, otherwise it will terminate and return 1. If one of those verification queries results in a true answer then $\mathcal{B}$ will return 1, otherwise it returns b'.

In the real world of $\mathcal{B}$ (meaning when its challenge bit is b) if some verification query could result in a true answer then $\mathcal{B}$ will answer 1 anyway (even if it can't make this query due to the definitional restriction). If all verification queries are destined to give false answers then $\mathcal{B}$ correctly simulates the real world of $\mathcal{A}$, and it either gives the same answer as $\mathcal{A}$ or returns 1. Hence

$$\Pr[\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{B}) \mid b = 1] \geq \Pr[\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{A}) \mid d = 1] \ ,$$

where b and d are the challenge bit of games $\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{B})$ and $\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{A})$ respectively. Let Bad be the event that in the ideal world of $\mathcal{B}$, there is a verification query $\textsc{Vf}(i, C)$ and then later there is an encryption query $C \leftarrow \textsc{Enc}(i, M)$. If Bad doesn't happen then $\mathcal{B}$ correctly simulates the ideal

world of $\mathcal{A}$ and has the same answer as $\mathcal{A}$, and thus

$$\Pr[\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{B}) \mid b = 0] \leq \Pr[\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{A}) \mid d = 0] + \Pr[\mathrm{Bad}] \ .$$

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

$$\sum_{j=0}^{|C|-\lambda} 2^j < 2^{|C|-\lambda+1}$$

encryption queries. However, the chance that one of those encryption queries can result in C is at most

$$2^{|C|-\lambda+1} \cdot \frac{1}{2^{|C|}} = \frac{2}{2^\lambda}.$$

Summing this over q_v verification queries,

$$\Pr[\mathrm{Bad}] \leq \frac{2q_v}{2^\lambda} \ .$$

Hence

$$\Pr[\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{B}) \mid b = 0] \leq \Pr[\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{A}) \mid d = 0] + \frac{2q_v}{2^\lambda} \ ,$$

and thus

$$\mathbf{Adv}^{\mathrm{ipf}}_{\mathsf{F}}(\mathcal{B}) \geq \mathbf{Adv}^{\mathrm{ipf}}_{\mathsf{F}}(\mathcal{A}) - \frac{2q_v}{2^\lambda} \ .$$

This concludes the proof. ▌

IPF SECURITY OF HtM FOR ORDERLY ATTACKERS. As a ciphertext of HtM is at least $(\tau + d)$-bit longer than its plaintext, we can restrict to orderly adversaries with a loss of $2q_v/2^{\tau+d} \leq 0.5q_v/2^\tau$ in the advantage. Thus, without loss of generality, assume that $\mathcal{A}$ is orderly.

Consider games G_1–G_6 in Fig. 13, Fig. 14, and Fig. 15. Game G_1 is essentially game $\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{A})$ with challenge bit 1, but for each verification query, we only check the tag. That is, if a verification query can pass the tag checking but end up with a bad padding, we will return true instead of false so that the adversary wins. This model the increase in the advantage due to a potential timing leakage. Game G_6 corresponds to game $\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{A})$ with challenge bit 0. Hence

$$\mathbf{Adv}^{\mathrm{ipf}}_{\mathsf{F}}(\mathcal{A}) \leq \Pr[G_1(\mathcal{A})] - \Pr[G_6(\mathcal{A})] \ .$$

We now explain the games. In game G_2, instead of using $E(K_i, \cdot)$, we lazily maintain a truly random function f_i via procedure $\mathsf{Map}(i, \cdot)$. To bound the gap between G_1 and G_2, we construct an adversary $\mathcal{B}$ attacking the (multi-user) PRF security of E as follows. Adversary $\mathcal{B}$ runs $\mathcal{A}$ and simulates game G_1. However, for each call to $E(K_i, \cdot)$, it instead queries the oracle $\textsc{Eval}(i, \cdot)$. Then

$$\begin{aligned} \Pr[G_1(\mathcal{A})] &= \Pr[\mathbf{G}^{\mathsf{prf}}_{E}(\mathcal{B}) \Rightarrow 1 \mid b = 1] \\ \Pr[G_2(\mathcal{A})] &= \Pr[\mathbf{G}^{\mathsf{prf}}_{E}(\mathcal{B}) \Rightarrow 0 \mid b = 0] \ , \end{aligned}$$

where b is the challenge bit of game $\mathbf{G}^{\mathsf{prf}}_{E}(\mathcal{B})$. Subtracting, we obtain

$$\Pr[G_1(\mathcal{A})] - \Pr[G_2(\mathcal{A})] = 2 \cdot \Pr[\mathbf{G}^{\mathsf{prf}}_{E}(\mathcal{B})] - 1 = \mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{B}) \ .$$

Next, in game G_3, we maintain a list S_i of padded plaintexts M that are queried for user i under encryption queries. Then for each verification query $\textsc{Vf}(i, C)$, if the tentative padded plaintext is in S_i then we immediately return false. We now argue that the input/output behavior of G_3 is the same as G_2. To see why, note that HtM is a tidy AE scheme, meaning that if we decrypt C to a plaintext P, then encrypting P must yield back the same C. In other words, if $\textsc{Vf}(i, C)$ produces a tentative $P \in S_i$ and still returns true, this means C must be a prior ciphertext from $\textsc{Enc}(i, P)$.

Game $G_1(\mathcal{A})$	Games $G_2(\mathcal{A})$
$v \leftarrow 0$; $b' \leftarrow \mathcal{A}^{\text{New},\text{Enc},\text{Dec}}$; Return b'	$v \leftarrow 0$; $b' \leftarrow \mathcal{A}^{\text{New},\text{Enc},\text{Dec}}$; Return b'
New()	New()
$v \leftarrow v+1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k$	$v \leftarrow v+1$
Enc(i,P)	Enc(i,P)
$M \leftarrow \mathsf{pad}(P,0)$	$M \leftarrow \mathsf{pad}(P,0)$
$T \leftarrow (E(K_i,M) \oplus M)[1:\tau]$	$T \leftarrow (\mathsf{Map}(i,M) \oplus M)[1:\tau]$
$Z \leftarrow \mathsf{pad}(T,1)$	$Z \leftarrow \mathsf{pad}(T,1)$
$C^* \leftarrow E(K_i,Z) \oplus Z \oplus M$	$C^* \leftarrow \mathsf{Map}(i,Z) \oplus Z \oplus M$
Return $C^* \| T$	Return $C^* \| T$
Vf$(i, C^* \| T)$	Vf$(i, C^* \| T)$
$Z \leftarrow \mathsf{pad}(T,1)$	$Z \leftarrow \mathsf{pad}(T,1)$
$M \leftarrow E(K_i,Z) \oplus Z \oplus C^*$	$M \leftarrow \mathsf{Map}(i,Z) \oplus Z \oplus C^*$
$V \leftarrow (E(K_i,M) \oplus M)[1:\tau]$	$V \leftarrow (\mathsf{Map}(i,M) \oplus M)[1:\tau]$
Return $(V = T)$	Return $(V = T)$
	$\mathsf{Map}(i,R)$
	If $\mathsf{Tbl}[i,R] = \perp$ then
	$\quad \mathsf{Tbl}[i,R] \leftarrow\!\!\$\ \{0,1\}^n$
	Return $\mathsf{Tbl}[i,R]$

Figure 13: Games G_1 and G_2 in the proof of Proposition 5.2.

This however violates the definitional restriction. Hence

$$\Pr[G_2(\mathcal{A})] = \Pr[G_3(\mathcal{A})] \ .$$

In game G_4, for every verification query, we simply return false. To bound the gap between G_3 and G_4, we claim that each verification query in G_3 can return true or $\perp$ with probability at most $1/2^\tau$. Summing over at most q_v queries yields

$$\Pr[G_3(\mathcal{A})] - \Pr[G_4(\mathcal{A})] \leq \frac{q_v}{2^\tau} \ .$$

To justify the claim above, consider a verification query $\text{Vf}(i,C)$ in game G_3. Since the adversary is orderly, without loss of generality, assume that this is the very first verification query. Assume further that this query produces a tentative padded plaintext $M \notin S_i$. Due to the domain separation in pad, this means that the game has never run $U \leftarrow \mathsf{Map}(i,M)$ before, and thus U is a uniformly random string, independent of C. Hence the chance that $(U \oplus M)[1:\tau]$ is the same as the tag in C is at most $1/2^\tau$, as claimed.

In game G_5, in each encryption query, we are supposed to call $\mathsf{Map}(i, \mathsf{pad}(P,0))$ and $\mathsf{Map}(i, \mathsf{pad}(T,1))$. For $\mathsf{Map}(i, \mathsf{pad}(P,0))$ we simply pick a fresh uniformly random answer but the answer of $\mathsf{Map}(i, \mathsf{pad}(T,1))$ is still consistent with prior calls to Map. Game G_5 is identical to game G_4, as the adversary must not repeat encryption queries and pad is a domain separation. Hence

$$\Pr[G_4(\mathcal{A})] = \Pr[G_5(\mathcal{A})] \ .$$

In game G_5, the answer of $\mathsf{Map}(i, \mathsf{pad}(T,1))$ is consistent with prior calls to Map. However, in game G_6, we instead pick a fresh uniformly random answer for each call. The two games are identical until the flag bad is set, and thus from the Fundamental Lemma of game playing [9],

$$\Pr[G_5(\mathcal{A})] - \Pr[G_6(\mathcal{A})] \leq \Pr[G_6(\mathcal{A}) \text{ set } \mathsf{bad}] \ .$$

Games $G_3(\mathcal{A})$	Game $G_4(\mathcal{A})$
$v \leftarrow 0$; $b' \leftarrow \mathcal{A}^{\text{New},\text{Enc},\text{Dec}}$; Return b'	$v \leftarrow 0$; $b' \leftarrow \mathcal{A}^{\text{New},\text{Enc},\text{Dec}}$; Return b'
$\underline{\text{New}()}$	$\underline{\text{New}()}$
$v \leftarrow v + 1$; $S_v \leftarrow \emptyset$	$v \leftarrow v + 1$
$\underline{\text{Enc}(i, P)}$	$\underline{\text{Enc}(i, P)}$
$M \leftarrow \mathsf{pad}(P, 0)$	$M \leftarrow \mathsf{pad}(P, 0)$
$T \leftarrow (\mathsf{Map}(i, M) \oplus M)[1 : \tau]$	$T \leftarrow (\mathsf{Map}(i, M) \oplus M)[1 : \tau]$
$S_i \leftarrow S_i \cup \{M\}$; $Z \leftarrow \mathsf{pad}(T, 1)$	$Z \leftarrow \mathsf{pad}(T, 1)$
$C^* \leftarrow \mathsf{Map}(i, Z) \oplus Z \oplus M$	$C^* \leftarrow \mathsf{Map}(i, Z) \oplus Z \oplus M$
Return $C^* \| T$	Return $C^* \| T$
$\underline{\text{Vf}(i, C^* \| T)}$	$\underline{\text{Vf}(i, C^* \| T)}$
$Z \leftarrow \mathsf{pad}(T, 1)$	Return false
$M \leftarrow \mathsf{Map}(i, Z) \oplus Z \oplus C^*$	
If $M \in S_i$ then return false	
$V \leftarrow (\mathsf{Map}(i, M) \oplus M)[1 : \tau]$	
Return $(V = T)$	
$\underline{\mathsf{Map}(i, R)}$	$\underline{\mathsf{Map}(i, R)}$
If $\mathsf{Tbl}[i, R] = \perp$ then	If $\mathsf{Tbl}[i, R] = \perp$ then
$\mathsf{Tbl}[i, R] \leftarrow\!\!\$\ \{0,1\}^n$	$\mathsf{Tbl}[i, R] \leftarrow\!\!\$\ \{0,1\}^n$
Return $\mathsf{Tbl}[i, R]$	Return $\mathsf{Tbl}[i, R]$

Figure 14: Games G_3 and G_4 in the proof of Proposition 5.2.

Games $\boxed{G_5(\mathcal{A})}$, $G_6(\mathcal{A})$	$\underline{\text{Vf}(i, C^* \| T)}$
$v \leftarrow 0$; $b' \leftarrow \mathcal{A}^{\text{New},\text{Enc},\text{Dec}}$; Return b'	Return false
$\underline{\text{New}()}$	$\underline{\mathsf{Map}(i, R)}$
$v \leftarrow v + 1$	$V \leftarrow\!\!\$\ \{0,1\}^n$
$\underline{\text{Enc}(i, P)}$	If $\mathsf{Tbl}[i, R] \neq \perp$ then
$M \leftarrow \mathsf{pad}(P, 0)$; $U \leftarrow\!\!\$\ \{0,1\}^n$	$\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{V \leftarrow \mathsf{Tbl}[i, R]}$
$T \leftarrow (U \oplus M)[1 : \tau]$	$\mathsf{Tbl}[i, R] \leftarrow V$; Return V
$Z \leftarrow \mathsf{pad}(T, 1)$	
$C^* \leftarrow \mathsf{Map}(i, Z) \oplus Z \oplus M$	
Return $C^* \| T$	

Figure 15: Games G_5 and G_6 in the proof of Proposition 5.2. Game G_5 contains the corresponding boxed statement, but game G_6 does not.

In game G_6, basically we pick $T \leftarrow\!\!\$\ \{0,1\}^\tau$ for each encryption query $\text{Enc}(i, \cdot)$ and set bad if this T hits one of the tags generated by prior encryption query to the same user i. As there are at most $B - 1$ prior queries per user, each encryption query sets bad with probability at most $(B - 1)/2^\tau$. Summing this over at most q_e queries,

$$\Pr[G_6(\mathcal{A}) \text{ set } \mathsf{bad}] \leq \frac{q_e(B-1)}{2^\tau} \ .$$

We now argue that game G_6 is identical to game $\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{A})$ with challenge bit 0, In game G_6, on

each encryption query (i, P), we first obtain M by padding P, and then samples $U \leftarrow\!\!\$\ \{0,1\}^n$ and computes $T \leftarrow (U \oplus M)[1:\tau]$. This means that effectively we sample $T \leftarrow\!\!\$\ \{0,1\}^\tau$. Next, since Map now always returns a fresh random answer $V \leftarrow\!\!\$\ \{0,1\}^n$, the code $C^* \leftarrow \mathsf{Map}(i, Z) \oplus Z \oplus M$ effectively means that we sample $C^* \leftarrow\!\!\$\ \{0,1\}^n$. Hence effectively, each encryption query returns a fresh uniformly random answer from $\{0,1\}^{2n}$.

Summing up

$$\mathbf{Adv}^{\mathrm{ipf}}_{\mathsf{F}}(\mathcal{A}) \;\leq\; \sum_{i=1}^{5} \Pr[G_i(\mathcal{A})] - \Pr[G_{i+1}(\mathcal{A})] \leq \mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{B}) + \frac{q_v + q_e(B-1)}{2^\tau} \;.$$

IPF SECURITY OF HtM FOR GENERAL ATTACKERS. If we account for the loss of $0.5q/2^\tau$ in the advantage by assuming that $\mathcal{A}$ is orderly, then for a general adversary $\mathcal{A}$,

$$\mathbf{Adv}^{\mathrm{ipf}}_{\mathsf{F}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{prf}}_{E}(\mathcal{B}) + \frac{1.5q_v + q_e(B-1)}{2^\tau} \;.$$

F Proof of Proposition 5.3

Consider the following sequence of games. Game G_0 corresponds to game $\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{A})$ with challenge bit $b = 1$. Game G_1 is similar to game G_1, but instead of using $E(K_i, \cdot)$, we use a truly random permutation π_i. To bound the gap between the two games, we construct an adversary $\mathcal{B}$ attacking E as follows. It simulates game G_0, but calls to $E(K_i, \cdot)$ are replaced by corresponding calls to $\textsc{Enc}(i, \cdot)$, and calls to $E^{-1}(K_i, \cdot)$ are replaced by corresponding calls to $\textsc{Dec}(i, \cdot)$. Then

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

Game G_2 is similar to game G_1, except that verification queries will always return false. 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. Assume that $\mathcal{A}$ doesn't make redundant queries: (i) it doesn't repeat prior queries, and (ii) if it queries $Y \leftarrow E(L, X)$ then it won't later query $E^{-1}(L, Y)$ and vice versa. Assume that the claimed bound is smaller than 1. Consider a verification query (i, C) in game G_1. Let $\mathsf{Dom} = \{\mathsf{pad}(X) \mid X \in \{0,1\}^{\leq n-s}\}$; note that $|\mathsf{Dom}| \leq 2^{n-s+1}$. Then the answer $M \leftarrow \pi_i^{-1}(C)$ is uniformly chosen from a set of at least $2^n - q \geq 2^{n-1}$ elements, and the chance that $M \in \mathsf{Dom}$ is at most $2^{n-s+1}/2^{n-1} = 4/2^s$. Hence the chance that this verification query returns true is at most $4/2^s$. Summing this over q verification queries,

$$\Pr[G_1(\mathcal{A})] - \Pr[G_2(\mathcal{A})] \leq \frac{4q}{2^s} \;.$$

In game G_3, we pick the answer for the encryption queries uniformly at random. From the multi-user PRP/PRF Switching Lemma [5],

$$\Pr[G_2(\mathcal{A})] - \Pr[G_3(\mathcal{A})] \leq \frac{qB}{2^n} \;.$$

Note that game G_3 corresponds to game $\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{A})$ with challenge bit $b = 1$. Hence

$$\mathbf{Adv}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{A}) = \Pr[G_0(\mathcal{A})] - \Pr[G_3(\mathcal{A})] \leq \mathbf{Adv}^{\pm\mathrm{prp}}_{E}(\mathcal{B}) + \frac{4q}{2^s} + \frac{qB}{2^n} \;.$$

This concludes the proof.

G Proof of Proposition 5.4

Since we consider computationally unbounded adversaries, without loss of generality, assume that $\mathcal{A}$ is deterministic. Assume that the adversary doesn't make redundant queries: (i) it doesn't repeat

prior queries, and (ii) if it queries $Y \leftarrow E(L, X)$ then it won't later query $E^{-1}(L, Y)$ and vice versa. Assume that the claimed bound is smaller than 1. Let $\mathsf{Dom} = \{\mathsf{pad}(X) \mid X \in \{0,1\}^{\leq n-s}\}$. Note that $|\mathsf{Dom}| \leq 2^{n-s+1}$.

Let Bad be the event that there is a backward query $M \leftarrow E^{-1}(L, C)$ such that $M \in \mathsf{Dom}$. We now bound Pr[Bad]. Note that for each query $M \leftarrow E^{-1}(L, C)$, the variable M is uniformly distributed over a set of at least $2^n - q \geq 2^{n-1}$ elements, and thus the chance that $M \in \mathsf{Dom}$ is at most $2^{n-s+1}/2^{n-1} = 4/2^s$. Summing this over q queries,

$$\Pr[\text{Bad}] \leq \frac{4q}{2^s} \ .$$

Assume that Bad doesn't happen. Suppose that $\mathcal{A}$ outputs (K_1, P_1, K_2, P_2). For each $i \in \{1, 2\}$, let $C_i \leftarrow E(K_i, \mathsf{pad}(P_i))$. We first consider the case that there is some $i \in \{1, 2\}$ such that there is no query $E(K_i, \mathsf{pad}(P_i))$. Without loss of generality, assume that we do not have such a query for (K_1, P_1). Then given C_2, the variable C_1 is uniformly distributed over a set of at least $2^n - q \geq 2^{n-1}$ elements, and thus the chance that $C_1 = C_2$ is at most $2/2^n$.

Next consider the case that for every $i \in \{1, 2\}$, the adversary does query $E(K_i, \mathsf{pad}(P_i))$. For each query $E(L, M)$, the chance that its output hits the answer of a prior query is at most $(q-1)/(2^n - q) \leq 2(q-1)/2^n$. Summing this over q queries, the chance that the adversary can find a collision is at most $2q(q-1)/2^n$.

Hence by the union bound, the chance that the adversary can break the collision resistance of PtE is at most

$$\frac{4q}{2^s} + \frac{2q(q-1)}{2^n} + \frac{2}{2^n} \leq \frac{4q}{2^s} + \frac{2q^2}{2^n}$$

as claimed.

H Proof of Proposition 6.1

We first construct adversary $\mathcal{B}$. It runs $\big((K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2)\big) \leftarrow\!\!\$\ \mathcal{A}$, and then parses $S_i \| P_i \leftarrow M_i$ with $|P_i| = \min\{m, |M_i|\}$ for every $i \in \{1, 2\}$. Finally, it outputs $\big((K_1, N_1, A_1, S_1), (K_2, N_2, A_2, S_2)\big)$.

We now construct $\mathcal{D}$. It runs $\big((K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2)\big) \leftarrow\!\!\$\ \mathcal{A}$, then parses $S_i \| P_i \leftarrow M_i$ with $|P_i| = \min\{m, |M_i|\}$ for every $i \in \{1, 2\}$, and then encrypts $C_i^* \| T_i \leftarrow \mathsf{SE.Enc}(K_i, N_i, A_i, S_i)$. Finally, it outputs (T_1, P_1, T_2, P_2).

To analyze the advantage of the adversaries, suppose that adversary $\mathcal{A}$ outputs $\big((K_1, N_1, A_1, M_1), (K_2, N_2, A_2, M_2)\big)$, and let $C_i \leftarrow \overline{\mathsf{SE}}.\mathsf{Enc}(K_i, N_i, A_i, M_i)$ and $C_i^* \| T_i \leftarrow \mathsf{SE.Enc}(K_i, N_i, A_i, S_i)$. Note that $C_i = C_i^* \| R_i$, where $R_i \leftarrow \mathsf{F}(T_i, P_i)$. Without loss of generality, suppose $(K_1, N_1, A_1) \neq (K_2, N_2, A_2)$. Indeed if $(K_1, N_1, A_1) = (K_2, N_2, A_2)$ and $C_1 = C_2$ then $M_1 = M_2$ due to the perfect correctness of $\overline{\mathsf{SE}}$, violating the definitional restriction. Then the output of $\mathcal{B}$ is legitimate. Since $C_1 = C_2$, we have $R_1 = R_2$ and $C_1^* = C_2^*$. If $\mathcal{D}$ breaks the collision resistance of F then we are done. Assume to the contrary that $\mathcal{D}$ does not break the collision resistance of F. Since $R_1 \leftarrow \mathsf{F}(T_1, P_1)$ and $R_2 \leftarrow \mathsf{F}(T_2, P_2)$ are the same, the only way $\mathcal{D}$ can't be deemed to break the collision resistance of F is when its output is invalid, meaning $(T_1, P_1) = (T_2, P_2)$. But then $C_1^* \| T_1 = C_2^* \| T_2$, meaning $\mathcal{B}$ breaks the committing security of SE.

I Proof of Theorem 6.3

Note that since $\overline{\mathsf{SE}}$ and SE share the same decryption-nonce derivation function, they have the same decryption-nonce density d. Since SE is tag-based, from Lemma 3.1, $d \le \mathsf{SE.ce}(m) - \mathsf{SE.tl} = \ell$ for every m. Thus $\overline{\mathsf{SE}}.\mathsf{ce}(m) \ge \ell + s \ge d + s$ for every m. From Lemma 3.2, without loss of generality, assume that $\mathcal{A}$ is orderly. This assumption creates a difference of at most $2q_v/2^s$, which we will account later. Assume that the adversary doesn't make forbidden queries. Let λ be the output length of H.

Consider games G_1–G_8 in Fig. 16–Fig. 18. Game G_1 is essentially game $\mathbf{G}^{\mathrm{real}}_{\overline{\mathsf{SE}}}(\mathcal{A})$, but for verification queries, we only check if the decrypted prefix P is $\perp$, but drop the check $(|C^*| \neq \ell) \wedge (|P| \neq m)$. This corresponds to an increase in the advantage due to timing leakage of the two checks, and that should only help the adversary. For clarity, $\mathcal{A}$ is given an interface $\textsc{Ro}$ to the random oracle, and $\overline{\mathsf{SE}}.\mathsf{Enc}$ and $\overline{\mathsf{SE}}.\mathsf{Dec}$ are implemented using another interface $\overline{\textsc{Ro}}$. Game G_8 corresponds to game $\mathbf{G}^{\mathrm{rand}}_{\overline{\mathsf{SE}}}(\mathcal{A})$. Hence

$$\mathbf{Adv}^{\mathrm{ae}}_{\overline{\mathsf{SE}}}(\mathcal{A}) \le \Pr[G_1(\mathcal{A})] - \Pr[G_8(\mathcal{A})] \ .$$

We now describe the game chain. In game G_2, we maintain a set Dom that is initialized to $\emptyset$. This set keeps track of the keys $K_1, \ldots, K_v$ so far, and also the components K in random-oracle queries $\textsc{Ro}(K, N, V)$ of $\mathcal{A}$. Thus $|\mathsf{Dom}| \le p + u$. In game G_2, each time when we initialize a user i and sample its key, if the key falls within Dom then we set bad to true, and re-sample the key uniformly at random from $\{0,1\}^k \backslash \mathsf{Dom}$. Games G_1 and G_2 are identical until bad, and thus from the Fundamental Lemma of Game Playing [9],

$$\Pr[G_1(\mathcal{A})] - \Pr[G_2(\mathcal{A})] \le \Pr[G_2(\mathcal{A}) \text{ sets } \mathsf{bad}] \le \frac{u(u+p)}{2^k} \ .$$

Game G_3 is a different way to implement G_2. Instead of using the same underlying table Tbl for both $\textsc{Ro}$ and $\overline{\textsc{Ro}}$, we use two tables Tbl and $\overline{\mathsf{Tbl}}$, where calls to $\overline{\textsc{Ro}}$ are implemented in $\overline{\mathsf{Tbl}}$. For calls (K, N, V) to $\textsc{Ro}$, generally they are implemented via Tbl, but if K somehow falls within the current set of keys $\{K_1, \ldots, K_v\}$ then we implement it by calling $\overline{\textsc{Ro}}(K, N, V)$. Hence

$$\Pr[G_2(\mathcal{A})] = \Pr[G_3(\mathcal{A})] \ .$$

In game G_4, for each call $\textsc{Ro}(K, N, V)$, if $K \in \{K_1, \ldots, K_v\}$ then we implement it via Tbl (meaning $\textsc{Ro}$ and $\overline{\textsc{Ro}}$ are now independent) and set bad to true. The game will return true if bad is set. For $i \in \{3, 4\}$ and $b \in \{\mathsf{true}, \mathsf{false}\}$, let $G_i(\mathcal{A}, b)$ denote the event that $G_i(\mathcal{A})$ returns true and $\mathsf{bad} = b$. Note that G_4 and G_3 are identical until bad is set. Then

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

Moreover,

$$\begin{aligned} \Pr[G_3(\mathcal{A}, \mathsf{true})] &\le \Pr[G_3(\mathcal{A}) \text{ sets } \mathsf{bad}] \\ &= \Pr[G_4(\mathcal{A}) \text{ sets } \mathsf{bad}] = \Pr[G_4(\mathcal{A}, \mathsf{true})] \ . \end{aligned}$$

Hence

$$\begin{aligned} \Pr[G_3(\mathcal{A})] &= \Pr[G_3(\mathcal{A}, \mathsf{false})] + \Pr[G_3(\mathcal{A}, \mathsf{true})] \\ &\le \Pr[G_4(\mathcal{A}, \mathsf{false})] + \Pr[G_4(\mathcal{A}, \mathsf{true})] = \Pr[G_4(\mathcal{A})] \ . \end{aligned}$$

In game G_5, for each encryption query, instead of using F, we simply sample the ciphertext X at random. Likewise, for each verification query $\textsc{Vf}(i, N, A, C^* \| X)$, instead of calling $\overline{\textsc{Ro}}(K_i, N, A \| R)$ and then using $\mathsf{F.Dec}$, we check if there is a prior encryption query on (i, N, A) that also has the same internal tag R and returns the same X. We return true if this encryption query exists.

Games $G_1(\mathcal{A})$, $G_2(\mathcal{A})$

$v \leftarrow 0$; $\mathsf{Dom} \leftarrow \emptyset$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Enc},\text{Vf},\text{Ro}}$
Return $(b' = 1)$

New()
$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k$
If $K_v \in \mathsf{Dom}$ then
  $\mathsf{bad} \leftarrow \mathsf{true}$; $K_v \leftarrow\!\!\$\ \{0,1\}^k \backslash \mathsf{Dom}$
$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{K_v\}$

Enc(i, N, A, M)
$S \| P \leftarrow M$
$C^* \| R \leftarrow \mathsf{SE.Enc}(K_i, N, \varepsilon, S)$
$T \leftarrow \overline{\text{Ro}}(K_i, N, A \| R)$
$X \leftarrow \mathsf{F}(T, P)$
Return $C^* \| X$

Vf$(i, D, A, C^* \| X)$
$(S, N, R) \leftarrow \mathsf{SE.Tag}(K_i, D, \varepsilon, C^*)$
$T \leftarrow \overline{\text{Ro}}(K_i, N, A \| R)$
$P \leftarrow \mathsf{F}^{-1}(T, X)$; Return $(P \neq \bot)$

Ro(K, N, V)
If $\mathsf{Tbl}[K, N, V] = \bot$ then
  $\mathsf{Tbl}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{K\}$
Return $\mathsf{Tbl}[K, N, V]$

$\overline{\text{Ro}}(K, N, V)$
If $\mathsf{Tbl}[K, N, V] = \bot$ then
  $\mathsf{Tbl}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
Return $\mathsf{Tbl}[K, N, V]$

Games $G_3(\mathcal{A})$, $G_4(\mathcal{A})$

$v \leftarrow 0$; $\mathsf{win} \leftarrow \mathsf{false}$
$\mathsf{Dom} \leftarrow \emptyset$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Enc},\text{Vf},\text{Ro}}$
$\mathsf{win} \leftarrow \mathsf{bad}$; $\mathsf{win} \leftarrow \mathsf{false}$
Return $(b' = 1) \vee \mathsf{win}$

New()
$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k \backslash \mathsf{Dom}$
$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{K_v\}$

Enc(i, N, A, M)
$S \| P \leftarrow M$
$C^* \| R \leftarrow \mathsf{SE.Enc}(K_i, N, \varepsilon, S)$
$T \leftarrow \overline{\text{Ro}}(K_i, N, A \| R)$
$X \leftarrow \mathsf{F}(T, P)$
Return $C^* \| X$

Vf$(i, D, A, C^* \| X)$
$(S, N, R) \leftarrow \mathsf{SE.Tag}(K_i, D, \varepsilon, C^*)$
$T \leftarrow \overline{\text{Ro}}(K_i, N, A \| R)$
$P \leftarrow \mathsf{F}^{-1}(T, X)$; Return $(P \neq \bot)$

Ro(K, N, V)
If $K \in \{K_1, \ldots, K_v\}$ then
  $\mathsf{bad} \leftarrow \mathsf{true}$; return $\overline{\text{Ro}}(K, N, V)$
If $\mathsf{Tbl}[K, N, V] = \bot$ then
  $\mathsf{Tbl}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{K\}$
Return $\mathsf{Tbl}[K, N, V]$

$\overline{\text{Ro}}(K, N, V)$
If $\overline{\mathsf{Tbl}}[K, N, V] = \bot$ then
  $\overline{\mathsf{Tbl}}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
Return $\overline{\mathsf{Tbl}}[K, N, V]$

Figure 16: Games G_1–G_4 in the proof of Theorem 6.3. Games G_2 and G_3 contain the corresponding boxed statements, but game G_1 and G_4 do not.

To bound the gap between G_4 and G_5, we construct an adversary $\mathcal{B}$ attacking the IPF security of F as follows. Adversary $\mathcal{B}$ runs $\mathcal{A}$ and simulates game G_4. However, for each encryption query of $\mathcal{A}$, instead of calling $T \leftarrow \overline{\text{Ro}}(K_i, N, A \| R)$ and computing $X \leftarrow \mathsf{F}(T, P)$, it makes an oracle call Enc(u, P), where $u = (i, N, A \| R)$.[4] Note that $\mathcal{B}$ never repeats the same query (u, P) to Enc. Likewise, for each verification query of $\mathcal{A}$, instead of calling $T \leftarrow \overline{\text{Ro}}(K_i, N, A \| R)$ and checking if $\mathsf{F}^{-1}(T, X) \neq \bot$, if there is a prior call $X \leftarrow \text{Enc}(u, P)$, with $u = (i, N, A \| R)$, then $\mathcal{B}$ simply returns true. Otherwise it returns Vf(u, X). Finally, if win is set then $\mathcal{B}$ returns 1. Otherwise, it gives the

[4]Implicitly, $\mathcal{B}$ has to lazily maintain a map to turn a string tuple (i, N, V) into a number u.

Game $G_5(\mathcal{A})$

$v \leftarrow 0$; $\mathsf{win} \leftarrow \mathsf{false}$
$\mathsf{Dom} \leftarrow \emptyset$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Vf},\textsc{Ro}}$
Return $(b' = 1) \vee \mathsf{win}$

$\textsc{New}()$
$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k \backslash \mathsf{Dom}$
$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{K_v\}$

$\textsc{Enc}(i, N, A, M)$
$S \| P \leftarrow M$
$C^* \| R \leftarrow \mathsf{SE.Enc}(K_i, N, \varepsilon, S)$
$X \leftarrow\!\!\$\ \{0,1\}^{m+s}$
$\mathsf{Map}[i, N, A \| R, X] \leftarrow P$
Return $C^* \| X$

$\textsc{Vf}(i, D, A, C^* \| X)$
$(S, N, R) \leftarrow \mathsf{SE.Tag}(K_i, D, \varepsilon, C^*)$
Return $(\mathsf{Map}[i, N, A \| R, X] \neq \bot)$

$\textsc{Ro}(K, N, V)$
If $K \in \{K_1, \ldots, K_v\}$ then $\mathsf{win} \leftarrow \mathsf{true}$
If $\mathsf{Tbl}[K, N, V] = \bot$ then
  $\mathsf{Tbl}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{K\}$
Return $\mathsf{Tbl}[K, N, V]$

Game $G_6(\mathcal{A})$

$v \leftarrow 0$; $\mathsf{win} \leftarrow \mathsf{false}$
$\mathsf{Dom} \leftarrow \emptyset$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Vf},\textsc{Ro}}$
Return $(b' = 1) \vee \mathsf{win}$

$\textsc{New}()$
$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k \backslash \mathsf{Dom}$
$\mathsf{Dom} \leftarrow \mathsf{Dom} \cup \{K_v\}$

$\textsc{Enc}(i, N, A, M)$
$S \| P \leftarrow M$
$C^* \| R \leftarrow \mathsf{SE.Enc}(K_i, N, \varepsilon, S)$
$X \leftarrow\!\!\$\ \{0,1\}^{m+s}$
$\mathsf{Map}[i, N, A \| R, X] \leftarrow P$
Return $C^* \| X$

$\textsc{Vf}(i, D, A, C^* \| X)$
For any (N, R) with $\mathsf{SE.df}(N) = D$ do
  If $\mathsf{Map}[i, N, A \| R, X] \neq \bot$ then
    Return $(\mathsf{SE.Dec}(K_i, D, \varepsilon, C^* \| R) \neq \bot)$
Return false

$\textsc{Ro}(K, N, V)$
If $K \in \{K_1, \ldots, K_v\}$ then $\mathsf{win} \leftarrow \mathsf{true}$
If $\mathsf{Tbl}[K, N, V] = \bot$ then
  $\mathsf{Tbl}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
Return $\mathsf{Tbl}[K, N, V]$

Figure 17: Games G_5 and G_6 in the proof of Theorem 6.3.

same answer as $\mathcal{A}$. Then

$$\begin{aligned} \Pr[\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{B}) \mid b = 1] &= \Pr[G_4(\mathcal{A})] \ , \text{ and} \\ 1 - \Pr[\mathbf{G}^{\mathsf{ipf}}_{\mathsf{F}}(\mathcal{B}) \mid b = 0] &= \Pr[G_5(\mathcal{A})] \ . \end{aligned}$$

Subtracting, we obtain

$$\mathbf{Adv}^{\mathrm{ipf}}_{\mathsf{F}}(\mathcal{B}) = \Pr[G_4(\mathcal{A})] - \Pr[G_5(\mathcal{A})] \ .$$

In game G_6, for each verification query $(i, D, A, C^* \| X)$, we try to find a prior encryption query (i, N, A, M) that returns the same X and $\mathsf{SE.df}(N) = D$. If there is such an encryption query, we return $(\mathsf{SE.Dec}(K_i, D, \varepsilon, C^* \| R) \neq \bot)$, where R is the internal tag of this encryption query. Otherwise, return false.

We now bound the gap between game G_5 and G_6. Without loss of generality, assume that $\mathcal{A}$ returns 1 if some verification query returns true. Let Bad be the event that in game G_6, there are two encryption queries (i, N, A, M) and (i, N^*, A, M^*) that returns the same X. Then $\Pr[\mathrm{Bad}] \leq q^2/2^{m+s}$, because each encryption query returns a fresh random $X \leftarrow\!\!\$\ \{0,1\}^{m+s}$. Suppose that Bad doesn't happen. Note that the two games have the same implementation of the encryption queries. Consider a verification query $(i, D, A, C^* \| X)$. Suppose that $(S, N, R) \leftarrow \mathsf{SE.Tag}(K_i, D, \varepsilon, C^*)$. If there is an encryption query (i, N, A, M^*) of the same internal tag R that returns the same X then G_6 is simply a different way to implement G_5, using $\mathsf{SE.Dec}$ instead of $\mathsf{SE.Tag}$. Suppose that

Game $G_7(\mathcal{A})$

$v \leftarrow 0$; $\mathsf{win} \leftarrow \mathsf{false}$
$\mathsf{Dom} \leftarrow \emptyset$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Vf},\textsc{Ro}}$
Return $(b' = 1) \vee \mathsf{win}$

$\textsc{New}()$

$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k$

$\textsc{Enc}(i, N, A, M)$

$S \| P \leftarrow M$
$C^* \| R \leftarrow \mathsf{SE.Enc}(K_i, N, \varepsilon, S)$
$X \leftarrow\!\!\$\ \{0,1\}^{m+s}$
$\mathsf{Map}[i, N, A \| R, X] \leftarrow P$
Return $C^* \| X$

$\textsc{Vf}(i, D, A, C^* \| X)$

For any (N, R) with $\mathsf{SE.df}(N) = D$ do
  If $\mathsf{Map}[i, N, A \| R, X] \neq \perp$ then
    Return $(\mathsf{SE.Dec}(K_i, D, \varepsilon, C^* \| R) \neq \perp)$
Return false

$\textsc{Ro}(K, N, V)$

If $K \in \{K_1, \ldots, K_v\}$ then $\mathsf{win} \leftarrow \mathsf{true}$
If $\mathsf{Tbl}[K, N, V] = \perp$ then
  $\mathsf{Tbl}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
Return $\mathsf{Tbl}[K, N, V]$

Game $G_8(\mathcal{A})$

$v \leftarrow 0$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Vf},\textsc{Ro}}$
Return $(b' = 1)$

$\textsc{New}()$

$v \leftarrow v + 1$

$\textsc{Enc}(i, N, A, M)$

$S \| P \leftarrow M$;
$C^* \leftarrow\!\!\$\ \{0,1\}^{\mathsf{SE.cl}(|S|) - \mathsf{SE.tl}}$
$X \leftarrow\!\!\$\ \{0,1\}^{m+s}$
Return $C^* \| X$

$\textsc{Vf}(i, D, A, C^* \| X)$

Return false

$\textsc{Ro}(K, N, V)$

If $\mathsf{Tbl}[K, N, V] = \perp$ then
  $\mathsf{Tbl}[K, N, V] \leftarrow\!\!\$\ \{0,1\}^\lambda$
Return $\mathsf{Tbl}[K, N, V]$

Figure 18: Games G_7 and G_8 in the proof of Theorem 6.3.

there is no such encryption query. Without loss of generality, assume that G_6 returns false for this verification query. This can only decrease $\Pr[G_6(\mathcal{A})]$ and thus increase $\Pr[G_5(\mathcal{A})] - \Pr[G_6(\mathcal{A})]$. But then G_5 also returns false in this case. Hence

$$\Pr[G_5(\mathcal{A})] - \Pr[G_6(\mathcal{A})] \leq \Pr[\mathrm{Bad}] \leq \frac{q^2}{2^{m+s}} \ .$$

In game G_7, we sample the keys uniformly at random. Thus

$$\Pr[G_6(\mathcal{A})] - \Pr[G_7(\mathcal{A})] \leq \frac{u(u+p)}{2^k} \ .$$

In game G_8, for each encryption query, instead of using $\mathsf{SE.Enc}$ we simply pick a random ciphertext. Likewise, for each verification query, instead of using $\mathsf{SE.Dec}$, we simply return false.

To bound the gap between G_7 and G_8, we construct an adversary $\mathcal{D}$ attacking the AE security of SE as follows. It first initializes $\mathsf{Nonces}, \mathsf{Keys} \leftarrow \emptyset$. It then runs $\mathcal{A}$ and simulates game G_6, but with the following differences. For each encryption query (i, N, A, M), instead of using $\mathsf{SE.Enc}(K_i, N, \varepsilon, S)$, it uses $\textsc{Enc}(i, N, \varepsilon, S)$ and adds N to Nonces. For each verification query $(i, D, A, C^* \| X)$, instead of checking $\mathsf{SE.Dec}(K_i, D, \varepsilon, C^* \| R) \neq \perp$, it runs $\textsc{Vf}(i, N, \varepsilon, C^* \| R)$. For each random-oracle query $\textsc{Ro}(K, N, V)$, instead of checking if $K \in \{K_1, \ldots, K_v\}$, it simply adds K to Keys. Finally, when $\mathcal{A}$ terminates, adversary $\mathcal{D}$ picks $N^* \in \mathcal{N} \backslash \mathsf{Nonces}$, and computes $C'_i \leftarrow \textsc{Enc}(i, N^*, \varepsilon, 0^k)$ for every user i. Then, for each $K \in \mathsf{Keys}$, it computes $C \leftarrow \mathsf{SE.Enc}(K, N^*, \varepsilon, 0^k)$,

and sets $\mathsf{win} \leftarrow \mathsf{true}$ if there is some $C'_i = C$. Then

$$\Pr[G_7(\mathcal{A})] \leq \Pr[\mathbf{G}^{\mathrm{real}}_{\overline{\mathsf{SE}}}(\mathcal{D})] \ .$$

Here we have to use an inequality, because there might be false positives in checking $C'_i = C$ where the string K is not the key K_i. On the other hand,

$$\Pr[G_8(\mathcal{A})] \geq \Pr[\mathbf{G}^{\mathrm{rand}}_{\overline{\mathsf{SE}}}(\mathcal{D})] - \frac{up}{2^{k+\mathsf{SE.tl}}} \ ,$$

because each C'_i is uniformly chosen from $\{0,1\}^{k+\mathsf{SE.tl}}$, independent of what $\mathcal{A}$ receives. Subtracting, we obtain

$$\Pr[G_7(\mathcal{A})] - \Pr[G_8(\mathcal{A})] \ \leq \ \mathbf{Adv}^{\mathrm{ae}}_{\overline{\mathsf{SE}}}(\mathcal{D}) + \frac{up}{2^{k+\mathsf{SE.tl}}} \leq \mathbf{Adv}^{\mathrm{ae}}_{\overline{\mathsf{SE}}}(\mathcal{D}) + \frac{up}{2^k} \ .$$

Summing up,

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{ae}}_{\overline{\mathsf{SE}}}(\mathcal{A}) &= \Pr[G_1(\mathcal{A})] - \Pr[G_8(\mathcal{A})] \\
&= \sum_{i=1}^{7} \Pr[G_i(\mathcal{A})] - \Pr[G_{i+1}(\mathcal{A})] \\
&\leq \mathbf{Adv}^{\mathrm{ipf}}_{\mathsf{F}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{ae}}_{\overline{\mathsf{SE}}}(\mathcal{D}) + \frac{q^2}{2^{m+s}} + \frac{u(2u+3p)}{2^k} \ .
\end{aligned}$$

By accounting for the loss of $2q_v/2^s$ in the advantage by assuming that $\mathcal{A}$ is orderly,

$$\mathbf{Adv}^{\mathrm{ae}}_{\overline{\mathsf{SE}}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{ipf}}_{\mathsf{F}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{ae}}_{\overline{\mathsf{SE}}}(\mathcal{D}) + \frac{q^2}{2^{m+s}} + \frac{u(2u+3p)}{2^k} + \frac{2q_v}{2^s} \ .$$

This concludes the proof.