Universal Context Commitment without Ciphertext Expansion

Arghya Bhattacharjee[1,2] Ritam Bhaumik[2,3] Chandranan Dhar[1]

[1] Indian Statistical Institute, Kolkata, India
[2] Technology Innovation Institute, Abu Dhabi, UAE
[3] EPFL, Lausanne, Switzerland
{bhattacharjeearghya29, bhaumik.ritam, chandranandhar}@gmail.com

Abstract

An ongoing research challenge in symmetric cryptography is to design an authenticated encryption (AE) with a commitment to the secret key or preferably to the entire context. One way to achieve this is to use a transform on an existing AE scheme, if possible with no output length expansion. At EUROCRYPT'22, Bellare and Hoang proposed the HtE transform, which lifts key-commitment to context-commitment. In the same year at ESORICS'22, Chan and Rogaway proposed the CTX transform, which works on any AE scheme where the tag is not required for decryption. However, for AE schemes which are not key-committing to begin with and which use the tag for decryption, no such transform exists till date. The latter category encompasses all AE schemes based on the design paradigms SIV, MAC-then-Encrypt, and Encode-then-Encipher. In this work, we propose PACT, a transform to convert any AE scheme into a context-committing one without any output length expansion. In addition, PACT preserves both nonce-respecting and nonce-misuse security of the legacy AE scheme. However, this is not the case with all the existing transforms. To demonstrate this, we show that a combination of CTY and SC (proposed by Bellare and Hoang, CRYPTO'24) doesn't preserve the nonce-misuse security of the legacy AE scheme. PACT requires only one call to a collision-resistant unkeyed hash function and one call to a block cipher. Finally, we propose a lighter transform comPACT, which converts a nonce-respecting AE scheme into a context-committing one.

1 Introduction

Authenticated Encryption (AE) plays a pivotal role in modern symmetric cryptography by facilitating both encryption and authentication of plaintext. Frequently, AE schemes also offer the ability to authenticate additional data (called Associated Data), which is transmitted without encryption. These schemes are known as Authenticated Encryption with Associated Data (AEAD). Many contemporary AE schemes predominantly rely on nonces [Rog04], which in its most basic form require users to provide a unique nonce for each plaintext. Such schemes are known as Unique Nonce AE (UNAE). In contrast, Deterministic Authenticated Encryption (DAE) [RS06] is used where leveraging existing entropy or redundancy in the input makes more sense, thereby avoiding the overhead of nonces. This is particularly advantageous, for instance, when encrypting cryptographic keys.

Over time, the comprehension of AE security has undergone several revisions. Misuse-Resistant AE (MRAE) [RS06] was introduced to ensure that nonce repetitions do not compromise the security of the scheme, provided that a combination of nonce, associated data, and message values is not repeated. Subsequently, AE security was broadened to include scenarios where unverified plaintexts are released [ABL+14]. Further developments led to the notion of robust AE [HKR15] which allows the user to specify the ciphertext expansion, i.e., how much longer the ciphertext is compared to the plaintext. In recent years, a diverse array of leakage-resilient AE notions [PSV15, BMOS17, BPPS17] have spawned. More

recent works have emphasised that nonce and associated data present in the plaintext could potentially divulge crucial information, such as identifying sessions and users, thereby advocating for anonymous AE [CR19] and AE2 [BNT19].

Despite the existence of AE schemes with desirable features nonetheless, the primary focus has consistently centered on ensuring confidentiality and authenticity across different scenarios. Both the CAESAR competition for authenticated encryption [CAE19] and the recent NIST lightweight cryptography (LWC) standardisation process [NIS18] underscored the importance of AE schemes that guarantee security by delivering both confidentiality and authenticity.

However, a recent wave of attacks has demonstrated the necessity for a reevaluation of our conception of a secure AE scheme. The Facebook message franking attack [DGRW18] show how to break Facebook's message franking scheme, which means a malicious user can send an objectionable image to a recipient but that recipient cannot report it as abuse. Vulnerabilities have also been observed in "Subscribe with Google" [ADG+22] and Shadowsocks proxy servers [LGR21]. In the latter, adversaries can build a practical partitioning oracle attack that quickly recovers passwords from the servers. Indeed, all these attacks can be attributed to a common issue: the presence of ciphertexts that decrypt correctly under multiple keys. This vulnerability persists despite efforts to ensure confidentiality and authenticity, underscoring the necessity for an additional security notion.

Committing Authenticated Encryption. In pursuit of this goal, the notion of commitment security [BH22] was introduced, necessitating that every ciphertext serves as a commitment to the key (which we call CMT_k security) or even to the entire context (which we call CMT security, following the trend in concurrent work like [BCC+24] and [BH24]). The latter notion represents the strongest form and is formalised through the following security game: The adversary provides two tuples (K, N, A, M), (K', N', A', M'), each comprising a key, a nonce, an associated data, and a message. The adversary wins if their contexts differ, (i.e., $(K, N, A) \neq (K', N', A')$) and $\Pi.\mathsf{Enc}(K, N, A, M) = \Pi.\mathsf{Enc}(K', N', A', M')$ holds, where Π denotes the scheme under scrutiny. If we relax the condition of different contexts to different keys, it becomes the CMT_k security game.

The aforementioned attacks underscore the serious repercussions of employing non-committing authenticated encryption. Given the likelihood of additional undiscovered attacks, addressing this issue becomes imperative. In this regard, it is essential to subject AE schemes used in practice to scrutiny regarding commitment security. This evaluation process has already commenced, with several commonly used AE schemes (such as GCM, ChaCha20-Poly1305, SIV, CCM, EAX, OCB3) undergoing examination [BH22, MLGR23, ADG+22]. Regrettably, a majority of these schemes have been found lacking in achieving commitment security.

To attain commitment security, a viable strategy entails constructing authenticated encryption schemes from the ground up, ensuring they inherently offer commitment security. This approach alleviates worries about commitment attacks when these schemes are deployed in diverse protocols. Nonetheless, a more efficient method involves devising techniques to ensure commitment security of the existing AE schemes (which we call legacy AE schemes in this work), preferably meeting the criteria of CMT security. While some of these techniques are AE-specific, others are applicable to a class of AE schemes. We call a technique of this later category a *generic AE transform*. A generic AE transform is a technique which can be combined with any existing AE scheme, and results in a new AE scheme.

Ensuring the CMT security of Legacy AE Schemes. Several generic AE transforms for converting non-committing AE schemes into committing AE schemes have been suggested. Initially, these transforms focused solely on achieving CMT_k security, but our interest lies in transforms that achieve CMT security. Bellare and Hoang [BH22] introduce the first generic transform, known as HtE (Hash-then-Encrypt), which

$\mathsf{HtE}[\Pi, \mathcal{H}].\mathsf{Enc}(K, N, A, M)$

1: $K^* \leftarrow \mathcal{H}(K, N, A)$
2: $(C, T) \leftarrow \Pi.\mathsf{Enc}(K^*, N, \epsilon, M)$
3: **return** (C, T)

$\mathsf{HtE}[\Pi, \mathcal{H}].\mathsf{Dec}(K, N, A, C, T^*)$

1: $K^* \leftarrow \mathcal{H}(K, N, A)$
2: $X \leftarrow \Pi.\mathsf{Dec}(K^*, N, \epsilon, C, T)$
3: **return** X

Figure 1: Specification of HtE. HtE.Enc and HtE.Dec are the encryption and decryption algorithms of HtE respectively. Π is the input AE scheme, and Π.Enc and Π.Dec are its encryption and decryption algorithms respectively. K, N, A, M, C, and T denote the key, the nonce, the associated data, the message, the ciphertext and the tag, respectively. $\mathcal{H}$ is a hash function whose output-size is equal to the key-size of Π.

converts a CMT_k-secure scheme into a CMT-secure scheme. Fig. 1 gives the complete specification of HtE. HtE imposes no ciphertext expansion, and preserves both UNAE and MRAE security. While this serves as a promising starting point, HtE only works for AE schemes that are already CMT_k-secure. To address this, the same paper introduces two transforms: UtC (UNAE-then-Commit) and RtC (MRAE-then-Commit), which transform non-committing UNAE and MRAE schemes, respectively, into CMT_k-secure schemes. However, both of these transforms result in ciphertext expansion. Consequently, any non-committing nonce-based AE scheme can be transformed into a CMT-secure AE scheme using either HtE∘UtC or HtE∘RtC, but both schemes entail ciphertext expansion. Also note that both these variants provide $t/2$-bit CMT security, where t is the tag size of the legacy AE scheme.

As a separate contribution, the authors also suggest modifications for GCM and AES-GCM-SIV to make them CMT_k-secure. Specifically, they introduce CAU and CAU-SIV as generalisations of GCM and AES-GCM-SIV respectively, and propose adjustments to yield CMT_k-secure variants CAU-C1 and CAU-SIV-C1 respectively. Unlike UtC and RtC, these modifications do not lead to any ciphertext expansion, allowing the application of HtE to obtain CMT-secure schemes without ciphertext expansion.

Chan and Rogaway [CR22] introduce the CTX transform, which converts an AE scheme Π into a CMT-secure AE scheme CTX[Π], provided that the encryption algorithm of Π can be decomposed into two independent algorithms Π.Enc and Π.Auth, where Π.Enc produces the ciphertext C and Π.Auth generates the tag T. Fig. 2 gives the complete specification of CTX. While the authors argue that commonly used UNAE schemes like GCM and OCB meet these structural requirements, most frequently employed MRAE schemes such as SIV and its variants like GCM-SIV and NSIV, schemes employing the Mac-then-Encrypt paradigm or the Encode-then-Encipher paradigm, and several DAE schemes fall beyond the scope of CTX. In short, CTX is not applicable to any AE in which the decryption algorithm is dependent on the tag. We call such schemes TDD (Tag Dependent Decryption based) schemes. CTX also provides $t/2$-bit CMT security.

Naito et al. [NSS24] introduce a CMT transform labelled KIVR, which avoids ciphertext expansion by leveraging redundancy inherent in the plaintext. While this transform could prove advantageous for protocols where redundancy in plaintexts is commonplace, it deviates from assumptions typically encountered in classical settings.

Recently, in a concurrent work, Bellare et al. [BH24] have introduced the CTY transform as a faster alternative to the CTX transform. Unlike CTX, which processes the associated data twice—once for encryption and once for hashing, the CTY transform demonstrates that using the associated data only for hashing is sufficient to achieve same CMT security as CTX. In the same paper, the authors argue that since CMT attacks are offline, having approximately 128 bits of CMT security is advantageous. Given that most AE schemes employ 128-bit tags, HtE, CTX, and CTY provide 64 bits of CMT security. To address this limitation, the authors suggest that the output of the CTY transform does not need to be output-length-preserving, leading them to propose a technique called SC, which reduces the output-size

$\mathsf{CTX}[\Pi, \mathcal{H}].\mathsf{Enc}(K, N, A, M)$

1: $C \leftarrow \Pi.\mathsf{Enc}(K, N, A, M)$
2: $T \leftarrow \Pi.\mathsf{Auth}(K, N, A, M)$
3: $T^* \leftarrow \mathcal{H}(K, N, A, T)$
4: **return** (C, T^*)

$\mathsf{CTX}[\Pi, \mathcal{H}].\mathsf{Dec}(K, N, A, C, T^*)$

1: $M \leftarrow \Pi.\mathsf{Dec}(K, N, A, C)$
2: $T \leftarrow \Pi.\mathsf{Auth}(K, N, A, M)$
3: **If** $T^* \neq \mathcal{H}(K, N, A, T)$
4: **return** $\perp$
5: **return** M

Figure 2: Specification of CTX. CTX.Enc and CTX.Dec are the encryption and decryption algorithms of CTX respectively. Π is the input AE scheme. Π.Enc and Π.Auth are the algorithms of Π to generate the ciphertext and the tag respectively; and Π.Dec is the decryption algorithm of Π. K, N, A, M, C, and T denote the key, the nonce, the associated data, the message, the ciphertext and the tag, respectively. $\mathcal{H}$ is a hash function whose output-size is equal to the tag-size.

of CTY with minimal CMT security loss. They propose that if the tag-size of the legacy AE is t bits, the output tag-size of CTY can be $2t$ bits, achieving t bits of CMT security, and then SC can shorten the ciphertext-size to t bits, while (almost) preserving the t-bit CMT security. Although this is an improvement over CTX, both CTX and SC $\circ$ CTY are limited to the same class of AEs, excluding TDD schemes from their applicability.

Challenges. We observe that numerous AE schemes are incompatible with CTX. For instance, schemes following the SIV paradigm or the Mac-then-Encrypt paradigm utilise the tag for encryption, rendering the decryption function dependent on the tag, thus precluding the application of CTX. Similarly, AE schemes adhering to the Encode-then-Encipher paradigm necessitate the tag for decryption, as the output of enciphering contains both the ciphertext and the tag, making them unsuitable for CTX as well.

Additionally, during our survey, we observed that certain AE schemes possess an MRAE counterpart to their UNAE design, with the MRAE designs typically employing the tag for encryption (and hence decryption), thereby rendering them unsuitable for CTX. Moreover, among these AE schemes, many are insecure even against a constant-time $\mathrm{CMT}_{\mathrm{k}}$ adversary, thus ruling out the applicability of HtE.

To the best of our knowledge, there is currently no CMT transform that is both universal, meaning it can be applied to any legacy AE schemes, and output-length-preserving, meaning it avoids ciphertext expansion. We stress the significance of zero ciphertext expansion for any CMT transform from a practical standpoint, as preserving ciphertext-size is crucial for compatibility with existing hardware, databases, or communication protocols. Additionally, we highlight the need for universality because, although efficient transforms like SC $\circ$ CTY are available, they are not applicable to TDD schemes, as mentioned earlier.

1.1 Our Contributions

As a first contribution, we show simple two-query MRAE attacks on the AE transforms CTY and SC$\circ$CTY. Our attacks demonstrate that unlike CTX, these transforms do not preserve the MRAE security of legacy AE schemes.

As the main contribution of this work, we propose a new generic AE transform, which we call PACT. This is the first output-length-preserving generic AE transform which is *universal*, i.e., it can ensure the CMT security of *any* legacy AE scheme. Specifically, this is the first such transform which can provide CMT security to TDD AE schemes. In addition, unlike CTY and SC $\circ$ CTY, PACT also has the useful property of preserving the MRAE security of the underlying AE scheme. Thus, for TDD schemes which are MRAE secure to begin with (like Deoxys-II and Joltik$^{=}$), PACT preserves MRAE security while additionally providing CMT security.

Technical Overview of PACT. To give a brief overview, we start with an AE scheme Π that produces a ciphertext C and a tag T using $\Pi.\mathsf{Enc}(K, N, A, M)$. PACT modifies the tag T to a new tag T^* by hashing (K, N, A) using an unkeyed hash function and then making a block cipher call with the hash output as the key and T as the input.

At a high level, PACT differs from CTX in the fact that PACT encrypts the tag instead of hashing it. This design choice ensures that the modification of the tag from T to T^* is reversible, allowing the retrieval of the original tag T. This allows for decryption using $\Pi.\mathsf{Dec}(K, N, A, C, T^*)$ for all Π, even if their decryption algorithm utilises T.

We show that PACT is not heavier than HtE and CTX from the design point of view. It's also worth noting that unlike PACT, both HtE and CTX rely on the Random Oracle or PRF assumption for the underlying hash function. In contrast, PACT relies solely on the collision-resistance property of the hash and utilises a single block cipher call, which we model as an ideal cipher for security proof.

Ciphertext Collision. For the concrete security analysis of PACT, we introduce a new security notion for authenticated encryption schemes, termed the "ciphertext collision advantage". We demonstrate that to achieve a robust CMT security bound for PACT, it's essential for the legacy AE scheme to possess a small ciphertext collision advantage. Conversely, we contend that an AE scheme deemed "good" should inherently exhibit a small ciphertext collision advantage, roughly of the order $(q^2/2^{|S|} + q^2/2^{|T|})$, where the number of AE computations by the ciphertext collision adversary is q, and $|S|, |T|$ refer to the state-size (e.g. block-size for block cipher-based AEs) and tag-size of the AE respectively. Furthermore, we substantiate this assertion by examining several classes of AE schemes to confirm that they indeed possess a small ciphertext collision advantage. It's important to note that we don't view this requirement as a special condition solely for a legacy AE scheme to be compatible with PACT.

comPACT: Lighter variant for UNAE security. While PACT is universally applicable, we also propose a more efficient version, termed comPACT. It applies to the cases where we don't need to preserve the MRAE security of the legacy AE, i.e., it remains only UNAE secure after applying comPACT. The difference lies in how associated data is handled: in PACT, it's processed by both the legacy AE and the hash function, whereas comPACT only involves the associated data for the hash function, leaving the legacy AE with empty associated data. The CMT security of PACT extends to comPACT, with similar security analyses for privacy and authenticity for UNAE schemes. We also compare comPACT with CTY, and give an attack to show that neither of them is MRAE-secure.

1.2 CMT_k Security: Can we do better?

So far most of the practical commitment attacks on the AE schemes which are used in important applications have been CMT_k attacks which motivated the cryptography community to explore this particular research direction in the first place. Surprisingly all the generic length-preserving transforms so far are CMT transforms. One might intuitively think that a good exercise could be to try and design a generic length-preserving CMT_k-only transform which prevents the practical CMT_k attacks and potentially has a lighter design as compared to the generic CMT transforms. We tried to explore this direction and found some intuition about why it might be a little difficult.

A general underlying theme of all the CMT transforms proposed so far is to use a collision-resistant function on the key-nonce-associated data tuple, which is very much in line with the definition of CMT security itself. Now, if we try to extrapolate this idea to design a CMT_k-only transform, we don't need the collision-resistant function, and can simply use the key only. Unfortunately, however, this strategy doesn't work in most of the AE schemes. The issue is that once an input-output tuple is fixed, the associated data can often be used to adjust an entire new input tuple and force the output to be equal to the previous

output. As a result, one needs a somewhat novel idea to achieve a potentially lighter CMT_k-only transform. In the following we illustrate this with the concrete example of Deoxys-II.

Suppose a CMT_k adversary of Deoxys-II obtains a (K, N, A, M, C, T) tuple by a Deoxys-II computation. Then it queries (K, T) to the block cipher E to obtain T^*. Then it fixes some (K', N') with $K' \neq K$ and makes a (K', T^*) query to E^{-1} to obtain T'. Now, given K', N', C, T', the adversary can compute an unverified message M'. Since the adversary has yet to choose an associated data, it can just choose one block associated data A' such that M' is verified. In this way, the adversary generates two tuples (K, N, A, M, C, T^*) and (K', N', A', M', C, T^*) of the CMT_k-only transform and breaks its CMT_k security.

1.3 Related Work

The research in the field of committing encryption goes back to 2003 with Gertner and Herzberg [GH03], who considered the problem in both the symmetric and asymmetric settings. Abdalla et al. [ABN10] gave definitions for what they termed robustness. This work was in the asymmetric setting and required an adversary to produce a ciphertext that validly decrypts under two different keys. Their notion encompassed keys that are honestly generated. Later, Farshim et al. [FLPQ13] pointed out that, for some applications, robustness against adversarially chosen keys is critical. They strengthened Abdalla et al.'s notion to address this observation. Farshim et al. [FOR17] contextualised Abdalla et al.'s robustness in the AE setting, initialising the study of what we call committing AE. Shortly after, Grubbs et al. [GLR17] defined compactly committing AE with the goal of constructing schemes that support message franking. Dodis et al. [DGRW18] also targeted message franking and further develop the definitions from [GLR17] by introducing encryptment as a core component of compactly committing AE, which was later explored further in [Hir20, HM22, HM23]. Albertini et al. [ADG+22] observed the possibility of mitigating the attacks described in [GLR17] and [DGRW18] under a weak form of commitment.

Bellare and Hoang [BH22] introduced new encryption-based and decryption-based commitment security notions, which also incorporate the number of inputs the adversary can produce for a single output. Chan and Rogaway [CR22] introduced another notion, which incorporates the degree of control the adversary has on the key. Other contributions in [BH22] and [CR22] have already been discussed. Menda et al. [MLGR23] introduced the notion of context discoverability and investigated discoverability attacks on popular schemes. Len et al. [LGR21] demonstrated password-recovery attacks on non-committing password-based AE schemes. Chen et al. [CFI+23] investigated both key-commitment and context-commitment security of a few AE schemes that are built from a variable-length tweakable cipher or wide block cipher (WBC) via Encode-then-Encipher (EtE) approach [BR00]. Krämer et al. [KSW23] investigated the commitment security of the finalists of the NIST Lightweight Cryptography Standardisation Process [NIS18]. Daemen et al. [DMA23] designed sponge-based commitment secure AE schemes from SHAKE and TurboSHAKE. Degabriele et al. [DFG23] proved commitment security of SpongeWrap. Struck and Weishäupl [SW24] designed leakage resilient commitment secure AE schemes. Derbez et al. [DFI+24] analysed CMT_k attacks against AES-based AE schemes. Dhar et al. [DEJ+24] analysed CMT security of leakage-resilient AE schemes. Bhaumik et al. [BCC+24] analysed commitment and context-discovery security of MACs and AE schemes. Naito et al. [NSS22] proposed FFF, an AE scheme designed by Encode-then-Encipher paradigm on top of a wide block cipher.

1.4 Outline of the Paper

In Section 2 we introduce some notation and cryptographic preliminaries. Section 3 demonstrates a nonce-misuse privacy attack on CTY. In Section 4, we introduce the notion of ciphertext collision advantage for an AE scheme, and explain why we believe it should be small for all good AE schemes. In Section 5, we introduce the PACT transform, and state our security results on the commitment security, privacy,

and authenticity of PACT; we also introduce the lighter transform comPACT suitable for nonce-respecting AE schemes. In Section 6, we illustrate the applications of PACT and comPACT by exhibiting some well-known AE schemes for which we have no known generic context-committing transform without ciphertext expansion. In Section 7, we provide the security proofs for the results from Section 5. In Section 8 we summarise our work and discuss open research problems in this area.

2 Preliminaries

For an event A, $\Pr[A]$ denotes its probability. For a binary string X, $|X|$ denotes its bit-length, while $\lceil X \rceil_n$ and $\lfloor X \rfloor_n$ denote the n most significant bits (MSBs) and least significant bits (LSBs) of X respectively, where $|X| \geq n$. For a set $\mathcal{X}$, $X \leftarrow_{\$} \mathcal{X}$ denotes that X is sampled from $\mathcal{X}$ uniformly at random.

2.1 Cryptographic Primitives

Block Cipher. A block cipher is a function $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ (where n is a positive integer) such that for any $K \in \mathcal{K}$, $E_K(\cdot) := E(K, \cdot)$ is a permutation on $\{0,1\}^n$. K is called the key of the block cipher. In other words, a block cipher is a set of permutations indexed by a key. The typical expectation from a block cipher is that for a randomly sampled secret key K, E_K should behave like a random permutation.

Ideal Cipher. Let $\mathcal{BC}$ be the set of all block ciphers from $\mathcal{K} \times \{0,1\}^n$ to $\{0,1\}^n$. An ideal cipher from $\mathcal{K} \times \{0,1\}^n$ to $\{0,1\}^n$ is an element from $\mathcal{BC}$ chosen uniformly at random. Ideal ciphers are not practical primitives, but are often used as a theoretical representation of a perfect block cipher while studying block cipher-based constructions where the block cipher key is dynamically generated.

An ideal cipher E can be implemented by *lazy sampling* in the following way. E maintains a table $\mathcal{T}_{\text{ic}}$, where each row $\mathcal{T}_{\text{ic}}[K]$ stores the query-response tuples for ideal cipher queries with key K. Let $\mathcal{T}_{\text{ic1}}[K] := \{M | (M, \cdot) \in \mathcal{T}_{\text{ic}}[K]\}$ and $\mathcal{T}_{\text{ic2}}[K] := \{C | (\cdot, C) \in \mathcal{T}_{\text{ic}}[K]\}$. When a new forward (or backward) query (K, M) (or (K, C)) is received, then it is checked if $(M, \cdot)$ (or $(\cdot, C)$) is present in $\mathcal{T}_{\text{ic}}[K]$. If yes, then the response is returned from the table. If not, then E samples and returns $C \leftarrow_{\$} \{0,1\}^n \setminus \mathcal{T}_{\text{ic2}}[K]$ (or $M \leftarrow_{\$} \{0,1\}^n \setminus \mathcal{T}_{\text{ic1}}[K]$) is returned, and updates the row $\mathcal{T}_{\text{ic}}[K]$ to $\mathcal{T}_{\text{ic}}[K] \cup \{(M, C)\}$.

Hash Function. A hash function is a function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^n$ (where n is a positive integer). Sometimes a hash function can take a key as an additional input; while we mostly limit ourselves to unkeyed hash functions in this paper, we will occasionally encounter keyed hash functions while analysing some schemes.

For our proposals, we will consider *collision resistant* hash functions—hash functions for which it is difficult to find two inputs X_1 and X_2 such that $\mathcal{H}(X_1) = \mathcal{H}(X_2)$. We make this notion more formal later in this section. When we write a hash function call with multiple inputs, we implicitly assume an injective padding that combines the inputs into a single input.

Authenticated Encryption. A *nonce-based Authenticated Encryption with associated data* (or in short nAE) involves a key-space $\mathcal{K}$, a nonce-space $\mathcal{N}$, an associated-data-space $\mathcal{AD}$, a message space $\mathcal{M}$ and a ciphertext space $\mathcal{C}$ along with two functions $\mathsf{Enc} : \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{M} \to \mathcal{C}$ (called the Encryption Function) and $\mathsf{Dec} : \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$ (called the Decryption Function). For any key $K \in \mathcal{K}$, nonce $N \in \mathcal{N}$ and associated data $A \in \mathcal{AD}$, an nAE scheme Π is *correct* if for any $M \in \mathcal{M}$, $C \leftarrow \Pi.\mathsf{Enc}(K, N, A, M)$ implies $\Pi.\mathsf{Dec}(K, N, A, C) = M$; and Π is *tidy* if for any $C \in \mathcal{C}$, $M \leftarrow \Pi.\mathsf{Dec}(K, N, A, C)$ implies $\Pi.\mathsf{Enc}(K, N, A, M) = C$. Note that $\Pi.\mathsf{Enc}(K, N, A, \cdot)$ and $\Pi.\mathsf{Dec}(K, N, A)$ are inverse of each other if Π

is both correct and tidy. Also note that if Π is correct, then for a fixed $K \in \mathcal{K}, N \in \mathcal{N}$ and $A \in \mathcal{AD}$, $\mathsf{Enc}(K, N, A, \cdot)$ is an injective function from $\mathcal{M}$ to $\mathcal{C}$. The schemes we consider in this work are tidy.

For an nAE scheme $\Pi := (\Pi.\mathsf{Enc}, \Pi.\mathsf{Dec})$, there is a constant t such that if $C = \Pi.\mathsf{Enc}(K, N, A, M)$, then $|C| = |M| + t$. Usually t bit positions in C are specified, and they are said to constitute the tag, while only the rest of the $|M|$ bit positions are said to constitute the ciphertext. In this paper, without loss of generality, we assume the last t bits of the output to be the tag unless otherwise specified. A *Deterministic Authenticated Encryption with associated data* (or in short DAE) is similar to nAE except that it does not take any nonce as input. In this work, we use the generic term AE to denote an authenticated encryption which can be either nonce-based or deterministic.

We write $(K, N, A, M, C, T) \in \Pi$ (or $(K, N, A, M, C, T) \notin \Pi$) to denote that a party interacting with an AE Π has already verified that the tuple (K, N, A, M, C, T) is compatible (or incompatible, respectively) with Π. This is different from the notation $(C, T) = \Pi.\mathsf{Enc}(K, N, A, M)$ or $M = \Pi.\mathsf{Dec}(K, N, A, C, T)$, either of which only states the correctness condition for the tuple (K, N, A, M, C, T) with respect to Π, but doesn't guarantee any verification of the correctness by the interacting party.

2.2 Security Notions

Distinguishing Advantage. For two oracles $\mathcal{O}_0$ and $\mathcal{O}_1$, an algorithm $\mathcal{A}$ which tries to distinguish between $\mathcal{O}_0$ and $\mathcal{O}_1$ is called a distinguishing adversary. $\mathcal{A}$ plays an interactive game with $\mathcal{O}_b$ where b is unknown to $\mathcal{A}$, and then outputs a guess for b; $\mathcal{A}$ wins when the guessed bit matches b. The distinguishing advantage of $\mathcal{A}$ is defined as

$$\mathbf{Adv}_{\mathcal{O}_1, \mathcal{O}_0}(\mathcal{A}) := \Big| \Pr_{\mathcal{O}_0}[\mathcal{A} \Rightarrow 1] - \Pr_{\mathcal{O}_1}[\mathcal{A} \Rightarrow 1] \Big|,$$

where the subscript of Pr denotes the oracle with which $\mathcal{A}$ is playing. We note that this notation is general enough to capture games where each oracle implements multiple functions, e.g., F can handle both encryption and decryption queries by accepting an extra bit to indicate the direction of queries. We'll interchangeably use the distinguishing advantage notation with games instead of oracles when the adversary is trying to distinguish between two games.

Privacy and Authenticity. We consider two types of adversaries of an nAE scheme. A *privacy adversary* $\mathcal{A}$ of an nAE scheme $\Pi := (\Pi.\mathsf{Enc}, \Pi.\mathsf{Dec})$ is a distinguishing adversary which tries to distinguish between $\Pi.\mathsf{Enc}$ and an oracle Π^* which works as follows. For an encryption query (N, A, M), it samples an element from $\{0,1\}^{|M|+t}$ uniformly at random, and outputs it. The advantage of $\mathcal{A}$ is called *privacy advantage*, denoted by $\mathbf{Adv}^{\mathrm{AE-PRIV}}_{\Pi}(\mathcal{A})$.

Sometimes $\mathcal{A}$ is assumed to be *nonce-respecting*, i.e., it isn't supposed to repeat a nonce. Depending on whether we make this assumption or not, we use the terms UNAE (Unique Nonce AE) or MRAE (Misuse Resistant AE) in place of AE. In the case of an MRAE scheme, Π^* can be fine-tuned to $\Pi^\dagger$ for the privacy game in the following game. For every encryption query $(N, A, \cdot)$, it samples an element from $\{0, 1\}^{|M|+t}$ uniformly at random without replacement, and outputs it. But if we use Π^* instead of $\Pi^\dagger$, $\mathcal{A}$ can distinguish it from $\Pi.\mathsf{Enc}$ only after it can observe a collision at the output of Π^* for the same (N, A) tuple, which will happen after about $2^{|M|+t}$ encryption queries. Since we don't deal with beyond-birthday-bound security in this work, we use Π^* for MRAE as well.

An *authenticity adversary* $\mathcal{B}$ of Π is an adversary which has access to both $\Pi.\mathsf{Enc}$ and $\Pi.\mathsf{Dec}$, and tries to "forge", i.e. make a successful query to $\Pi.\mathsf{Dec}$ and this query is not the result of a previous encryption query, and its advantage is called *authenticity advantage*, denoted by $\mathbf{Adv}^{\mathrm{AE-AUTH}}_{\Pi}(\mathcal{B})$. We always impose a restriction on each adversary that it can't make *pointless* queries, i.e., can't repeat the same query multiple times or can't make the query (N, A, C, T) to $\Pi.\mathsf{Dec}$ if it has already made a query (N, A, M) to $\Pi.\mathsf{Enc}$ and received (C, T) in response.

$\mathsf{CTY}[\Pi, \mathcal{H}].\mathsf{Enc}(K, N, A, M)$

1: $C \leftarrow \Pi.\mathsf{Enc}(K, N, \epsilon, M)$
2: $T \leftarrow \Pi.\mathsf{Auth}(K, N, \epsilon, M)$
3: $T^* \leftarrow \mathcal{H}(K, N, A, T)$
4: **return** (C, T^*)

$\mathsf{CTY}[\Pi, \mathcal{H}].\mathsf{Dec}(K, N, A, C, T^*)$

1: $M \leftarrow \Pi.\mathsf{Dec}(K, N, \epsilon, C)$
2: $T \leftarrow \Pi.\mathsf{Auth}(K, N, \epsilon, M)$
3: **If** $T^* \neq \mathcal{H}(K, N, A, T)$
4: **return** $\perp$
5: **return** M

Figure 3: Specification of CTY. ϵ denotes empty string.

Collision Advantage. Let $\mathcal{X}$ and $\mathcal{Y}$ be two non-empty sets, and F be a function from $\mathcal{X}$ to $\mathcal{Y}$. Suppose an adversary $\mathcal{A}$ outputs a tuple (x_1, x_2). $\mathcal{A}$ wins if $x_1 \neq x_2$ and $F(x_1) = F(x_2)$. The *collision advantage* of $\mathcal{A}$ is defined by

$$\mathbf{Adv}_F^{\text{Coll}}(\mathcal{A}) := \Pr[\mathcal{A} \text{ wins}] = \Pr[F(x_1) = F(x_2)].$$

We call $\mathcal{A}$ a *collision adversary*.

2.3 Commitment Security

The two prevalent notions of commitment security in the literature are: committing solely to the key K, which we call CMT_k security, and committing to the complete context, denoted as (K, N, A), which we call CMT security. Bellare and Hoang [BH22] demonstrated that incorporating the message M into the context is unnecessary, as committing to (K, N, A) alone is equivalent to committing to (K, N, A, M).

CMT_k **game for AE.** In the CMT_k game against an AE scheme Π, an adversary $\mathcal{A}$ outputs (K, N, A, M) and (K', N', A', M'); $\mathcal{A}$ wins if:

- $K \neq K'$;
- $\Pi.\mathsf{Enc}(K, N, A, M) = \Pi.\mathsf{Enc}(K', N', A', M')$.

We write $\mathbf{Adv}_\Pi^{\text{CMT}_\text{k}}(\mathcal{A})$ to denote the probability that $\mathcal{A}$ wins.

CMT **game for AE.** In the CMT game against an AE scheme Π, an adversary $\mathcal{A}$ outputs (K, N, A, M) and (K', N', A', M'); $\mathcal{A}$ wins if:

- $(K, N, A) \neq (K', N', A')$;
- $\Pi.\mathsf{Enc}(K, N, A, M) = \Pi.\mathsf{Enc}(K', N', A', M')$.

We write $\mathbf{Adv}_\Pi^{\text{CMT}}(\mathcal{A})$ to denote the probability that $\mathcal{A}$ wins.

3 A Nonce-misuse Privacy Attack on CTY

Figure 3 gives the complete specification of CTY. When CTY is applied to a legacy AE scheme, the ciphertext C is independent of the associated data A. An adversary that can misuse nonces can use this to mount the following simple 2-query privacy attack.

It makes two queries to the encryption oracle: (N, A, M) and (N, A', M) with $A \neq A'$, receiving (C, T) and (C', T') as responses. If $C = C'$, the adversary concludes that the oracle is real; otherwise, it concludes the oracle is ideal. The adversary almost always succeeds since, in the real scenario, $\Pr[C = C'] = 1$, while in the ideal scenario, the probability is only $1/2^{|C|}$.

Thus, while CTX and CTY offer comparable CMT security, CTY does not maintain MRAE security. Because SC on its own does not preserve AE security, SC ∘ CTY also fails to preserve MRAE security.

While we have noted that MRAE schemes typically use the tag during decryption, making CTY inapplicable in those cases, there are still MRAE schemes to which CTY can be applied. For instance, KIASU is an MRAE scheme where CTY is applicable. However, for such schemes, although CTY provides CMT security, it compromises MRAE security.

4 Ciphertext Collision in AE Schemes

In this section, we introduce a concept related to AE schemes called the "ciphertext collision". In essence, for any two key-nonce-associated data tuples (K_1, N_1, A_1) and (K_2, N_2, A_2), and a tag T_1, a ciphertext collision in a scheme Π refers to the situation where upon receiving a randomly generated tag T_2 one obtains two messages M_1 and M_2 such that for some ciphertext C it holds that

$$\Pi.\mathsf{Enc}(K_1, N_1, A_1, M_1) = (C, T_1), \text{ and } \Pi.\mathsf{Enc}(K_2, N_2, A_2, M_2) = (C, T_2).$$

This is typically unnecessary in AE design scenarios where adversarial access to the key is not assumed. However, when an adversary can choose the key, it can launch commitment attacks using the ciphertext collision. Therefore, from a committing security standpoint, our goal is to ensure that the probability of ciphertext collision in an AE scheme is minimal. We first formally define the relevant terms.

4.1 Ciphertext Collision Advantage

Let Π be an AE scheme with tag space $\mathcal{T}$. In the course of a *ciphertext collision security game*, an adversary $\mathcal{A}$ chooses q_1 key-nonce-associated data-tag tuples $\{(K_i^0, N_i^0, A_i^0, T_i^0) \mid i \in [q_1]\}$. For each $i \in [q_1]$, it further chooses m_i key-nonce-associated data tuples $\{(K_i^j, N_i^j, A_i^j) \mid j \in [m_i]\}$, where $m_1 + \cdots + m_{q_1} = q_2$. For each $i \in [q_1], j \in [m_i]$, and an $\epsilon > 0$, a tag T_i^j is sampled from a distribution which guarantees that for any $c \in \mathcal{T}, \Pr[T_i^j = c] \leq \epsilon$. $\mathcal{A}$ can make its choices adaptively and in any order, based on previously sampled values of T_i^j.

At the end of the game, $\mathcal{A}$ outputs a couple of messages M_1 and M_2, and three indices i, j_1 and j_2 with $0 \leq j_1 < j_2$, and wins if there exists a ciphertext C such that $\Pi.\mathsf{Enc}(K_i^{j_1}, N_i^{j_1}, A_i^{j_1}, M_1) = (C, T_i^{j_1})$ and $\Pi.\mathsf{Enc}(K_i^{j_2}, N_i^{j_2}, A_i^{j_2}, M_2) = (C, T_i^{j_2})$. The *ciphertext collision advantage* of $\mathcal{A}$ is defined as

$$\mathbf{Adv}_{\Pi}^{\mathrm{CC}}(\mathcal{A}) := \Pr[\mathcal{A} \text{ wins}].$$

$\mathcal{A}$ is called a *(q_1, q_2, ϵ)-ciphertext collision adversary* of Π. Among all (q_1, q_2, ϵ)-ciphertext collision adversaries of Π, the advantage of the adversary with maximum advantage is called the *(q_1, q_2, ϵ)-ciphertext collision security* of Π.

Here, we provide a rough intuition about what q_1 and q_2 represent. In Section 5, we define the PACT transform, which converts any AE scheme Π into a CMT-secure one. The PACT transform takes as input the AE scheme Π, a hash function $\mathcal{H}$, and a block cipher E. In the security analysis of PACT, we consider a (q_1, q_2, ϵ)-ciphertext collision adversary $\mathcal{A}$, where q_1 is the number of ideal cipher queries and q_2 is the number of Π computations made by $\mathcal{A}$. This is a very high-level intuition, with the detailed explanation provided in Section 5.

Our next objective is to demonstrate that for any AE scheme Π, the ciphertext collision advantage is not substantial. In the following subsections, we analyse the ciphertext collision advantage for two prominent AEs, Deoxys-II (in Section 4.2) and Ascon (in Section 4.3) in detail. Then, in Section 4.4, we argue why this advantage should not be large for any AE scheme. We would like to mention that bounding the ciphertext

Deoxys-II[$\tilde{E}$].Enc($K, N, A = A[1]\| \dots \|A[a], M = M[1]\| \dots \|M[m]$)

1: Auth $\leftarrow 0$
2: **for** $i \in [1..a]$ **do**
3: Auth $\leftarrow$ Auth $\oplus \tilde{E}_K(0010\|i, A[i])$ ▷ $\tilde{E}$ is a tweakable block cipher (TBC)
4: **end for**
5: $T \leftarrow$ Auth
6: **for** $j \in [1..m-1]$ **do**
7: $T \leftarrow T \oplus \tilde{E}_K(0000\|j, M[j])$
8: **end for**
9: $T \leftarrow T \oplus \tilde{E}_K(0100\|m, \mathsf{ozpad}(M[m])$ ▷ ozpad is the 10^* padding if $|M[m]| < n$
10: $T \leftarrow \tilde{E}_K(0001\|0^4\|N, T)$
11: **for** $j \in [1..m]$ **do**
12: $Z[j] \leftarrow \tilde{E}_K(1\|T \oplus j, 0^8\|N)$
13: **end for**
14: $Z \leftarrow Z[1]\| \dots \|Z[m]$
15: **return** $(C := M \oplus \lceil Z \rceil_{|M|}, T)$

Figure 4: Encryption of Deoxys-II

collision advantage of a general AE scheme depends on some equations intrinsic to the scheme, and thus providing an exact bound for a general scheme is tough. We have reviewed numerous AEs in practice, and our findings support our argument. The choice of Deoxys-II and Ascon for detail analysis is also justified in Section 4.4.

4.2 Ciphertext Collision Security of Deoxys-II

Let us first look at Deoxys-II [JNPS19,JNPS21], an AE selected as the first choice for the "in-depth security" portfolio of the CAESAR competition. The encryption algorithm for Deoxys-II is given in Fig. 4. Let $\mathcal{A}$ be a (q_1, q_2, ϵ)-ciphertext collision adversary of Deoxys-II. We calculate $\mathbf{Adv}^{\mathrm{CC}}_{\mathsf{Deoxys\text{-}II}}(\mathcal{A})$. In the following analysis Π refers to Deoxys-II.

Let us fix i, j_1, j_2, the indices for which $\mathcal{A}$ finds a collision. First assume that the Π computation at index j_1 happens after the Π computation at index j_2. Let the Π computation at index j_2 yield

$$\Pi.\mathsf{Enc}(K_i^{j_2}, N_i^{j_2}, A_i^{j_2}, M_2) = (C, T_i^{j_2}).$$

Then, for $\mathcal{A}$ to win, we must obtain a message M_1 such that

$$\Pi.\mathsf{Enc}(K_i^{j_1}, N_i^{j_1}, A_i^{j_1}, M_1) = (C, T_i^{j_1}).$$

We divide this into two cases depending on whether $\mathcal{A}$ starts with C and tries to generate an M_1 or starts with some M_1 and tries to obtain C as the ciphertext. In each case, we calculate

$$\Pr[\Pi.\mathsf{Enc}(K_i^{j_1}, N_i^{j_1}, A_i^{j_1}, M_1) = (C, T_i^{j_1})]$$

for the M_1 selected. When $\mathcal{A}$ starts with C at index j_1, the analysis differs depending on whether $j_1 = 0$ or $j_1 > 0$. Therefore, we consider the following three cases:

- Case 1: $\mathcal{A}$ starts with C at index $j_1 > 0$. In this case, given $K_i^{j_1}, N_i^{j_1}, A_i^{j_1}, T_i^{j_1}$, the unverified message $M_1 = M_{1,1}\| \dots \|M_{1,l}$ is generated blockwise. The next step is the verification of the tag $T_i^{j_1}$. Using $K_i^{j_1}, N_i^{j_1}, A_i^{j_1}, M_1$, the adversary can compute a new tag $\tilde{T}$, and $\tilde{T}$ is independent of $T_i^{j_1}$. For the

tag verification to be successful, we must have $\tilde{T} = T_i^{j_1}$. Since, $j_1 > 0$, $T_i^{j_1}$ is sampled following a distribution which guarantees that for any $c \in \mathcal{T}, \Pr[T_i^j = c] \leq \epsilon$. Hence, the probability that the tag verification is successful is at most ϵ.

- Case 2: $\mathcal{A}$ starts with C at index $j_1 = 0$. In this case, T_i^0 is chosen by $\mathcal{A}$. Given $K_i^0, N_i^0, A_i^0, T_i^0$, the unverified message $M_1 = M_{1,1} \| \dots \| M_{1,l}$ is generated blockwise. The next step is the verification of the tag T_i^0. Since K_i^0, N_i^0, A_i^0 is fixed, so is Auth_i^0. First, let us assume $|M_{1,l}| = n$. Then, the last message block $M_{1,l}$ must satisfy

$$M_{1,l} = \tilde{E}^{-1}_{K_i^0}\Big(0^4\|(l-1), \big(\mathsf{Auth}_i^0 \oplus \sum_{u=1}^{l-1} \tilde{E}(0^4\|(u-1), M_{1,u}) \oplus \tilde{E}^{-1}_{K_i^0}(1\|0^4\|N_i^0, T_i^0)\big)\Big). \tag{1}$$

  This occurs with probability $1/2^n$. If $|M_{1,l}| < n$, then we replace $M_{1,l}$ by $M_{1,l}\|10^*$ in the above equation and the probability remains the same.

- Case 3: $\mathcal{A}$ starts with an M_1 at index j_1. Then, $\mathcal{A}$ must obtain

$$\Pi.\mathsf{Enc}(K_i^{j_1}, N_i^{j_1}, A_i^{j_1}, M_1) = (C, T_i^{j_1}).$$

  Regardless of whether j_1 equals 0 or not, $\mathcal{A}$ must choose $M_1 = M_{1,1} \| \dots \| M_{1,l}$ such that $T_i^{j_1}$ is verified. To verify the tag, note that $\mathcal{A}$ can freely choose any $(l-1)$ blocks of M_1 (for instance, the last $(l-1)$ blocks), while the first block must satisfy

$$M_{1,1} = \tilde{E}^{-1}_{K_i^{j_1}}\Big(0, \big(\mathsf{Auth}_i^{j_1} \oplus \sum_{u=2}^{l} \tilde{E}(0^4\|(u-1), M_{1,u}) \oplus \tilde{E}^{-1}_{K_i^{j_1}}(1\|0^4\|N_i^{j_1}, T_i^{j_1})\big)\Big). \tag{2}$$

  One thing to note here is that $M_{1,1}$ must be a full block. Otherwise, the verification of $T_i^{j_1}$ is probabilistic, and if $|M_1| = n - k$, then $\Pr[T_i^{j_1}$ is verified$] = 1/2^k$ accounting for the k bits of fixed padding.

  Now that the tag is verified, $\mathcal{A}$ needs to obtain C as the ciphertext. Since $\mathcal{A}$ can choose the last $(l-1)$ blocks of M_1 freely, $\mathcal{A}$ can match the last $(l-1)$ blocks, but given that $M_{1,1}$ is fixed,

$$\Pr[C_1 = M_{1,1} \oplus \tilde{E}_K(1\|T \oplus 1, 0^8\|N)] = \frac{1}{2^{|C_1|}}.$$

  If $|C_1| = n$, then the probability of $\mathcal{A}$ winning in this case is bounded by $1/2^n$. If $|C_1| = n - k$, then for $\mathcal{A}$ to win both $T_i^{j_1}$ needs to be verified and C_1 needs to match, the probability of which is bounded above by $1/2^k \times 1/2^{n-k} = 1/2^n$.

Next, if we assume that the Π computation at index j_2 happens after the Π computation at index j_1, note that Case 2 does not arise since $j_2 > 0$, and the analysis of the other cases are exactly the same.

Hence, $\Pr[\mathcal{A}$ wins $\mid i, j_i, j_2] = 2\epsilon + 3/2^n$. Summing over $0 \leq j_1 < j_2$, we have

$$\Pr[\mathcal{A} \text{ wins} \mid i] \leq m_{q_i}(m_{q_i} + 1)(2\epsilon + 3/2^n).$$

Finally, summing over all i, we have

$$\Pr[\mathcal{A} \text{ wins}] \leq q_2(q_2 + 1)(2\epsilon + 3/2^n).$$

$\mathsf{Ascon}[P,Q].\mathsf{Enc}(K,N,A=A[1]\|\ldots\|A[a],M=M[1]\|\ldots\|M[m])$

$S \leftarrow IV\|K\|N$ ▷ $|S|=b, |A[i]|=|M[i]|=r, c=b-r, |K|=k$
$S \leftarrow P(S)\oplus(0^{b-k}\|K)$ ▷ P is the outer permutation of Ascon
for $i \in [1..a]$ **do**
  $S \leftarrow Q(\lceil S\rceil_r \oplus A[i])\|\lfloor S\rfloor_c$ ▷ Q is the inner permutation of Ascon
end for
$S \leftarrow S\oplus(0^{b-1}\|1)$
for $j \in [1..m-1]$ **do**
  $\lceil S\rceil_r \leftarrow \lceil S\rceil_r \oplus M[i]$
  $C[i] \leftarrow \lceil S\rceil_r$
  $S \leftarrow Q(S)$
end for
$\lceil S\rceil_r \leftarrow \lceil S\rceil_r \oplus M[m]$
$C[m] \leftarrow \lceil S\rceil_{M[m]}$
$S \leftarrow P(S\oplus(0^r\|K\|0^{b-r-k})$
$T \leftarrow \lfloor S\rfloor_\tau \oplus \lfloor K\rfloor_\tau$
return $(C := C[1]\|\ldots\|C[m],T)$

Figure 5: Encryption of Ascon

4.3 Ciphertext Collision Security of Ascon

Now, we look at Ascon, the NIST LWC standard, and also the primary choice for the "lightweight authenticated encryption" portfolio of the CAESAR competition. The encryption algorithm for Ascon is given in Fig. 5. Let $\mathcal{A}$ be a (q_1,q_2,ϵ)-ciphertext collision adversary of Ascon. We calculate $\mathbf{Adv}^{\mathrm{CC}}_{\mathsf{Ascon}}(\mathcal{A})$. In the following analysis, Π refers to Ascon.

As in the case of Deoxys-II, we fix i, j_1, j_2 and assume that the Π computation at index j_1 happens after the Π computation at index j_2. It is easy to see that the winning event of $\mathcal{A}$ can be divided into the same three cases as above. We analyse the three cases for Ascon now.

- Case 1: $\mathcal{A}$ starts with C at index $j_1 > 0$. Note that Ascon is CTX-friendly. Hence, given $K_i^{j_1}, N_i^{j_1}, A_i^{j_1}, C$, the adversary $\mathcal{A}$ can generate the verification tag $\tilde{T}$ as

$$\tilde{T} = \Pi.\mathsf{Auth}(K,N,A,\Pi.\mathsf{Dec}(K,N,A,C)),$$

  and hence $\tilde{T}$ and $T_i^{j_1}$ are necessarily independent. Since, $j_1 > 0$, $T_i^{j_1}$ is sampled following a distribution which guarantees that for any $c \in \mathcal{T}, \Pr[T_i^j = c] \leq \epsilon$. Hence, the probability that the tag verification is successful is at most ϵ.

- Case 2: $\mathcal{A}$ starts with C at index $j_1 = 0$. In this case, T_i^0 is chosen by $\mathcal{A}$. But since $\mathcal{A}$ gets C after it chooses T_i^0, the tag verification is not guaranteed. In fact, we must have

$$C_l\|X = P^{-1}(Y\|(T_i^0 \oplus \lceil K_i^0\rceil_\tau)), \tag{3}$$

  where X is a function of $(K_i^0, N_i^0, A_i^0, C_1, \ldots, C_{l-1})$ and Y can be chosen freely by $\mathcal{A}$. For an ideal permutation P, the probability of this event is $1/2^b$, where b is the permutation-size.

- Case 3: $\mathcal{A}$ starts with an M_1 at index j_1. For a successful ciphertext collision, we must have $M_1 = \Pi.\mathsf{Dec}(K_i^{j_1}, N_i^{j_1}, A_0^{j_1}, C)$. Even if the adversary chooses this M_1, the tag verification boils down to either Case 1 or Case 2 depending on the value of j_1.

Considering the instance when the Π computation at index j_2 happens after the Π computation at index j_1, and summing over all i, j_1, j_2, we have

$$\Pr[\mathcal{A} \text{ wins}] \leq q_2(q_2+1)(4\epsilon + 3/2^b).$$

4.4 Generalisation

In this section, we conjecture that if $\epsilon \leq 2/2^{|T|}$, then for a "well-designed" AE scheme, the ciphertext collision advantage of $\mathcal{A}$ is

$$\mathcal{O}\left(\frac{q_2^2}{2^{|T|}} + \frac{q_2^2}{2^{|S|}}\right),$$

where q_2 is total the number of Π computations available to $\mathcal{A}$, $|S|$ is the size of the primitive (block-size for block cipher-based AE or permutation-size for permutation-based AE), and $|T|$ is the tag-size. By "well-designed", we mean that the AE scheme must be built upon near-ideal primitives. We now provide an intuition for why we believe our conjecture is correct. Note that the exact ciphertext collision advantage of a scheme would depend on the scheme itself. For any Π, the cases remain the same as above, although their analyses will differ based on the scheme.

Starting with the first case, note that the analysis hinges on the fact that the tag $\tilde{T}$ generated during the verification process and the randomly chosen starting tag T_i^j are independent for both Deoxys-II and Ascon. We argue that this holds for any well-designed AE scheme.

First, note that in case of a CTX-friendly AE scheme Π, the valid tag $\tilde{T}$ corresponding to a tuple (K_i^j, N_i^j, A_i^j, C) can be obtained by

$$\tilde{T} = \Pi.\mathsf{Auth}(K_i^j, N_i^j, A_i^j, \Pi.\mathsf{Dec}(K_i^j, N_i^j, A_i^j, C)),$$

and hence $\tilde{T}$ and T_i^j are necessarily independent.

Next, for any AE scheme Π where decryption precedes verification in the decryption algorithm, if we select a tuple $(K_i^j, N_i^j, A_i^j, C, T_i^j)$ with a random T_i^j and decrypt, the unverified message M should essentially be almost independent of T_i^j for any well-designed AE scheme. Many schemes like Mac-then-Encrypt and SIV (most prominent CTX-unfriendly schemes fall in this category) then employ a verification algorithm to generate T^*, stating that verification is successful if $T_i^j = T^*$. This T_i^j and T^* are independent. Therefore, the success probability of $\mathcal{A}$ in this case is of the order $1/2^{|T|}$ for all well-designed schemes.

Next, for Case 2, note that $\mathcal{A}$ chooses a starting tag T_i^0 before receiving any C from which it needs to construct M. The verification tag $\tilde{T}$ must be a function of either M or C; it cannot be solely a function of (K_i^0, N_i^0, T_i^0). We select Deoxys-II and Ascon for detailed discussions because $\tilde{T}$ depends on M in the case of Deoxys-II and on C in the case of Ascon.

If $\tilde{T}$ is a function of C, then since C is selected later, not all values of C will decrypt successfully. For successful decryption (i.e., $\tilde{T} = T_i^0$), C and T_i^0 must be related in some way (e.g., as in equation 3 for Ascon). In most schemes, at least one ciphertext block needs to be part of the relation.

If $\tilde{T}$ is a function of M, then a later-selected C can be decrypted to obtain an unverified message M, but for successful verification (i.e., $\tilde{T} = T_i^0$), M and T_i^0 must be related in some way (e.g., as in equation 1 for Deoxys-II). In most schemes, at least one message block needs to be part of the relation. In both these subcases, for all well-designed schemes, the success probability of $\mathcal{A}$ is of the order of $1/2^{|S|}$.

Finally, in Case 3 too, note that the starting tag T_i^j is chosen (either by $\mathcal{A}$ or randomly) before $\mathcal{A}$ receives C and selects an M to generate this C. For generating C, successful verification is necessary (i.e., the verification tag $\tilde{T}$ must match T_i^j).

Similar to Case 2, at least one block of either C or M needs to be related to T_i^j, depending on whether $\tilde{T}$ is a function of C or M. Therefore, in this case as well, the success probability of $\mathcal{A}$ is of the order of $1/2^{|S|}$.

$\mathsf{PACT}[\Pi, \mathcal{H}, E].\mathsf{Enc}(K, N, A, M)$

1: $(C, T) \leftarrow \Pi.\mathsf{Enc}(K, N, A, M)$
2: $K^* \leftarrow \mathcal{H}(K, N, A)$
3: $T^* \leftarrow E(K^*, T)$
4: **return** (C, T^*)

$\mathsf{PACT}[\Pi, \mathcal{H}, E].\mathsf{Dec}(K, N, A, C, T^*)$

1: $K^* \leftarrow \mathcal{H}(K, N, A)$
2: $T \leftarrow E^{-1}(K^*, T^*)$
3: $X \leftarrow \Pi.\mathsf{Dec}(K, N, A, C, T)$
4: **return** X

Figure 6: Specification of PACT

Then, taking a sum over i, j_1, j_2, our intuition is that the ciphertext collision advantage is

$$\mathcal{O}\left(\frac{q_2^2}{2^{|T|}} + \frac{q_2^2}{2^{|S|}}\right).$$

5 The PACT Transform

Let Π be an AE scheme having an encryption algorithm $\Pi.\mathsf{Enc}(K, N, A, M)$ that generates a ciphertext C and tag T. (If the AE scheme doesn't specify the tag and instead returns an expanded ciphertext C^* with $\lceil C^* \rceil = |M| + t$, we denote the ciphertext core as $C = \lceil C^* \rceil_{|M|}$ and the remaining t bits as the tag T.) Our PACT construction uses the same ciphertext but replaces the tag T with an alternative tag T^*. We compute T^* by hashing (K, N, A) using an unkeyed hash function and then feeding the hash output as the key into a block cipher along with the original tag T as the input.

Let $\mathcal{H}$ be a hash function and E be a block cipher such that the key-size of E is equal to the output-size of $\mathcal{H}$ and the block-size of E is equal to the tag-size of Π. We denote as $\mathsf{PACT}[\Pi, \mathcal{H}, E]$ the PACT transform using $\mathcal{H}$ and E applied to Π. The complete specification of $\mathsf{PACT}[\Pi, \mathcal{H}, E]$ is given in Figure 6. Note that we treat the nonce and the associated data in a similar fashion, and the interface of our transform accepts DAE schemes as well.

It is worth mentioning that while hashing the context (K, N, A) has become somewhat standard for committing security (both HtE and CTX employ this method), PACT introduces a block cipher call at the end. This design ensures that the modification of the tag from T to T^* is reversible. Unlike CTX, the information on the original tag T is preserved, allowing for decryption using $\Pi.\mathsf{Dec}(K, N, A, C, T^*)$ for all Π, even if their decryption algorithm utilises T. For the security proofs, we model E as an ideal cipher.

We claim that $\mathsf{PACT}[\Pi, \mathcal{H}, E]$ is CMT-secure as long as $\mathcal{H}$ is collision-resistant, and the advantage of any ciphertext collision adversary of Π is small.

Theorem 1. *For any* CMT *adversary* $\mathcal{A}$ *of* $\mathsf{PACT}[\Pi, \mathcal{H}, E]$ *making* q *computations of* Π *and* q_C *queries to the* t*-bit ideal cipher* E*, we can construct a collision adversary* $\mathcal{B}_1$ *of* $\mathcal{H}$ *and a* $(q_C, q, 2/2^t)$*-ciphertext collision adversary* $\mathcal{B}_2$ *of* Π *such that*

$$\mathbf{Adv}^{\mathrm{CMT}}_{\mathsf{PACT}[\Pi,\mathcal{H},E]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{Coll}}_{\mathcal{H}}(\mathcal{B}_1) + \mathbf{Adv}^{\mathrm{CC}}_{\Pi}(\mathcal{B}_2) + \frac{2q_C^2}{2^t} + \frac{2qq_C}{2^t} + \frac{2q_C}{2^t},$$

where $q_C < 2^{t-1}$*, and the computation time of* $\mathcal{B}_1$ *doesn't exceed that of* $\mathcal{A}$ *by more than a constant.*

The proof of Theorem 1 is deferred to Section 7.1. We have already argued that the ciphertext collision advantage should be $\mathcal{O}(q^2/2^{|S|} + q^2/2^t)$ (since the tag-size $|T|$ equals the block-size t in the definition of PACT) for any practical AE. Further, the CMT security of PACT is bounded by the collision security of the hash function that it employs. One can expect to break this with about $2^{|K^*|/2}$ operations using a birthday attack, where $|K^*|$ is the key-size of E. Hence, roughly, PACT guarantees $\min\{\frac{|S|}{2}, \frac{|K^*|}{2}, \frac{t}{2}\}$ bits of CMT security.

It is also crucial to ensure that a secure AE scheme Π retains its traditional AE security properties after undergoing the PACT transform. In Theorems 2 and 3, we demonstrate that PACT maintains privacy security and authenticity security, respectively, thus preserving the original AE security of Π.

Theorem 2. *Let $\mathcal{A}$ be a privacy adversary of $\mathsf{PACT}[\Pi, \mathcal{H}, E]$ making q encryption queries and q_C ideal cipher queries. Then we can construct a collision adversary $\mathcal{B}$ for $\mathcal{H}$ and a privacy adversary $\mathcal{B}'$ for Π making q encryption queries to its own oracle, such that*

$$\mathbf{Adv}^{\mathrm{AE}_{\mathrm{PRIV}}}_{\mathsf{PACT}[\Pi,\mathcal{H},E]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{Coll}}_{\mathcal{H}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{AE}_{\mathrm{PRIV}}}_{\Pi}(\mathcal{B}') + \frac{q_C}{2^k} + \frac{2q^2}{2^t},$$

where k and t are the key-size and tag-size of Π respectively. The computation time of $\mathcal{B}$ and $\mathcal{B}'$ don't exceed that of $\mathcal{A}$ by more than a constant.

The proof of Theorem 2 is deferred to Section 7.2.

Theorem 3. *Let $\mathcal{A}$ be an authenticity adversary of $\mathsf{PACT}[\Pi, \mathcal{H}, E]$ making q_1 forging attempts and q_2 ideal cipher queries. Then we can construct a collision adversary $\mathcal{B}$ for $\mathcal{H}$ and an authenticity adversary $\mathcal{B}'$ of Π making q_2 forging attempts, such that*

$$\mathbf{Adv}^{\mathrm{AE}_{\mathrm{AUTH}}}_{\mathsf{PACT}[\Pi,\mathcal{H},E]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{Coll}}_{\mathcal{H}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{AE}_{\mathrm{AUTH}}}_{\Pi}(\mathcal{B}') + \frac{2q_1}{2^t}.$$

where t is the tag-size of Π. The computation time of both $\mathcal{B}$ and $\mathcal{B}'$ don't exceed that of $\mathcal{A}$ by more than a constant.

The proof of Theorem 3 is deferred to Section 7.3. While we prove these theorems for a nonce-based AE, it's evident from the proofs that the nonce doesn't have a unique role. Therefore, these proofs can also apply to DAE schemes.

5.1 CMT Security: PACT vs Others

We'd like to point out that from CMT security point of view, PACT performs on par with other existing transforms. HtE and CTX guarantee at most $t/2$ bits of CMT security. As already mentioned above, PACT guarantees $\min\{|S|/2, |K^*|/2, t/2\}$ bits of CMT security.

Firstly, the state-size $|S|$ is the block-size for block cipher-based AE schemes, and on the permutation-size for public permutation-based AE schemes. Since the size of the permutations commonly used to design AE schemes is generally greater than the tag-size, this does not worsen our bound. Moreover, for block cipher-based AE schemes, the tag-size is usually not larger than the block-size. Hence, typically, $|S| \geq t$. Therefore, our bound is not worse in that case either.

Secondly, for all practical block ciphers the key-size $|K^*|$ is at least as large as the block-size t. Hence, $|K^*| \geq t$. Therefore, for all practical AEs, PACT also guarantees $t/2$ bits of CMT security.

5.2 Efficiency Comparison with HtE and CTX

PACT uses a call to a hash function $\mathcal{H}$ with K, N and A, and a call to a block cipher with T using $\mathcal{H}(K, N, A)$ as the key. While HtE uses a call to a hash function with K, N and A, CTX uses a call to a hash function with K, N, A and T. If we compare PACT to HtE or CTX, the only additional cost of PACT is one single call to a block cipher, in place of processing one additional block in the hash function. On the other hand, PACT requires only the collision-resistance assumption on $\mathcal{H}$, while both HtE and CTX require stronger assumptions on their respective hashes—a PRF assumption for HtE and a random oracle assumption for CTX, so it's possible to potentially instantiate PACT with a lighter hash function. Overall PACT does not lose out in efficiency when compared to HtE and CTX, while having wider applicability.

comPACT$[\Pi, \mathcal{H}, E]$.Enc(K, N, A, M)

1: $(C, T) \leftarrow \Pi.\mathsf{Enc}(K, N, \epsilon, M)$
2: $K^* \leftarrow \mathcal{H}(K, N, A)$
3: $T^* \leftarrow E(K^*, T)$
4: **return** (C, T^*)

comPACT$[\Pi, \mathcal{H}, E]$.Dec(K, N, A, C, T^*)

1: $K^* \leftarrow \mathcal{H}(K, N, A)$
2: $T \leftarrow E^{-1}(K^*, T^*)$
3: $X \leftarrow \Pi.\mathsf{Dec}(K, N, \epsilon, C, T)$
4: **return** X

Figure 7: Specification of comPACT

5.3 comPACT for UNAE Schemes

PACT is designed to transform an AE scheme to a CMT secure one. As we have already mentioned, it is a universal transform which is applicable to any AE scheme. But we can make PACT even more efficient if we don't need to preserve the MRAE security of the legacy AE. We call this variant comPACT. If the legacy AE is UNAE secure, it remains so after the comPACT transform. comPACT differs from PACT in the way it processes the associated data. In PACT, the associated data goes into both the legacy AE as well as into the hash function of PACT. In comPACT, the associated data is processed only by the hash function. The legacy AE is called with empty associated data. The CMT security of PACT also holds for comPACT, as the security analysis is similar. The UNAE privacy and authenticity security analyses also carry over without any major changes. Fig. 7 gives the complete specification of comPACT.

Comparison between comPACT and CTY. The CTX transform processes the associated data (AD) twice: once for encryption and once for hashing. The CTY transform is a faster variant, which encrypts with empty AD and processes the AD only for hashing. Since comPACT is derived from PACT in a similar manner, it is natural to compare CTY and comPACT. We note the nonce-misuse attack on CTY also holds for comPACT. Therefore, in terms of security, both transforms offer similar guarantees, while comPACT has the advantage of being more widely applicable than CTY.

While still limited to non-TDD schemes, SC $\circ$ CTY achieves better CMT security than comPACT, but this comes at a significant loss of efficiency, since the SC layer requires a call to an 'invertible PRF' (IPF) in addition to the entire CTY layer. The proposed IPF HtM [BH24] requires, for instance, two additional evaluations of a cryptographic hash function.

6 CMT_k Attacks on TDD Schemes

In this section, we seek to support our proposal by illustrating constant time CMT_k attacks on various AE schemes. Each of these schemes exhibits characteristics that make them unsuitable for CTX transform, as their decryption process relies on the tag. Moreover, none of these schemes adhere to the CAU-SIV paradigm. Consequently, neither HtE nor CTX can effectively be applied to any of them.

Our CMT_k attacks fall into two main categories. In some AE schemes, we expose vulnerabilities in the scheme that utilise polynomial hashing for the MAC, with the hash key being independent of the encryption key. These schemes are prone to CMT_k attacks whenever a CMT_k adversary can exploit the hash, such as by finding a collision. We expose such a vulnerability for GCM-SIV in Section 6.1. In certain other schemes, a CMT_k adversary can backtrack from the tag, they can modify a few blocks of nonce or associated data to launch a CMT_k attack. We give such an attack for Deoxys-II in Section 6.2. In Section A, we give CMT_k attacks to some more AE schemes from both categories.

6.1 GCM-SIV

Fig. 10 gives the encryption algorithm of GCM-SIV [GL15], a widely used MRAE scheme. GCM-SIV uses three different keys: K_1 for GHASH (which is a polynomial hash), K_2 for the block cipher call on the GHASH output, and K_3 for the block cipher calls in the counter mode encryption. The authors suggest three instances of GCM-SIV. A three-key instance (GCM-SIV$_3$) where K_1, K_2 and K_3 are independently chosen, a two-key instance (GCM-SIV$_2$) where K_1 and K_2 are independently chosen and $K_3 = K_2$, and finally a single-key instance (GCM-SIV$_1$) where a single key K_0 is chosen first, then K_1 and K_2 are derived from K_0 by encrypting 0^{128} and $0^{127}||1$ respectively with AES instantiated with K_0, and $K_3 = K_2$. We observe that both GCM-SIV$_3$ and GCM-SIV$_2$ are vulnerable to a common CMT$_\text{k}$ attack, and the fix by Bellare and Hoang (which converts CAU-SIV to CAU-SIV-C1) [BH22] does not work for either of them. Note that GCM-SIV$_1$ is not vulnerable to this attack, since K_1 and K_2 are not independent.

Let $\mathcal{A}$ be a CMT$_\text{k}$ adversary attacking GCM-SIV$_i$ (where $i \in \{2, 3\}$). It proceeds as follows.

1. $\mathcal{A}$ computes (C, T) from (K_1, K_2, K_3, N, A, M).
2. $\mathcal{A}$ chooses $K_1' \neq K_1$ and A', and calculates N' such that

$$N' = \mathsf{GHASH}_{K_1}(\mathsf{Encode}(A, M)) \oplus \mathsf{GHASH}_{K_1'}(\mathsf{Encode}(A', M)) \oplus N,$$

   where M is the same as the first query, and Encode is the GCM encoding function.
3. $\mathcal{A}$ outputs (K_1, K_2, K_3, N, A, M) and $(K_1', K_2, K_3, N', A', M)$.

6.2 Deoxys-II and Joltik$^=$

Now, we move on to some schemes where a CMT$_\text{k}$ adversary needs to exploit the structure of the schemes to produce an attack. SCT (Synthetic Counter-in-Tweak) or Joltik$^=$ [JNP19], and SCT-2 or Deoxys-II [JNPS21, JNPS19] were submitted to the CAESAR competition [CAE19], where Joltik$^=$ was selected as a second-round candidate, and Deoxys-II was selected as the first choice in the final portfolio for defence-in-depth. Both these constructions follow the NSIV(nonce-based SIV) paradigm [PS16] and are MRAE schemes. Here, we show a CMT$_\text{k}$ attack on Deoxys-II. A similar attack also holds for Joltik$^=$. Fig. 4 gives the encryption algorithm of Deoxys-II.

Let $\mathcal{A}$ be a CMT$_\text{k}$ adversary attacking Deoxys-II. It proceeds as follows.

1. $\mathcal{A}$ computes (C, T) from (K, N, A, M). Let $C = C[1]\| \ldots \|C[m]$.
2. $\mathcal{A}$ chooses K' and N', and fixes $M' = M'[1]\| \ldots \|M'[m]$ in a way such that the XOR of $M'[i]$ and the encryption of $0\|N'$ yields $C[i]$. Formally, $M'[i] = \tilde{E}_{K'}(1\|T \oplus (i-1), 0^8\|N') \oplus C[i](\forall i \in [m])$.
3. $\mathcal{A}$ chooses all but one of the blocks of A' (say, the first block $A'[1]$), and makes forward queries to $\tilde{E}$ with all the associated data and message blocks as inputs (except $A'[1]$). Let X be the xor of all the outputs from $\tilde{E}$, and Y be the output of the backward query to $\tilde{E}$ with T as the input.
4. $\mathcal{A}$ makes a backward query to $\tilde{E}$ with $X \oplus Y$ as the input and sets the output as $A'[1]$.
5. $\mathcal{A}$ outputs (K, N, A, M) and (K', N', A', M').

7 Security Proofs

7.1 Proof of Theorem 1

In this section, we investigate the CMT security of PACT. We assume the adversary does not make any pointless queries, which include:

- repeated queries to Π or E;
- a query (K, N, A, C, T) to $\Pi.\mathsf{Dec}$ following a query (K, N, A, M) to $\Pi.\mathsf{Enc}$ that returned (C, T);
- a query (K, N, A, M) to $\Pi.\mathsf{Enc}$ following a query (K, N, A, C, T) to $\Pi.\mathsf{Dec}$ that returned M;
- a query (K^*, T^*) to E^{-1} following a query (K^*, T) to E that returned T^*;
- a query (K^*, T) to E following a query (K^*, T^*) to E^{-1} that returned T.

First, we list the adversaries we construct to simulate $\mathcal{A}$.

- $\mathcal{B}_1$: A collision adversary of $\mathcal{H}$. It outputs a colliding $\mathcal{H}$ input pair of $\mathcal{A}$, if any such pair exists. Otherwise, it outputs any two random $\mathcal{H}$ inputs.
- $\mathcal{B}_2$: A $(q_C, q, 2/2^t)$-ciphertext collision adversary of Π. Whenever $\mathcal{A}$ queries $T_j^* = E(K_j^*, T_j)$ where $K_i^* = \mathcal{H}(K_i, N_i, A_i)$ and $T_j^* \neq T_k^*$ for $k \in [j-1]$, it makes a new list $\mathcal{L}[T_j^*]$, and submits (K_j, N_j, A_j, T_j). Whenever $\mathcal{A}$ queries $E^{-1}(K_{j'}^*, T_j^*)$ where $K_{i'}^* = \mathcal{H}(K_{i'}, N_{i'}, A_{i'})$, $\mathcal{B}_2$ includes $(K_{j'}, N_{j'}, A_{j'})$ in $\mathcal{L}[T_j^*]$ and submits the same, receives $T_{j'} = E^{-1}(K_{j'}^*, T_j^*)$, and forwards it to $\mathcal{A}$. Finally, if $\mathcal{A}$ makes any two Π computations to obtain the tuples $(K_y, N_y, A_y, M_y, C, T_y)$ and $(K_z, N_z, A_z, M_z, C, T_z)$ where both the tuples are in the same list $\mathcal{L}[T_x^*]$ and $y < z$, $\mathcal{B}_2$ outputs M_y, M_z, x, y and z. Otherwise, $\mathcal{B}_2$ aborts.

Next, we list the bad events.

- B1: $\mathcal{B}_1$ wins.

$$\Pr[\mathsf{B1}] \leq \mathbf{Adv}_{\mathcal{H}}^{\textsc{Coll}}(\mathcal{B}_1).$$

- B2: $\mathcal{B}_2$ wins.

$$\Pr[\mathsf{B2}] \leq \mathbf{Adv}_{\Pi}^{\textsc{CC}}(\mathcal{B}_2).$$

- B3: $\mathcal{A}$ makes two E queries to obtain the tuples (K, X, Y) and (K', X', Y) such that $\mathcal{A}$ obtains (K', X', Y) after it obtains (K, X, Y), and through a forward query (i.e., $\mathcal{A}$ queries $Y = E(K', X')$). For a fixed pair of queries, the probability of this event can be upper-bounded by $1/(2^t - q_C)$. Because such a pair of E queries can be chosen in $\binom{q_C}{2}$ ways, and $q_C < 2^{t-1}$, applying union bound we obtain

$$\Pr[\mathsf{B3}] \leq 2\binom{q_C}{2}/2^t < 2q_C^2/2^t.$$

- B4: $\mathcal{A}$ makes one Π computation to obtain an input-output tuple (K, N, A, M, C, T), and after that, makes an inverse query to E with K' and T', such that $K \neq K'$ and $E^{-1}(K', T') = T$. As we model E as an ideal cipher, for a fixed Π computation and a fixed E query, the probability of this event can be upper-bounded by $1/(2^t - q_C)$. As $\mathcal{A}$ can make at most q computations of Π and q_C queries to E, applying union bound we obtain

$$\Pr[\mathsf{B4}] \leq 2qq_C/2^t.$$

Suppose $\mathcal{A}$ submits (K_1, N_1, A_1, M_1) and (K_2, N_2, A_2, M_2) with $(K_1, N_1, A_1) \neq (K_2, N_2, A_2)$ and wins. W.l.o.g., we assume that if $\mathcal{A}$ has made either $E(K_1^*, T_1)$ or $E^{-1}(K_1^*, T^*)$, as well as either $E(K_2^*, T_2)$ or $E^{-1}(K_2^*, T^*)$, it has made them in that order, i.e., the latter is made after the former. Then the following must hold for some C, T^*, T_1, T_2, K_1^*, and K_2^*:

$$\begin{aligned}(C, T_1) &= \Pi.\mathsf{Enc}(K_1, N_1, A_1, M_1),\\ (C, T_2) &= \Pi.\mathsf{Enc}(K_2, N_2, A_2, M_2),\\ K_1^* &= \mathcal{H}(K_1, N_1, A_1),\\ K_2^* &= \mathcal{H}(K_2, N_2, A_2),\\ T^* &= E(K_1^*, T_1) = E(K_2^*, T_2).\end{aligned}$$

Next, we list the events, one of which must happen.

- E1: $\mathcal{A}$ makes both the queries (1) $T^* = E(K_1^*, T_1)$ or $T_1 = E^{-1}(K_1^*, T^*)$, and (2) $T^* = E(K_2^*, T_2)$ or $T_2 = E^{-1}(K_2^*, T^*)$.
  - E11: $\mathcal{A}$ queries $E(K_2^*, T_2)$.
  - E12: $\mathcal{A}$ queries $E^{-1}(K_2^*, T^*)$.
    - E121: $(K_1, N_1, A_1, M_1, C, T_1) \in \Pi$, $(K_2, N_2, A_2, M_2, C, T_2) \in \Pi$.
    - E122: $(K_2, N_2, A_2, M_2, C, T_2) \in \Pi$, $(K_1, N_1, A_1, M_1, C, T_1) \notin \Pi$.
    - E123: $(K_1, N_1, A_1, M_1, C, T_1) \in \Pi$, $(K_2, N_2, A_2, M_2, C, T_2) \notin \Pi$.
    - E124: $(K_1, N_1, A_1, M_1, C, T_1) \notin \Pi$, $(K_2, N_2, A_2, M_2, C, T_2) \notin \Pi$.
- E2: $\mathcal{A}$ doesn't make both the queries (1) $T^* = E(K_1^*, T_1)$ or $T_1 = E^{-1}(K_1^*, T^*)$, and (2) $T^* = E(K_2^*, T_2)$ or $T_2 = E^{-1}(K_2^*, T^*)$.

Next, we analyse the winning probability of $\mathcal{A}$.

- E11: This event is impossible because B3 doesn't happen.
- E121: $K_1^* = K_2^*$ is impossible as B1 doesn't happen. And if $K_1^* \neq K_2^*$, then this event is impossible as B4 doesn't happen.
- E122: This event is impossible, and the argument is the same as E121.
- E123: If $\mathcal{A}$ makes a forward query $T^* = E(K_1^*, T_1)$, then a previous forward query $T^* = E(K, T)$ by $\mathcal{A}$ is not possible because B3 doesn't happen. Otherwise, E123 is impossible as B2 doesn't happen.
- E124: This event is impossible, and the argument is the same as E123.

$E[\mathscr{T}](K^*, T)$

$T^* \leftarrow_\$ \{0,1\}^t \setminus \mathscr{T}[K^*, \cdot]$
if $\mathscr{T}[K^*, T] \neq \perp$ **then**
  $T^* \leftarrow \mathscr{T}[K^*, T]$
else
  $\mathscr{T}[K^*, T] \leftarrow T^*$
end if
return T^*

$\mathsf{PACT}[\Pi, \mathcal{H}, \mathscr{T}, K](N, A, M)$

$(C, T) \leftarrow \Pi.\mathsf{Enc}(K, N, A, M)$
$T^* \leftarrow_\$ \{0,1\}^t \setminus \mathscr{T}[\mathcal{H}(K, N, A), \cdot]$
if $\mathscr{T}[\mathcal{H}(K, N, A), T] \neq \perp$ **then**
  $T^* \leftarrow \mathscr{T}[\mathcal{H}(K, N, A), T]$
else
  $\mathscr{T}[\mathcal{H}(K, N, A), T] \leftarrow T^*$
end if
return (C, T^*)

Figure 8: Lazy-sampled versions of E and PACT, using a shared table.

$\mathsf{PACT}^\circ[\Pi, \mathscr{T}^\circ, K](N, A, M)$

$(C, T) \leftarrow \Pi.\mathsf{Enc}(K, N, A, M)$
$T^* \leftarrow_\$ \{0,1\}^t \setminus \mathscr{T}^\circ[N, A, \cdot]$
if $\mathscr{T}^\circ[N, A, T] \neq \perp$ **then**
  $T^* \leftarrow \mathscr{T}^\circ[N, A, T]$
else
  $\mathscr{T}^\circ[N, A, T] \leftarrow T^*$
end if
return (C, T^*)

$\mathsf{PACT}^\dagger[\mathscr{T}^\circ](N, A, M)$

$(C, T) \leftarrow_\$ \{0,1\}^{|M|+t}$
$T^* \leftarrow_\$ \{0,1\}^t \setminus \mathscr{T}^\circ[N, A, \cdot]$
if $\mathscr{T}^\circ[N, A, T] \neq \perp$ **then**
  $T^* \leftarrow \mathscr{T}^\circ[N, A, T]$
else
  $\mathscr{T}^\circ[N, A, T] \leftarrow T^*$
end if
return (C, T^*)

$\mathsf{PACT}^\ddagger[\mathscr{T}^\circ](N, A, M)$

$(C, T) \leftarrow_\$ \{0,1\}^{|M|+t}$
$T^* \leftarrow_\$ \{0,1\}^t \setminus \mathscr{T}^\circ[N, A, \cdot]$
$\mathscr{T}^\circ[N, A, T] \leftarrow T^*$
return (C, T^*)

$\mathsf{PACT}^\star(N, A, M)$

$(C, T) \leftarrow_\$ \{0,1\}^{|M|+t}$
$T^* \leftarrow_\$ \{0,1\}^t$
return (C, T^*)

Figure 9: The hybrid construction oracles used in the proof of Theorem 2.

- E2: W.l.o.g., suppose $E(K_1^*, T_1)$ or $E^{-1}(K_1^*, T^*)$ is not made by $\mathcal{A}$. Then the probability of the event $E(K_1^*, T_1) = T^*$ (in case $E(K_2^*, T_2)$ or $E^{-1}(K_2^*, T^*)$ is made by $\mathcal{A}$, so that T^* is fixed) or $E(K_1^*, T_1) = E(K_2^*, T_2)$ (in case $E(K_2^*, T_2)$ or $E^{-1}(K_2^*, T^*)$ is not made by $\mathcal{A}$) is upper-bounded by $1/(2^t - q_C)$. As $\mathcal{A}$ can query E maximum q_C times, and $q_C < 2^{t-1}$, we obtain by applying union bound

$$\Pr[\mathcal{A} \text{ wins}] \leq 2q_C/2^t.$$

The final upper bound of the winning probability of $\mathcal{A}$ is obtained by adding the probabilities of all the bad events and the winning probability of $\mathcal{A}$ in the case of E2. □

7.2 Proof of Theorem 2

We bound the privacy advantage of $\mathcal{A}$ against $\mathsf{PACT}[\Pi, \mathcal{H}, E]$ through a series of games. In game $\mathcal{G}_0$, we begin with lazy-sampled versions of PACT and the ideal cipher E, such that PACT doesn't make oracle calls to E, but instead they share the same table $\mathscr{T}$ (Fig. 8). The adversary plays against $(\mathsf{PACT}[\Pi, \mathcal{H}, \mathscr{T}, K], E[\mathscr{T}])$, where $K \leftarrow_\$ \mathcal{K}$. In the subsequent games, we transform $\mathsf{PACT}[\Pi, \mathcal{H}, \mathscr{T}, K]$ through a series of hybrids which culminate in $\mathsf{PACT}^\star$ in game $\mathcal{G}_5$, which is identical in output distribution to Π^*. The

Game	Construction Oracle
$\mathcal{G}_0$	$\mathsf{PACT}[\Pi, \mathcal{H}, \mathscr{T}, K]$
$\mathcal{G}_1$	$\mathsf{PACT}[\Pi, \mathcal{H}, \mathscr{T}', K]$
$\mathcal{G}_2$	$\mathsf{PACT}^{\circ}[\Pi, \mathscr{T}^{\circ}, K]$
$\mathcal{G}_3$	$\mathsf{PACT}^{\dagger}[\mathscr{T}^{\circ}]$
$\mathcal{G}_4$	$\mathsf{PACT}^{\ddagger}[\mathscr{T}^{\circ}]$
$\mathcal{G}_5$	$\mathsf{PACT}^{\star}$

Table 1: The adversary's construction oracles corresponding to the different games proof of Theorem 2. In each game the adversary also queries $E[\mathscr{T}]$. In $\mathcal{G}_0$ the table $\mathscr{T}$ is shared between PACT and E; in $\mathcal{G}_1$ it is replaced in PACT by an identical but independent table $\mathscr{T}'$. In the first three games $K \leftarrow_{\$} \mathcal{K}$ as part of the initialisation.

different hybrid construction oracles used are described in Fig. 9. Table 1 lists the series of games and the corresponding construction oracles.

In game $\mathcal{G}_1$, PACT remains unchanged, but uses an independent table $\mathscr{T}'$. Suppose $\mathcal{A}_1$ is an adversary trying to distinguish between $\mathcal{G}_0$ and $\mathcal{G}_1$. Since the different rows $\mathscr{T}[K^*, \cdot]$ of $\mathscr{T}$ are always sampled independently, the games $\mathcal{G}_0$ and $\mathcal{G}_1$ are identical unless at some point in $\mathcal{G}_0$, PACT° and E access the same row of $\mathscr{T}^{\circ}$; this can only happen if $\mathcal{A}_1$ can correctly guess the construction key K and sets $K^* = \mathcal{H}(K, N, A)$ for some N, A in one of its queries to E, so we have

$$\mathbf{Adv}_{\mathcal{G}_0,\mathcal{G}_1}(\mathcal{A}_1) \leq \frac{q_C}{2^k}.$$

In game $\mathcal{G}_2$, we replace PACT with PACT° which doesn't make calls to $\mathcal{H}$ and uses a modified table $\mathscr{T}^{\circ}$ where the rows are indexed with (N, A) instead of $\mathcal{H}(K, N, A)$. Let $\mathcal{A}_2$ be an adversary trying to distinguish between $\mathcal{G}_1$ and $\mathcal{G}_2$. Let $\mathcal{B}$ be a collision adversary which simulates both the challenger and $\mathcal{A}_2$; thus $\mathcal{B}$ knows K and all the online queries (N, A, M) of $\mathcal{A}_2$, and computes $\mathcal{H}(K, N, A)$ for each such online query. If any two of these inputs X_1 and X_2 have $\mathcal{H}(X_1) = \mathcal{H}(X_2)$, $\mathcal{B}$ outputs (X_1, X_2), and otherwise $\mathcal{B}$ outputs a pair of random hash inputs (Z_1, Z_2). Then the games $\mathcal{G}_1$ and $\mathcal{G}_2$ are identical unless $\mathcal{B}$ wins. Thus,

$$\mathbf{Adv}_{\mathcal{G}_1,\mathcal{G}_2}(\mathcal{A}_2) \leq \mathbf{Adv}^{\textsc{Coll}}_{\mathcal{H}}(\mathcal{B}).$$

In game $\mathcal{G}_3$, we replace PACT° with $\mathsf{PACT}^{\dagger}$, which samples (C, T) randomly instead of calling Π.Enc. Let $\mathcal{A}_3$ be an adversary trying to distinguish between $\mathcal{G}_2$ and $\mathcal{G}_3$. Then we construct a privacy adversary $\mathcal{B}'$ for Π as follows: $\mathcal{B}'$ simulates $\mathcal{A}_3$ and lets $\mathcal{A}_3$ play against $(\mathsf{PACT}^{?}[\mathscr{T}^{\circ}], E^{\circ})$, where $\mathsf{PACT}^{?}$ is identical to PACT° or $\mathsf{PACT}^{\dagger}$ except that the first line is replaced by a call to $\mathcal{B}'$'s own oracle. $\mathcal{B}'$ replicates the output bit of $\mathcal{A}_3$. Then we have

$$\mathbf{Adv}_{\mathcal{G}_2,\mathcal{G}_3}(\mathcal{A}_3) \leq \mathbf{Adv}^{\textsc{AEpriv}}_{\Pi}(\mathcal{B}').$$

In game $\mathcal{G}_4$, we replace $\mathsf{PACT}^{\dagger}$ with $\mathsf{PACT}^{\ddagger}$, which updates the table $\mathscr{T}^{\circ}$ without checking for previous entries. Let $\mathcal{A}_4$ be an adversary trying to distinguish between $\mathcal{G}_3$ and $\mathcal{G}_4$. The two games are identical unless the if-clause in $\mathsf{PACT}^{\dagger}$ is executed in $\mathcal{G}_3$ for some query, which can only happen if a randomly sampled T collides with an earlier T' for the same (N, A). Thus we have

$$\mathbf{Adv}_{\mathcal{G}_3,\mathcal{G}_4}(\mathcal{A}_4) \leq \frac{q^2}{2^t}.$$

In game $\mathcal{G}_5$, we replace $\mathsf{PACT}^{\ddagger}$ with $\mathsf{PACT}^{\star}$, which does not maintain a table and samples T^* uniformly at random. Let $\mathcal{A}_5$ be an adversary trying to distinguish between $\mathcal{G}_4$ and $\mathcal{G}_5$. The two games are identical unless a randomly sampled T^* collides with an earlier T'^* for the same (N, A). Thus we have

$$\mathbf{Adv}_{\mathcal{G}_4,\mathcal{G}_5}(\mathcal{A}_5) \leq \frac{q^2}{2^t}.$$

If $\mathcal{A}$ wins the privacy game against $\mathsf{PACT}[\Pi, \mathcal{H}, E]$, at least one of $\mathcal{A}_1, \ldots, \mathcal{A}_5$ must win. Thus we have

$$\begin{aligned}\mathbf{Adv}^{\mathrm{AE}_{\mathrm{PRIV}}}_{\mathsf{PACT}[\Pi,\mathcal{H},E]}(\mathcal{A}) &\leq \sum_{i=1}^{5} \mathbf{Adv}_{\mathcal{G}_{i-1},\mathcal{G}_i}(\mathcal{A}_i) \\ &\leq \mathbf{Adv}^{\mathrm{Coll}}_{\mathcal{H}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{AE}_{\mathrm{PRIV}}}_{\Pi}(\mathcal{B}') + \frac{q_C}{2^k} + \frac{2q^2}{2^t}.\end{aligned}$$

This completes the proof. □

7.3 Proof of Theorem 3

Both $\mathcal{B}'$ and $\mathcal{B}$ simulate $\mathcal{A}$. $\mathcal{B}$ looks at the hash computations as $\mathcal{A}$, and if $\mathcal{A}$ finds two hash inputs X_1 and X_2 with $\mathcal{H}(X_1) = \mathcal{H}(X_2)$, $\mathcal{B}$ outputs (X_1, X_2), and otherwise $\mathcal{B}$ outputs a pair of random hash inputs (Z_1, Z_2). $\mathcal{B}'$ proceeds as described below.

Let the ℓ-th forging attempt of $\mathcal{A}$ be $(N_\ell, A_\ell, C_\ell, T^*_\ell)$ (where $\ell \in [q_1]$). For each $\ell \in [q_1]$, let $\mathcal{L}_\ell$ be the list of all ideal cipher query-response tuples of the form $(K^*_{\ell,i}, T_{\ell,i}, T^*_\ell)$, such that for each $i \in [|\mathcal{L}_\ell|]$ there is a hash computation with input $(K_{\ell,i}, N_\ell, A_\ell)$ and output $K^*_{\ell,i}$. For each $\ell \in [q_1]$, $i \in [|\mathcal{L}_\ell|]$, if $\Pi.\mathsf{Dec}(K_{\ell,i}, N_\ell, A_\ell, C_\ell, T_{\ell,i}) \neq \perp$, then $\mathcal{B}'$ makes a forging attempt $(N_\ell, A_\ell, C_\ell, T_{\ell,i})$.

Let K be the secret key of Π. Suppose the forging attempt $(N_\ell, A_\ell, C_\ell, T^*_\ell)$ by $\mathcal{A}$ is successful. This can only happen when at least one of the following four events occurs:

- $\mathsf{B1}$: $\mathcal{A}$ finds a collision-pair for $\mathcal{H}$. When this happens, $\mathcal{B}$ wins, so

$$\Pr[\mathsf{B1}] \leq \mathbf{Adv}^{\mathrm{Coll}}_{\mathcal{H}}(\mathcal{B}).$$

- $\mathsf{B2}$: $(N_\ell, A_\ell, C_\ell, T)$ is a corresponding forging attempt by $\mathcal{B}'$, and $K_{\ell,i} = K$ for $T_{\ell,i} = T$. In this case, whenever $\mathcal{A}$ wins, $\mathcal{B}'$ also wins. Also note that $\mathcal{B}'$ makes a maximum of q_2 forging attempts with this strategy; this is because each pair $(T^*_\ell, K^*_{\ell,i})$ can lead to at most one forging attempt. So we obtain

$$\Pr[\mathsf{B2} \mid \neg\mathsf{B1}] \leq \mathbf{Adv}^{\mathrm{AE}_{\mathrm{AUTH}}}_{\Pi}(\mathcal{B}').$$

- $\mathsf{B3}$: $(N_\ell, A_\ell, C_\ell, T)$ is a corresponding forging attempt by $\mathcal{B}'$, but $K_{\ell,i} \neq K$ for $T_i = T$. In this case, the forgery by $\mathcal{A}$ is accidental, i.e., $\mathcal{A}$ wins by random guess of the tag. Hence

$$\Pr[\mathsf{B3} \mid \neg\mathsf{B1}] \leq \frac{q_1}{2^t}.$$

- $\mathsf{B4}$: $\mathcal{B}'$ makes no corresponding forging attempt. This follows the same reasoning as $\mathsf{B2}$, so

$$\Pr[\mathsf{B4} \mid \neg\mathsf{B1}] \leq \frac{q_1}{2^t}.$$

The final upper bound of the winning probability of $\mathcal{A}$ is obtained by applying union-bound over the four events. □

8 Conclusion and Future Works

In this work, we proposed PACT, a CMT transform for any authenticated encryption. PACT is output-length-preserving, works on any AE scheme and preserves both UNAE and MRAE security of the legacy AE scheme. PACT uses only one call to a collision-resistant unkeyed hash function and one call to a block cipher, so it's efficient from an implementation point of view. We also proposed comPACT, a lighter version of PACT which preserves UNAE security.

Designing generic length-preserving transforms which can provide CMT_k security with a relatively lighter design can be an interesting research challenge. Also of interest will be to find a transform which does not use dynamic keys on the underlying block cipher or hash function, and hence lends itself to security proofs in the standard model.

Appendix

A CMT_k Attacks On CTX-unfriendly AE Schemes (Continued)

Here we continue our discussion from Section 6 on illustrating constant time CMT_k attacks on various AE schemes where neither HtE nor CTX can be applied.

A.1 GCM-SIVr

In GCM-SIVr [IM16] which runs r instances of GCM-SIV in parallel, all the keys are independent. So GCM-SIV1 is vulnerable to the same attack, which works on GCM-SIV_3 and GCM-SIV_2. For GCM-SIVr with $r > 1$, the CMT_k adversary $\mathcal{A}$ proceeds as follows.

1. $\mathcal{A}$ computes (C, T) from (K_1, K_2, K_3, N, A, M). Let $K_i = (K_{i,1}, \ldots, K_{i,r})$ for $i = 1, 3$ and $K_2 = (K_{2,1}, \ldots, K_{2,r^2})$.
2. $\mathcal{A}$ chooses $K'_1 \neq K_1$ and all but the first r blocks of A', and calculates them from the following set of r equations.

$$\begin{aligned}
\mathsf{GHASH}_{K_{1,1}}(\mathsf{Encode}(A, M)) &= \mathsf{GHASH}_{K'_{1,r}}.(\mathsf{Encode}(A', M)) \\
&\vdots \\
\mathsf{GHASH}_{K_{1,r}}(\mathsf{Encode}(A, M)) &= \mathsf{GHASH}_{K'_{1,r}}.(\mathsf{Encode}(A', M))
\end{aligned}$$

3. $\mathcal{A}$ outputs (K_1, K_2, K_3, N, A, M) and $(K'_1, K_2, K_3, N, A', M)$.

A.2 SCM

Synthetic Counter with Masking, or SCM [CLLL21] is yet another MRAE scheme following the NSIV paradigm. Fig. 11 gives the encryption algorithm of SCM. We show a CMT_k attack on SCM, when it is instantiated with an ϵ-AXU polynomial hash. Let $\mathcal{A}$ be a CMT_k adversary attacking SCM. It proceeds as follows.

1. $\mathcal{A}$ computes (C, T) from (K, N, A, M), where $K = (K_1, \ldots, K_4)$.

GCM-SIV$[E, \mathsf{GHASH}].\mathsf{Enc}(K_1, K_2, K_3, N, A, M = M[1]\| \dots \|M[m])$

$V \leftarrow \mathsf{GHASH}_{K_1}(\mathsf{Encode}(A, M)) \oplus N$ ▷ Encode is the GCM encoding function
$T \leftarrow E_{K_2}(V)$
$IV \leftarrow \lceil T \rceil_{n-k} \| 0^k$
$I[1] \leftarrow IV$
$Z[1] \leftarrow E_{K_3}(I[1])$
for $i \in [2..m]$ **do**
    $I[i] \leftarrow \mathsf{inc}(I[i-1])$ ▷ $\mathsf{inc}(X) := \lceil X \rceil_{n-32} \| ((\lfloor X \rfloor_{32} + 1) \mod 2^{32})$
    $Z[i] \leftarrow E_{K_3}(I[i])$
end for
$Z \leftarrow Z[1]\| \dots \|Z[m]$
return $(C := M \oplus Z, T)$

Figure 10: Encryption of GCM-SIV

SCM$[E, \mathcal{H}].\mathsf{Enc}(K_1, K_2, K_3, K_4, N, A, M = M[1]\| \dots \|M[m])$

$\Delta \leftarrow E_{K_3}(N\|00) \oplus E_{K_3}(N\|01)$
$\Delta' \leftarrow E_{K_3}(N\|00) \oplus E_{K_3}(N\|10)$
$\Delta'' \leftarrow E_{K_3}(N\|00) \oplus E_{K_3}(N\|11)$
$X \leftarrow \mathcal{H}_{K_1}(A, M) \oplus (N\|00)$ ▷ $\mathcal{H}$ is an ϵ-AXU hash
$T \leftarrow E_{K_2}(X) \oplus \Delta''$
for $i \in [1..m]$ **do**
    $X[i] \leftarrow T \oplus 2^{i-1}\Delta$
    $Z[i] \leftarrow E_{K_4}(X[i]) \oplus \Delta'$
end for
$Z \leftarrow Z[1]\| \dots \|Z[m]$
return $(C := M \oplus Z, T)$

Figure 11: Encryption of SCM

2. $\mathcal{A}$ chooses $K_1' \neq K_1$ and all but one of the blocks of A', (say, the first block $A'[1]$), and calculates $A'[1]$ from the equation

$$\mathcal{H}_{K_1}(A, M) = \mathcal{H}_{K_1'}(A', M).$$

3. $\mathcal{A}$ outputs (K, N, A, M) and (K', N, A', M), where $K' = (K_1', K_2, K_3, K_4)$.

A.3 DCT

We now have a look at a few DAE schemes, starting with Deterministic Counter in Tweak, or DCT [FLLW16]. Fig. 12 gives the encryption algorithm of DCT. A $\mathrm{CMT_k}$ attack similar to that on SCM also holds for DCT, when it is instantiated with an ϵ-AXU polynomial hash.

A.4 BCTR

Fig. 13 gives the encryption algorithm of BCTR [CLS18], which is a DAE scheme suitable for disk encryption. We first show a $\mathrm{CMT_k}$ attack on BCTR. In the case of BCTR, the $\mathrm{CMT_k}$ adversary $\mathcal{A}$ considers the

$\mathsf{DCT}[\mathcal{H}, E, \tilde{E}].\mathsf{Enc}(K_1, K_2, K_3, A, M)$

1: $(M_L, M_R \leftarrow \mathsf{Encode}(M))$
2: $X \leftarrow \mathcal{H}_{K_1}(A, M_R)$
3: $Y \leftarrow M_L \oplus X$
4: $C_L \leftarrow E_{K_2}(Y)$
5: $C_R \leftarrow \tilde{E}_{K_3}(C_L, M_R)$
6: **return** $(C := C_L \| C_R)$

Figure 12: Encryption of DCT

$\mathsf{BCTR}.\mathsf{Enc}[E, \mathsf{BRW}](K_1, K_2, \tau, M = M[1] \| \dots \| M[m])$

1: $X \leftarrow K_1 \cdot \mathsf{BRW}_{K_1}(M \| \tau)$
2: $T \leftarrow E_{K_2}(X)$
3: **for** $i \in [1..m]$ **do**
4: $\quad Z[i] \leftarrow E_{K_2}(T \oplus i)$
5: **end for**
6: **return** $(C := M \oplus Z, T)$

Figure 13: Encryption of BCTR

tuples (K_1, K_2, M, τ) and (K_1', K_2, M, τ') for the attack. It chooses the values for all the inputs except a single tweak block (say, the j-th block of τ, i.e., τ_j), and calculates its value from the equation

$$K_1 \cdot \mathsf{BRW}_{K_1}(M, \tau) = K_1' \cdot \mathsf{BRW}_{K_1'}(M, \tau').$$

A.5 SUNDAE and ANYDAE

We now have a look at SUNDAE [BBLT18], a lightweight DAE scheme, and its RUP secure modification ANYDAE [CDD+19]. Fig. 14 gives the encryption algorithm of SUNDAE. We first show a $\mathrm{CMT_k}$ attack on SUNDAE. Let $\mathcal{A}$ be a $\mathrm{CMT_k}$ adversary attacking SUNDAE. It proceeds as follows.

1. $\mathcal{A}$ computes (C, T) from (K, A, M).
2. $\mathcal{A}$ then chooses a $K' \neq K$, and fixes $M' = M'[1], \dots, M'[m]$ by setting $M'[1] = E^i_{K'}(T) \oplus C[i]$, where E^i denotes i compositions of E.
3. $\mathcal{A}$ then chooses all but one of the blocks of A', (say, the first block $A'[1]$), starts going backwards from T using inverse block cipher calls, and calculates $A'[1]$ such that the starting value 110^{n-2} is reached.
4. $\mathcal{A}$ outputs (K, A, M) and (K', A', M').

A similar attack also holds for ANYDAE, if the function ρ_1 is invertible. As a consequence, both MONDAE and TUESDAE, which are two instances of ANYDAE, are vulnerable to the attack.

SUNDAE$[E]$.Enc$(K, A = A[1]\| \dots \|A[a], M = M[1]\| \dots \|M[m])$

$T \leftarrow E_K(110^{n-2})$
for $i \in [1..a]$ **do**
  $T \leftarrow E_K(T \oplus A[i])$
end for
$T \leftarrow E_K(4 \cdot (T \oplus A[a]))$
for $i \in [1..m-1]$ **do**
  $T \leftarrow E_K(T \oplus M[i])$
end for
$T \leftarrow E_K(4 \cdot (X \oplus M[m]))$
$V \leftarrow T$
for $i \in [1..m]$ **do**
  $V \leftarrow E_K(V)$
  $C[i] \leftarrow M[i] \oplus V$
end for
return $(C := (C[1], \dots, C[m]), T)$

Figure 14: Encryption of SUNDAE

A.6 LM-DAE

Fig. 15 gives the encryption algorithm of LM-DAE [NSS20], another DAE scheme. We first show a CMT_k attack on LM-DAE. Let $\mathcal{A}$ be a CMT_k adversary attacking LM-DAE. It proceeds as follows.

1. $\mathcal{A}$ computes (C, T) from (K, A, M).
2. $\mathcal{A}$ chooses $K' \neq K$, and uses K', C and T to compute M'.
3. $\mathcal{A}$ chooses all but two two blocks of A' (say, $A'[1]$ and $A'[2]$).
4. $\mathcal{A}$ starts to compute towards the IV of the construction (i.e., 0^{2n}). Let the output of the penultimate $\tilde{E}^{-1}$ call be X, and the output of the penultimate π^{-1} call be Y.
5. $\mathcal{A}$ calculates $A'[1]$ and $A'[2]$ as follows.

$$A'[2] = X \oplus Y \oplus \pi(0^n),$$
$$\text{and, } A'[1] = \tilde{E}^{-1}(A'[2] \oplus X).$$

6. $\mathcal{A}$ outputs (K, A, M) and (K', A', M').

A.7 MRO, MRS and MRSO

Fig. 16 gives the encryption algorithm of MRO (Misuse Resistant Offset) [GJMN16]. We first show a CMT_k attack on MRO. Let $\mathcal{A}$ be a CMT_k adversary attacking MRO. It proceeds as follows.

1. $\mathcal{A}$ computes (C, T) from (K, N, A, M), where $C = C[1]\| \dots \|C[m]$.
2. $\mathcal{A}$ chooses $K' \neq K$ and N', defines $U' = (K', N')$, and computes $M'[i] = \tilde{E}_{U'}(0\|0\|1, T\|(i-1)) \oplus C[i]$, $\forall i \in [m]$.

LM-DAE$[\tilde{E}, \pi]$.Enc$(K, A = A[1]\| \dots \|A[a], M = M[1]\| \dots \|M[m])$

$h_t \leftarrow 0^n$
$h_b \leftarrow 0^n$
for $i \in [1..a-1]$ **do**
  $h_t \leftarrow \tilde{E}_K(1\|h_b, h_t \oplus A[i])$
  $h_b \leftarrow \pi(h_b) \oplus h_t$
end for
$h_t \leftarrow \tilde{E}_K(3\|h_b, h_t \oplus A[a])$
$h_b \leftarrow \pi(h_b) \oplus h_t$
for $i \in [1..m]$ **do**
  $h_t \leftarrow \tilde{E}_K(5\|h_b, h_t \oplus M[i])$
  $h_b \leftarrow \pi(h_b) \oplus h_t$
end for
$h_t \leftarrow \tilde{E}_K((5+6)\|h_b, h_t)$
$h_b \leftarrow \pi(h_b) \oplus h_t$
$h_t \leftarrow \tilde{E}_K((5+6)\|h_b, h_t)$
$h_b \leftarrow \pi(h_b) \oplus h_t$
$T \leftarrow h_t\|h_b$
$h_t[0] \leftarrow h_t$
$h_b[0] \leftarrow h_b$
for $i \in [1..m]$ **do**
  $h_t[i] \leftarrow \tilde{E}_K(0\|h_b[i-1], h_t[i-1])$
  $h_b[i] \leftarrow \pi(h_b[i-1]) \oplus h_t[i]$
end for
$Z \leftarrow h_t[1]\| \dots \|h_t[m]$
return $(C := M \oplus Z, T)$

Figure 15: Encryption of LM-DAE

3. $\mathcal{A}$ computes $W' = \tilde{E}^{-1}_{U'}(0\|2\|0, T\|0) \oplus an\|mn$.
4. $\mathcal{A}$ chooses all but one of the blocks of A' (say, the first block $A'[1]$), and computes $X'[i] = \tilde{E}_{U'}((i-1)\|0\|0, A'[i]), \forall i > 1$, and $Y'[i] = \tilde{E}_{U'}((i-1)\|1\|0, M'[i])\ \forall i$. Let $V' = W' \oplus \Sigma_{i>1} X'[i] + \Sigma_i Y'[i]$.
5. $\mathcal{A}$ computes $A'[1] = \tilde{E}^{-1}_{U'}(0, V')$.
6. $\mathcal{A}$ outputs (K, N, A, M) and (K', N', A', M').

The same attack strategy works for MRS and MRSO as well.

A.8 BTM

Fig. 17 gives the encryption algorithm of BTM (Bivariate Tag Mixing) [IY09]. BTM uses a bi-variate hash F that uses two keys L and U and accepts a vector of inputs $(X_1, \dots, X_d)$, where X_i is of length $x_i n$ bits for each i. (We skip the definition for incomplete blocks here.) F is defined as

$$F_{L,U}(X_1, \dots, X_d) := U^{d-1} \cdot f_L(X_1) \oplus \dots \oplus U \cdot f_L(X_{d-1}) \oplus f_L(X_d),$$

where $f_L(X_i)$ denotes the polynomial defined as

$$f_L(X_i) := 2 \cdot (L^{x_i} \oplus L^{x_i - 1} \cdot X_i[1] \oplus \dots \oplus L \cdot X_i[x_i - 1] \oplus X_i[x_i]).$$

$\mathsf{MRO}[\tilde{E}].\mathsf{Enc}(K, N, A = A[1]\| \ldots \|A[a], M = M[1]\| \ldots \|M[m])$

1: $U \leftarrow K\|N$
2: $S \leftarrow 0$
3: **for** $i \in [1..a]$ **do**
4: $\quad S \leftarrow S \oplus \tilde{E}_U((i-1)\|0\|0, A[i])$
5: **end for**
6: **for** $i \in [1..m]$ **do**
7: $\quad S \leftarrow S \oplus \tilde{E}_U((i-1)\|1\|0, M[i])$
8: **end for**
9: $S \leftarrow \tilde{E}_U(0\|2\|0, S \oplus an\|mn)$
10: $T \leftarrow \lceil S \rceil_t$
11: **for** $i \in [1..m]$ **do**
12: $\quad Z[i] \leftarrow \tilde{E}_U(0\|0\|1, T\|(i-1))$
13: **end for**
14: **return** $(C := M \oplus Z, T)$

Figure 16: Encryption of MRO

$\mathsf{BTM}[E, F].\mathsf{Enc}(K, A, M = M[1]\| \ldots \|M[m])$

1: $L \leftarrow E_K(0)$
2: $U \leftarrow E_K(1)$
3: $S \leftarrow F_{L,U}(A, M)$ ▷ F is a bivariate hash
4: $T \leftarrow E_K(S)$
5: $X \leftarrow T \boxplus U$ ▷ $\boxplus$ denotes half-wise addition modulo $2^{n/2}$
6: **for** $i \in [1..m]$ **do**
7: $\quad Z[i] \leftarrow E_K(X \boxplus (i-1))$
8: **end for**
9: $Z \leftarrow Z[1]\| \ldots \|Z[m]$
10: **return** $(C := M \oplus Z, T)$

Figure 17: Encryption of BTM

Thus, for two inputs X_1, X_2 such that X_1 is just one n-bit block, we have

$$F_{L,U}(X_1, X_2) = 2 \cdot U \cdot X_1 \oplus f_L(X_2).$$

Thus, if the associated data is a single block of n bits, line 3 of the algorithm in Fig. 17 gives

$$S = 2 \cdot U \cdot A \oplus f_L(M).$$

A $\mathrm{CMT}_{\mathrm{k}}$ adversary $\mathcal{A}$ can exploit this as follows.

1. $\mathcal{A}$ first chooses (K, A, M) with a single-block A and computes (C, T) as the output of $\mathsf{BTM}[E, F].\mathsf{Enc}(K, A, M)$, as well as the subkeys $L = E_K(0)$ and $U = E_K(1)$.

2. $\mathcal{A}$ chooses $K' \neq K$ and computes $L' = E_{K'}(0)$, $U' = E_{K'}(1)$. (In the unlikely event that $L' = L$, $U' = U$, try other values of K' till $L' \neq L$ or $U' \neq U$.)

3. $\mathcal{A}$ executes lines 5-9 of the encryption algorithm in Fig. 17 starting from T and U' and obtains Z'.

4. $\mathcal{A}$ computes $M' := C \oplus Z'$ and $S' := E_{K'}^{-1}(T)$.

5. $\mathcal{A}$ computes $A' := 2^{-1} \cdot U^{-1} \cdot (S' \oplus f_{L'}(M'))$.

6. $\mathcal{A}$ outputs (K, A, M) and (K', A', M').

References

[ABL+14] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. How to securely release unverified plaintext in authenticated encryption. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 105–125. Springer, 2014. *(Cited on p. 1.)*

[ABN10] Michel Abdalla, Mihir Bellare, and Gregory Neven. Robust encryption. In Daniele Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*, pages 480–497. Springer, 2010. *(Cited on p. 6.)*

[ADG+22] Ange Albertini, Thai Duong, Shay Gueron, Stefan Kölbl, Atul Luykx, and Sophie Schmieg. How to abuse and fix authenticated encryption without key commitment. In Kevin R. B. Butler and Kurt Thomas, editors, *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pages 3291–3308. USENIX Association, 2022. *(Cited on pp. 2 and 6.)*

[BBLT18] Subhadeep Banik, Andrey Bogdanov, Atul Luykx, and Elmar Tischhauser. SUNDAE: small universal deterministic authenticated encryption for the internet of things. *IACR Trans. Symmetric Cryptol.*, 2018(3):1–35, 2018. *(Cited on p. 26.)*

[BCC+24] Ritam Bhaumik, Bishwajit Chakraborty, Wonseok Choi, Avijit Dutta, Jérôme Govinden, and Yaobin Shen. The committing security of macs with applications to generic composition. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IV*, volume 14923 of *Lecture Notes in Computer Science*, pages 425–462. Springer, 2024. *(Cited on pp. 2 and 6.)*

[BH22] Mihir Bellare and Viet Tung Hoang. Efficient schemes for committing authenticated encryption. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 845–875. Springer, 2022. *(Cited on pp. 2, 6, 9, and 18.)*

[BH24] Mihir Bellare and Viet Tung Hoang. Succinctly-committing authenticated encryption. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IV*, volume 14923 of *Lecture Notes in Computer Science*, pages 305–339. Springer, 2024. *(Cited on pp. 2, 3, and 17.)*

[BMOS17] Guy Barwell, Daniel P. Martin, Elisabeth Oswald, and Martijn Stam. Authenticated encryption in the face of protocol and side channel leakage. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 693–723. Springer, 2017. *(Cited on p. 1.)*

[BNT19] Mihir Bellare, Ruth Ng, and Björn Tackmann. Nonces are noticed: AEAD revisited. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 235–265. Springer, 2019. *(Cited on p. 2.)*

[BPPS17] Francesco Berti, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. On leakage-resilient authenticated encryption with decryption leakages. *IACR Trans. Symmetric Cryptol.*, 2017(3):271–293, 2017. *(Cited on p. 1.)*

[BR00] Mihir Bellare and Phillip Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 317–330. Springer, 2000. *(Cited on p. 6.)*

[CAE19] CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness, 2013 - 2019. https://competitions.cr.yp.to/caesar-submissions.html. *(Cited on pp. 2 and 18.)*

[CDD+19] Donghoon Chang, Nilanjan Datta, Avijit Dutta, Bart Mennink, Mridul Nandi, Somitra Sanadhya, and Ferdinand Sibleyras. Release of unverified plaintext: Tight unified model and application to ANYDAE. *IACR Trans. Symmetric Cryptol.*, 2019(4):119–146, 2019. *(Cited on p. 26.)*

[CFI+23] Yu Long Chen, Antonio Flórez-Gutiérrez, Akiko Inoue, Ryoma Ito, Tetsu Iwata, Kazuhiko Minematsu, Nicky Mouha, Yusuke Naito, Ferdinand Sibleyras, and Yosuke Todo. Key committing security of AEZ and more. *IACR Trans. Symmetric Cryptol.*, 2023(4):452–488, 2023. *(Cited on p. 6.)*

[CLLL21] Wonseok Choi, ByeongHak Lee, Jooyoung Lee, and Yeongmin Lee. Toward a fully secure authenticated encryption scheme from a pseudorandom permutation. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part III*, volume 13092 of *Lecture Notes in Computer Science*, pages 407–434. Springer, 2021. *(Cited on p. 24.)*

[CLS18] Debrup Chakraborty, Cuauhtemoc Mancillas López, and Palash Sarkar. Disk encryption: do we need to preserve length? *J. Cryptogr. Eng.*, 8(1):49–69, 2018. *(Cited on p. 25.)*

[CR19] John Chan and Phillip Rogaway. Anonymous AE. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the*

Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part II, volume 11922 of *Lecture Notes in Computer Science*, pages 183–208. Springer, 2019. *(Cited on p. 2.)*

[CR22] John Chan and Phillip Rogaway. On committing authenticated-encryption. In Vijayalakshmi Atluri, Roberto Di Pietro, Christian Damsgaard Jensen, and Weizhi Meng, editors, *Computer Security - ESORICS 2022 - 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26-30, 2022, Proceedings, Part II*, volume 13555 of *Lecture Notes in Computer Science*, pages 275–294. Springer, 2022. *(Cited on pp. 3 and 6.)*

[DEJ+24] Chandranan Dhar, Jordan Ethan, Ravindra Jejurikar, Mustafa Khairallah, Eik List, and Sougata Mandal. Context-committing security of leveled leakage-resilient AEAD. *IACR Trans. Symmetric Cryptol.*, 2024(2):348–370, 2024. *(Cited on p. 6.)*

[DFG23] Jean Paul Degabriele, Marc Fischlin, and Jérôme Govinden. The indifferentiability of the duplex and its practical applications. In Jian Guo and Ron 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. *(Cited on p. 6.)*

[DFI+24] Patrick Derbez, Pierre-Alain Fouque, Takanori Isobe, Mostafizar Rahman, and André Schrottenloher. Key committing attacks against aes-based AEAD schemes. *IACR Trans. Symmetric Cryptol.*, 2024(1):135–157, 2024. *(Cited on p. 6.)*

[DGRW18] Yevgeniy Dodis, Paul Grubbs, Thomas Ristenpart, and Joanne Woodage. Fast message franking: From invisible salamanders to encryptment. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 155–186. Springer, 2018. *(Cited on pp. 2 and 6.)*

[DMA23] Joan Daemen, Silvia Mella, and Gilles Van Assche. Committing authenticated encryption based on SHAKE. *IACR Cryptol. ePrint Arch.*, page 1494, 2023. *(Cited on p. 6.)*

[FLLW16] Christian Forler, Eik List, Stefan Lucks, and Jakob Wenzel. Efficient beyond-birthday-bound-secure deterministic authenticated encryption with minimal stretch. In Joseph K. Liu and Ron Steinfeld, editors, *Information Security and Privacy - 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings, Part II*, volume 9723 of *Lecture Notes in Computer Science*, pages 317–332. Springer, 2016. *(Cited on p. 25.)*

[FLPQ13] Pooya Farshim, Benoît Libert, Kenneth G. Paterson, and Elizabeth A. Quaglia. Robust encryption, revisited. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings*, volume 7778 of *Lecture Notes in Computer Science*, pages 352–368. Springer, 2013. *(Cited on p. 6.)*

[FOR17] Pooya Farshim, Claudio Orlandi, and Razvan Rosie. Security of symmetric primitives under incorrect usage of keys. *IACR Trans. Symmetric Cryptol.*, 2017(1):449–473, 2017. *(Cited on p. 6.)*

[GH03] Yitchak Gertner and Amir Herzberg. Committing encryption and publicly-verifiable signcryption. *IACR Cryptol. ePrint Arch.*, page 254, 2003. *(Cited on p. 6.)*

[GJMN16] Robert Granger, Philipp Jovanovic, Bart Mennink, and Samuel Neves. Improved masking for tweakable blockciphers with applications to authenticated encryption. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 263–293. Springer, 2016. *(Cited on p. 27.)*

[GL15] Shay Gueron and Yehuda Lindell. GCM-SIV: full nonce misuse-resistant authenticated encryption at under one cycle per byte. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 109–119. ACM, 2015. *(Cited on p. 18.)*

[GLR17] Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. Message franking via committing authenticated encryption. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 66–97. Springer, 2017. *(Cited on p. 6.)*

[Hir20] Shoichi Hirose. Compactly committing authenticated encryption using tweakable block cipher. In Miroslaw Kutylowski, Jun Zhang, and Chao Chen, editors, *Network and System Security - 14th International Conference, NSS 2020, Melbourne, VIC, Australia, November 25-27, 2020, Proceedings*, volume 12570 of *Lecture Notes in Computer Science*, pages 187–206. Springer, 2020. *(Cited on p. 6.)*

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 15–44. Springer, 2015. *(Cited on p. 1.)*

[HM22] Shoichi Hirose and Kazuhiko Minematsu. Compactly committing authenticated encryption using encryptment and tweakable block cipher. *IACR Cryptol. ePrint Arch.*, page 1670, 2022. *(Cited on p. 6.)*

[HM23] Shoichi Hirose and Kazuhiko Minematsu. A formal treatment of envelope encryption. *IACR Cryptol. ePrint Arch.*, page 1727, 2023. *(Cited on p. 6.)*

[IM16] Tetsu Iwata and Kazuhiko Minematsu. Stronger security variants of GCM-SIV. *IACR Trans. Symmetric Cryptol.*, 2016(1):134–157, 2016. *(Cited on p. 24.)*

[IY09] Tetsu Iwata and Kan Yasuda. BTM: A single-key, inverse-cipher-free mode for deterministic authenticated encryption. In Michael J. Jacobson Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *Selected Areas in Cryptography, 16th Annual International Workshop, SAC 2009, Calgary, Alberta, Canada, August 13-14, 2009, Revised Selected Papers*, volume 5867 of *Lecture Notes in Computer Science*, pages 313–330. Springer, 2009. *(Cited on p. 28.)*

[JNP19] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Joltik v1.3. *Submission to CAESAR Competition*, 2013 - 2019. https://competitions.cr.yp.to/round2/joltikv13.pdf. *(Cited on p. 18.)*

[JNPS19] Jérémy Jean, Ivica Nikolic, Thomas Peyrin, and Yannick Seurin. Deoxys v1.41. *Submission to CAESAR Competition*, 2013 - 2019. https://competitions.cr.yp.to/round3/deoxysv141.pdf. *(Cited on pp. 11 and 18.)*

[JNPS21] Jérémy Jean, Ivica Nikolic, Thomas Peyrin, and Yannick Seurin. The deoxys AEAD family. *J. Cryptol.*, 34(3):31, 2021. *(Cited on pp. 11 and 18.)*

[KSW23] Juliane Krämer, Patrick Struck, and Maximiliane Weishäupl. Committing authenticated encryption: Sponges vs. block-ciphers in the case of the NIST LWC finalists. *IACR Cryptol. ePrint Arch.*, page 1525, 2023. *(Cited on p. 6.)*

[LGR21] Julia Len, Paul Grubbs, and Thomas Ristenpart. Partitioning oracle attacks. In Michael D. Bailey and Rachel Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 195–212. USENIX Association, 2021. *(Cited on pp. 2 and 6.)*

[MLGR23] Sanketh Menda, Julia Len, Paul Grubbs, and Thomas Ristenpart. Context discovery and commitment attacks - how to break ccm, eax, siv, and more. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 379–407. Springer, 2023. *(Cited on pp. 2 and 6.)*

[NIS18] NIST. Submission requirements and evaluation criteria for the Lightweight Cryptography Standardisation Process, 2018. https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf. *(Cited on pp. 2 and 6.)*

[NSS20] Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. LM-DAE: low-memory deterministic authenticated encryption for 128-bit security. *IACR Trans. Symmetric Cryptol.*, 2020(4):1–38, 2020. *(Cited on p. 27.)*

[NSS22] Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Secret can be public: Low-memory AEAD mode for high-order masking. Cryptology ePrint Archive, Paper 2022/812, 2022. *(Cited on p. 6.)*

[NSS24] Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. KIVR: committing authenticated encryption using redundancy and application to gcm, ccm, and more. In Christina Pöpper and Lejla Batina, editors, *Applied Cryptography and Network Security - 22nd International Conference, ACNS 2024, Abu Dhabi, United Arab Emirates, March 5-8, 2024, Proceedings, Part I*, volume 14583 of *Lecture Notes in Computer Science*, pages 318–347. Springer, 2024. *(Cited on p. 3.)*

[PS16] Thomas Peyrin and Yannick Seurin. Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 33–63. Springer, 2016. *(Cited on p. 18.)*

[PSV15] Olivier Pereira, François-Xavier Standaert, and Srinivas Vivek. Leakage-resilient authentication and encryption from symmetric cryptographic primitives. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 96–108. ACM, 2015. *(Cited on p. 1.)*

[Rog04] Phillip Rogaway. Nonce-based symmetric encryption. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 2004. *(Cited on p. 1.)*

[RS06] Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 373–390. Springer, 2006. *(Cited on p. 1.)*

[SW24] Patrick Struck and Maximiliane Weishäupl. Constructing committing and leakage-resilient authenticated encryption. *IACR Trans. Symmetric Cryptol.*, 2024(1):497–528, 2024. *(Cited on p. 6.)*