# Context Discovery and Commitment Attacks
## How to Break CCM, EAX, SIV, and More

Sanketh Menda[1], Julia Len[1], Paul Grubbs[2], and Thomas Ristenpart[1]

[1] Cornell Tech, New York, USA
[2] University of Michigan, Ann Arbor, USA

**Abstract.** A line of recent work has highlighted the importance of context commitment security, which asks that authenticated encryption with associated data (AEAD) schemes will not decrypt the same adversarially-chosen ciphertext under two different, adversarially-chosen contexts (secret key, associated data, and nonce). Despite a spate of recent attacks, many open questions remain around context commitment; most obviously nothing is known about the commitment security of important schemes such as CCM, EAX, and SIV.

We resolve these open questions, and more. Our approach is to, first, introduce a new framework that helps us more granularly define context commitment security in terms of what portions of a context are adversarially controlled. We go on to formulate a new security notion, called context discoverability, which can be viewed as analogous to preimage resistance from the hashing literature. We show that unrestricted context commitment security (the adversary controls all of the two contexts) implies context discoverability security for a class of schemes encompassing most schemes used in practice. Then, we show new context discovery attacks against a wide set of AEAD schemes, including CCM, EAX, SIV, GCM, and OCB3, and, by our general result, this gives new unrestricted context commitment attacks against them.

Finally, we explore the case of restricted context commitment security for the original SIV mode, for which no prior attack techniques work (including our context discovery based ones). We are nevertheless able to give a novel $\mathcal{O}(2^{n/3})$ attack using Wagner's k-tree algorithm for the generalized birthday problem.

**Keywords:** Secret-key cryptography · AEAD · Committing encryption

## 1 Introduction

Designers of authenticated encryption with associated data (AEAD) have traditionally targeted security in the sense of confidentiality and ciphertext integrity, first in the context of randomized authenticated encryption [6], and then nonce-based [32] and misuse-resistant AEAD [33].

But in recent years researchers and practitioners have begun realizing that confidentiality and integrity as previously formalized prove insufficient in a variety of contexts. In particular, the community is beginning to appreciate the danger of schemes that are not key committing, meaning that an attacker can compute a ciphertext such that it can successfully decrypt under two (or more) keys. Non-key-committing AEAD was first shown to be a problem in the context of moderation in encrypted messaging [16,24], and later in password-based encryption [29], password-based key exchange [29], key rotation schemes [2], and symmetric hybrid (or envelope) encryption [2].

Even more recently, new definitions have been proposed [5] that target committing to the key, associated data, and nonce. And while there have been proposals for new schemes [2,5] that meet these varying definitions, questions still remain about which current AEAD schemes are committing and in which ways. Moreover, there have been no commitment results shown for a number of important practical AEAD schemes, such as CCM [17], EAX [11], and SIV [33]. Implementing (and standardizing) new AEAD schemes takes time and so understanding which standard AEAD schemes can be securely used in which settings is a pressing issue.

This work makes four main contributions. First, we provide a new, more granular framework for commitment security, which expands on prior ones to better capture practical attack settings. Second, we show the first key commitment attack against the original SIV mode, which was previously an open question. Third, we introduce a new kind of commitment security notion for AEAD—what we call *context discoverability*—which is analogous to preimage resistance for cryptographic hash functions. Fourth, we give context discovery attacks against a range of schemes which, by a general implication, also yield new commitment attacks against those schemes. A summary of our new attacks, including comparison with prior ones, when relevant, is given in Figure 1.

**Granular commitment notions.** Recall that a nonce-based AEAD encryption algorithm $\mathsf{Enc}$ takes as input a key $K$, nonce $N$, associated data $A$, and a message $M$. It outputs a ciphertext $C$. Decryption $\mathsf{Dec}$ likewise takes in a $(K, N, A)$ triple, which we call the decryption *context*, along with a ciphertext $C$, and outputs either a message $M$ or special error symbol $\perp$.

While most prior work has focused on key commitment security, which requires commitment to only one part (the key) of the decryption context, Bellare and Hoang (BH) [5] suggest a more expansive sequence of commitment notions for nonce-based AEAD. For the first, CMT-1, an adversary wins if it efficiently computes a ciphertext $C$ and two decryption contexts $(K_1, N_1, A_1)$ and $(K_2, N_2, A_2)$ such that decryption of $C$ under either context works (does not output $\perp$) and $K_1 \neq K_2$. CMT-1 is often called key commitment.[3] CMT-3 relaxes the latter winning condition to allow a win should the decryption contexts differ in any way. We therefore refer to CMT-3 as *context commitment* and schemes that meet CMT-3 as *context committing*. These notions form a strict hierarchy, with CMT-3 being the strongest. Despite this, most prior attacks [16,24,29,2] have focused solely on key commitment (CMT-1).

[3] BH refer to this as CMTD-1, but for tidy AEAD schemes, CMT-1 and CMTD-1 are equivalent, so we prefer the compact term.

Our first contribution is to refine further the definitional landscape for nonce-based AEAD schemes in a way that is particularly useful for exploring context commitment attacks. In practice, attackers will often face application-specific restrictions preventing full control over the decryption context. For example, in the Dodis, Grubbs, Ristenpart, and Woodage (DGRW) [16] attacks against Facebook's message franking scheme, the adversary had to build a ciphertext that decrypts under two contexts with equivalent nonces. Their (in BH's terminology) CMT-1 attack takes on a special form, and we would like to be able to formally distinguish between attacks that achieve additional adversarial goals (e.g., different keys but equivalent nonces) and those that may not.

We therefore introduce a new, parameterized security notion that generalizes the BH notions. Our CMT$[\Sigma]$ notion specifies what we call a setting $\Sigma = (\mathsf{ts}, \mathsf{S}, \mathsf{P})$ that includes a target specifier $\mathsf{ts}$, a context selector $\mathsf{S}$, and a predicate $\mathsf{P}$. The parameters $\mathsf{ts}$ and $\mathsf{S}$ specify which parts of the context are attacker-controlled versus chosen by the game, and which of the latter are revealed to the attacker. Furthermore, the predicate $\mathsf{P}$ takes as input the two decryption contexts and decrypted messages, and outputs whether the pair of tuples satisfy a winning condition. An adversary wins if it outputs a ciphertext and two contexts satisfying the condition that each decrypt the ciphertext without error. The resulting family of commitment notions includes both CMT-1 and CMT-3 but also covers a landscape of further notions.

We highlight two sets of notions. The first set is composed of $\mathrm{CMT}_{\mathsf{k}}, \mathrm{CMT}_{\mathsf{n}}$, and $\mathrm{CMT}_{\mathsf{a}}$, which use predicates $(K_1 \neq K_2)$, $(N_1 \neq N_2)$, and $(A_1 \neq A_2)$, respectively. The first notion is equivalent to CMT-1; the latter two are new. All of them are orthogonal to each other and a scheme that meets all three simultaneously achieves CMT-3. We say these notions are *permissive* because the predicates used do not make any demands on other components of the context. In contrast, *restrictive* variants, which we denote via $\mathrm{CMT}^*_{\mathsf{k}}, \mathrm{CMT}^*_{\mathsf{n}}, \mathrm{CMT}^*_{\mathsf{a}}$, require equality for other context components. For example the first uses predicate $(K_1 \neq K_2) \wedge ((N_1, A_1) = (N_2, A_2))$. These capture the types of restrictions faced in real attacks mentioned above.

**Breaking the original SIV.** While prior work has shown (in our terminology) $\mathrm{CMT}^*_{\mathsf{k}}$ attacks for GCM [24,16], GCM-SIV [35,29], ChaCha20/Poly1305 [24,29], XChaCha20/Poly1305 [29], and OCB3 [2], an open question of practical interest [36] is whether there also exists a $\mathrm{CMT}^*_{\mathsf{k}}$ attack against Synthetic IV (SIV) mode [33]. We resolve this open question, showing an attack that works in time about $2^{53}$. It requires new techniques compared to prior attacks.

SIV combines a PRF $F$ with CTR mode encryption, encrypting by first computing a tag $T = F_K(N, A, M)$ and then applying CTR mode encryption to $M$, using $T$ as the (synthetic) IV and a second key $K'$. The tag and CTR mode output are, together, the ciphertext. Decryption recovers the message and then recomputes the tag, rejecting the ciphertext if it does not match. Schmieg [35] and Len, Grubbs, and Ristenpart (LGR) [29] showed that when $F$ is a universal hash-based PRF, in particular GHASH for AES-GCM-SIV, one can achieve a fast $\mathrm{CMT}^*_{\mathsf{k}}$ attack.

| **Scheme** | **CDY$^*_\mathsf{a}$** | **CDY$^*_\mathsf{n}$** | **CMT$^*_\mathsf{a}$** | **CMT$^*_\mathsf{k}$** | **CMT$_\mathsf{k}$** | **CMT-3** |
|---|---|---|---|---|---|---|
| GCM [19] | ★✗ §4 | ★✗ §4 | ★✗ §E | ☆✗ [24,16] | ☆✗ [24,16] | ☆✗ [24,16] |
| SIV [34] | ★✗ §4 | | | ★✗ §5 | ★✗ ⏩ | ★✗ ⏩ |
| CCM [17] | ★✗ §4 | | | | ★✗ ⏩ | ★✗ ⏩ |
| EAX [11] | ★✗ §4 | ★✗ §4 | | | ★✗ ⏩ | ★✗ ⏩ |
| OCB3 [28] | ★✗ §4 | | | ☆✗ [2] | ☆✗ [2] | ☆✗ [2] |
| PaddingZeros | ★✓ ⏩ | ★✓ ⏩ | ★✗ §E | ☆✓ [2] | ☆✓ [5] | ★✗ ⏩ |
| KeyHashing | ★✓ ⏩ | ★✓ ⏩ | ★✗ §E | ☆✓ [2] | ☆✓ [2] | ★✗ ⏩ |
| CAU-C1 [5] | ★✓ ⏩ | ★✓ ⏩ | | ☆✓ [5] | ☆✓ [5] | ★✗ ⏩ |

Fig. 1: Summary of context discovery and commitment attacks against a variety of popular AEAD schemes. Symbol ✓ indicates a proof that any attack will take at least $2^{64}$ time, while symbol ✗ indicates the existence of an attack that takes less than $2^{64}$ time; symbol ★ indicates results new to this paper and ☆ indicates prior work (citation given). CMT$_\mathsf{k}$ and CMT-3 are from Bellare and Hoang [5], where CMT$_\mathsf{k}$ was called CMT-1. The notions CDY$^*_\mathsf{a}$, CDY$^*_\mathsf{n}$, CMT$^*_\mathsf{a}$ and CMT$^*_\mathsf{k}$ are introduced in this paper, and are implied by CMT$_\mathsf{k}$. Symbol ⏩ indicates that the result is implied from one of the other columns by a reduction shown in this paper. §E indicates Appendix E in the full version.

Their attack does not extend to other versions of SIV, perhaps most notably the original version that uses for $F$ the S2V[$\mathsf{CMAC}$] PRF [33]. This version has been standardized [25] and is available in popular libraries like Tink [3]. For brevity here we describe the simpler case where $F$ is just $\mathsf{CMAC}$; the body will expand on the details. At first it might seem that $\mathsf{CMAC}$'s well-known lack of collision resistance (for adversarially-chosen keys), should extend to allow a simple CMT$^*_\mathsf{k}$ attack: find $K_1, K_2$ such that $T = \mathsf{CMAC}_{K_1}(N, A, M) = \mathsf{CMAC}_{K_2}(N, A, M')$ for $M \neq M'$. But the problem is that we need $M, M'$ to also satisfy that

$$M \oplus \mathsf{CTR}_{K_1'}(T) = M' \oplus \mathsf{CTR}_{K_2'}(T) \qquad (1)$$

where $\mathsf{CTR}_K(T)$ denotes running counter mode with initialization vector $T$ and block cipher key $K$. When using a GHASH-based PRF, the second equality condition "plays well" with the algebraic structure of the first condition, making it computationally easy to satisfy both simultaneously. But, here that does not work.

The core enabler for our attack is that we can recast the primary collision finding goal as a generalized birthday bound attack. For block-aligned messages, we show how the two constraints above can be rewritten as a single equation that is the xor-sum of four terms, each taking values over $\{0,1\}^n$. Were the terms independently and uniformly random, one would immediately have an instance of a 4-sum problem, which can be solved using Wagner's k-tree algorithm [38] in time $\mathcal{O}(2^{n/3})$. But our terms are neither independent nor uniformly random. Nevertheless, our main technical lemma shows that, in the ideal cipher model, the underlying block cipher and the structure of the terms (which are dictated by the details of CMAC-SIV) allows us to analyze the distribution of these terms and show that we can still apply the k-tree algorithm and achieve the same running time. This technique of applying the k-tree algorithm to biased values may be of independent interest.

Putting it all together we achieve a $\mathrm{CMT}_{\mathsf{k}}^{*}$ attack against S2V[CMAC]-SIV that works in time about $2^{53}$, making it practical and sufficiently damaging to rule out SIV as suitable for contexts where context commitment matters.

**Context discoverability.** Next we introduce a new type of security notion for AEAD. The cryptographic hashing community has long realized the significance of definitions for both collision resistance and preimage resistance [13], the latter of which, roughly speaking, refers to the ability of an attacker to find some input that maps to a target output. In analyzing $\mathrm{CMT}_{\mathsf{k}}$ security for schemes, we realized that in many cases we can give very strong attacks that, given any ciphertext, can find a context that decrypts it—a sort of preimage attack against AEAD. To avoid confusion, we refer to this new security goal for AEAD as *context discoverability (CDY)*, as the adversary is tasked with efficiently computing ("discovering") a suitable context for some target ciphertext.

While we have not seen real attacks that exploit context discoverability, since CDY is to CMT what preimage resistance is to collision resistance, we believe that they are inevitable. We therefore view it beneficial to get ahead of the curve and analyze the CDY security before concrete attacks surface.

We formalize a family of CDY definitions similarly to our treatment for CMT. Our $\mathrm{CDY}[\Sigma]$ notion is parameterized by a setting $\Sigma = (\mathsf{ts}, \mathsf{S})$ that specifies a target specifier $\mathsf{ts}$ and a context selector $\mathsf{S}$. Like for $\mathrm{CMT}[\Sigma]$, $\mathsf{ts}$ and $\mathsf{S}$ specify the parts of the context that the attacker can choose and which parts are chosen by the game and either hidden or revealed to the attacker. Unlike CMT, however, the attacker is always given a target ciphertext and needs to only produce one valid decrypting context.

Similar to $\mathrm{CMT}_{\mathsf{k}}^{*}, \mathrm{CMT}_{\mathsf{n}}^{*}, \mathrm{CMT}_{\mathsf{a}}^{*}$, we define the notions $\mathrm{CDY}_{\mathsf{k}}^{*}, \mathrm{CDY}_{\mathsf{n}}^{*}, \mathrm{CDY}_{\mathsf{a}}^{*}$. The notion $\mathrm{CDY}_{\mathsf{k}}^{*}$ captures the setting where an adversary is given arbitrary ciphertext $C$, nonce $N$, and associated data $A$, and must produce a key $K$ such that $C$ decrypts under $(K, N, A)$. Similarly, $\mathrm{CDY}_{\mathsf{n}}^{*}$ and $\mathrm{CDY}_{\mathsf{a}}^{*}$ require the adversary to provide a nonce and associated data, respectively, given the other components chosen arbitrarily. These model restricted attack settings where parts of the context are not in the adversary's control.

We also define $\mathrm{CDY}^{*}[\mathsf{ts}]$ which generalizes this intuition to any target specifier $\mathsf{ts}$. For example, in $\mathrm{CDY}^{*}[\mathsf{ts} = \{\mathsf{n}\}]$ the adversary is given arbitrary ciphertext and nonce $N$, and must produce a key $K$ and associated data $A$ such that the ciphertext decrypts under $(K, N, A)$.

We next analyze the relations between these sets of notions. In particular, we show that if an AEAD scheme is "context compressing"—ciphertexts are decryptable under more than one context—then CMT-3 security implies $\mathrm{CDY}^{*}$. This is analogous to collision resistance implying preimage resistance, though the details are different. Further, we observe that almost all deployed AEAD schemes are context compressing since they "compress" the nonce and associated data into a shorter tag. This allows us to focus on finding $\mathrm{CDY}^{*}[\Sigma]$ attacks for AEAD

$$\begin{tikzcd}
\mathrm{CMT}_{\mathsf{k}} \arrow[dr] \arrow[drr] & \mathrm{CMT}_{\mathsf{n}} \arrow[dl] \arrow[dr] & \mathrm{CMT}_{\mathsf{a}} \arrow[dll] \arrow[dl] \\
\mathrm{CDY}^*_{\mathsf{k}} & \mathrm{CDY}^*_{\mathsf{n}} & \mathrm{CDY}^*_{\mathsf{a}}
\end{tikzcd}$$

$$\begin{tikzcd}
 & & \mathrm{CMT\text{-}3} \arrow[dll] \arrow[dl] \arrow[d] \arrow[drr, dotted, "Th\ 1"] & & & \\
\mathrm{CMT}_{\mathsf{k}} \arrow[d] & \mathrm{CMT}_{\mathsf{n}} \arrow[d] & \mathrm{CMT}_{\mathsf{a}} \arrow[d] & & \mathrm{CDY}^* \arrow[dl] \arrow[d] \arrow[dr] & \\
\mathrm{CMT}^*_{\mathsf{k}} & \mathrm{CMT}^*_{\mathsf{n}} & \mathrm{CMT}^*_{\mathsf{a}} & \mathrm{CDY}^*_{\mathsf{k}} & \mathrm{CDY}^*_{\mathsf{n}} & \mathrm{CDY}^*_{\mathsf{a}}
\end{tikzcd}$$

Fig. 2: **(Top)** Selected relationships between permissive CMT notions and restrictive CDY notions. Solid arrows represent implications. **(Bottom)** Selected relationships between CMT-3 and the notions we introduce in this paper. Solid arrows represent implications. The dotted arrow from CMT-3 to CDY$^*$ holds assuming "context compression" as defined in Theorem 1.

schemes to show that these schemes also do not meet CMT[$\Sigma$] security. Selected relationships are shown in Figure 2.

This opens up a new landscape of analysis, which we explore. We characterize a large class of AEAD schemes that use non-preimage resistant MACs and, based on this weakness, we can develop fast $\mathrm{CDY}^*_{\mathsf{a}}$ attacks. The set includes CCM, EAX, SIV, GCM, and OCB3. For EAX and CCM, this represents the first attacks of any kind for committing security. For EAX and GCM, we are also able to give $\mathrm{CDY}^*_{\mathsf{n}}$ attacks, which is perhaps even more surprising a priori, given that an adversary in this case only controls the nonce.

All this sheds light on the deficiencies of several popular design paradigms for AEAD, when viewed from the perspective of context commitment security. These definitions also allow us to precisely communicate attacks and threat models. For example, CDY might suffice for some applications while others might want the more computationally expensive CMT security.

**Revisiting commitment-enhancing mechanisms.** Finally, in Appendix E in the full version we use this new framework to analyze proposed mechanisms for commitment security. First, we look at the folklore padding zeros transform which prefixes zeroes to a message before encrypting and verifies the existence of these zeroes at decryption. This transform was recommended in an early OPAQUE draft specification [27, §3.1.1] and was shown by Albertini et al. [2, §5.3] to achieve FROB security and by Bellare and Hoang [5] to achieve CMT-1 security. We show that this transform does not achieve our $\mathrm{CMT}^*_{\mathsf{a}}$ notion (and thus CMT-3) for all AEAD schemes, ruling it out as a candidate commitment security transform. We then make similar observations about the $\mathsf{CommitKey}$

transform which appends to the ciphertext a hash commitment to the key and the nonce. Finally, we conclude by considering the practical key commitment security of the recent CAU-C1 scheme from BH [5]. While a naive adaptation of DGRW's [16] "invisible salamanders" attack to this scheme takes about $2^{81}$ time, we show a more optimized attack which takes a little more than $2^{64}$ time, showing that 64-bit key-committing security does not preclude practical attacks.

**Next steps and open problems.** Our results resolve a number of open problems about AEAD commitment security, and overall highlight the value of new definitional frameworks that surface different avenues for attack. That said, we leave several open problems, such as whether different flavors of context discovery or commitment attacks can be found against popular schemes—the blank entries in Figure 1. Our attack techniques do not seem to work against these schemes, but whether positive security results can be shown is unclear.

## 2 Background

**Notation.** We refer to elements of $\{0,1\}^*$ as *bitstrings*, denote the length of a bitstring $x$ by $|x|$ and the left-most (i.e., "most-significant") bit by $\mathsf{msb}(x)$. Given two bitstrings $x$ and $y$, we denote their concatenation by $x \,\|\, y$, their bitwise xor by $x \oplus y$, and their bitwise and by $x \& y$. Given a number $n$, we denote its $m$-bit encoding as $\mathsf{encode}_m(n)$. For a finite set $X$, we use $x \leftarrow\!\!\$\; X$ to denote sampling a uniform, random element from $X$ and assigning it to $x$.

Sometimes, we operate in the finite field $\mathrm{GF}(2^n)$ with $2^n$ elements. This field is defined using an irreducible polynomial $f(\alpha)$ in $\mathrm{GF}(2)[\alpha]$ of degree $n$. The elements of the field are polynomials $x_0 + x_1\alpha + x_2\alpha^2 + \cdots + x_{n-1}\alpha^{n-1}$ of degree $n-1$ with binary coefficients $x_i \in \mathrm{GF}(2)$. These polynomials can be represented by the $n$-bit string $x_0x_1\cdots x_{n-1}$ of their coefficients. Both addition and subtraction of two $n$-bit strings, denoted $x+y$ and $x-y$, respectively, is their bitwise xor $x \oplus y$. Multiplication of two $n$-bit strings, denoted $x \cdot y$, corresponds to the multiplication of the corresponding polynomials $x$ and $y$ followed by modular reduction with the irreducible polynomial $f(\alpha)$.

**Probability.** An *$n$-bit random variable* $X$ is one whose value is probabilistically assigned, defined by *probability mass function* $p_X(x) \coloneqq \Pr[X = x]$. The *$n$-bit uniform random variable* $U$ is the random variable with the probability mass function $p_U(x) = \frac{1}{2^n}$ for all $x \in \{0,1\}^n$. Given two $n$-bit random variables $X$ and $Y$, we define the *total variation distance* between them as

$$\Delta(X,Y) \coloneqq \max_{i \in \{0,1\}^n} \big| \Pr(X = i) - \Pr(Y = i) \big| \,.$$

A *random function* $F$ from $n$-bit strings to $m$-bit strings is a collection $\{X_i : i \in \{0,1\}^n\}$ of $m$-bit random variables $X_i$, one for each $n$-bit input, such that for all $i \in \{0,1\}^n$, $F(i) \coloneqq X_i$. A random function $F$ from $n$-bit strings to $m$-bit strings is *uniformly random* if, for all $i \in \{0,1\}^n$, $F(i)$ is the $m$-bit uniform random variable. We say that two random functions $F_1$ and $F_2$ from $n$-bit strings to

$m$-bit strings are *independent* if, for all $i \in \{0,1\}^n$ and for all $j \in \{0,1\}^n$, $F_1(i)$ and $F_2(j)$ are independent $m$-bit random variables.

**Code-based games.** To formalize security experiments, we use the *code-based games* framework of Bellare and Rogaway [10]; with refinements from Ristenpart, Shacham, and Shrimpton [31]. A *procedure* $P$ is a sequence of code-like statements that accepts some input and produces some output. We use superscripts like $P^Q$ to denote that procedure $P$ calls procedure $Q$. We use $(G \Rightarrow x)$ to denote the event that the procedure $G$ outputs $x$, over the random coins of the procedure. Finally, given a game $G$ and an adversary $\mathcal{A}$, we denote the *advantage* of $\mathcal{A}$ at $G$ by $\mathbf{Adv}_G(\mathcal{A}) \coloneqq \Pr[G(\mathcal{A}) \Rightarrow \mathsf{true}]$.

**Cost of attacks.** We represent cryptanalytic attacks by procedures and compute their cost using a *unit-cost RAM model*. Specifically, following [31], we use the convention that each pseudocode statement of a procedure runs in unit time. This lets us write the running time of a procedure as the maximum number of statements executed, with the maximum taken over all inputs of a given size. Similarly, we define the number of queries as the maximum number of queries executed over inputs of a given size. We recognize that this is a simplification of the real-world (e.g., see Wiener [39]), but for the attacks discussed in this paper, we nevertheless believe that it provides a good estimate.

**Pseudorandom functions.** A *pseudorandom function (PRF)* is a function $\mathsf{F} : \mathcal{K} \times \mathcal{M} \to \mathcal{Y}$ defined over a key space $\mathcal{K} \subseteq \{0,1\}^*$, message space $\mathcal{M} \subseteq \{0,1\}^*$, and output space $\mathcal{Y} \subseteq \{0,1\}^*$, that is indistinguishable from a uniform random function. More formally, we define the PRF advantage of an adversary $\mathcal{A}$ as

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{F}}(\mathcal{A}) \coloneqq \left|\Pr[K \leftarrow_{\$} \mathcal{K} : \mathcal{A}(\mathsf{F}(K,\cdot))] - \Pr[R \leftarrow_{\$} \mathrm{Func} : \mathcal{A}(R)]\right|,$$

and say that $\mathsf{F}$ is a PRF if this advantage is small for all adversaries $\mathcal{A}$ that run in a feasible amount of time.

**Hash functions.** A *hash function* is a function $\mathsf{H} : \mathcal{K} \times \mathcal{M} \to \mathcal{Y}$, defined over a key space $\mathcal{K} \subseteq \{0,1\}^*$, message space $\mathcal{M} \subseteq \{0,1\}^*$, and hash space $\mathcal{Y} \subseteq \{0,1\}^*$. We define the collision-resistance advantage of adversary $\mathcal{A}$ for $\mathsf{H}$ as

$$\begin{aligned}\mathbf{Adv}^{\mathrm{coll}}_{\mathsf{H}}(\mathcal{A}) \coloneqq \Pr\big[&K \leftarrow_{\$} \mathcal{K}, (M_1, M_2) \leftarrow_{\$} \mathcal{A}(K) : \\ &(M_1 \neq M_2) \text{ and } (\mathsf{H}(K, M_1) = \mathsf{H}(K, M_2))\big].\end{aligned}$$

**Block ciphers and the ideal cipher model.** An $n$-bit *block cipher*, or a block cipher with *block length* $n$ bits, is a function $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$, where for each key $k \in \{0,1\}^n$, $E(k,\cdot)$ is a permutation on $\{0,1\}^n$. Since it is a permutation, it has an inverse which we denote by $E^{-1}(k,\cdot)$. To simplify notation, we sometimes use the shorthands $E_k(\cdot) \coloneqq E(k,\cdot)$ and $E_k^{-1}(\cdot) \coloneqq E^{-1}(k,\cdot)$.

An $n$-bit *ideal block cipher* [26] is a random map $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$, such that for each key $k \in \{0,1\}^n$, $E_k(\cdot)$ is a permutation on $\{0,1\}^n$. Alternatively, we can think of an ideal block cipher as one where for each key

$k \in \{0,1\}^n$, $E_k(\cdot)$ is uniformly, randomly sampled from the set of permutations on $n$-bits.

**Authenticated encryption schemes.** An *AEAD scheme* is a triple of algorithms $\mathsf{AEAD} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$, defined over a key space $\mathcal{K} \subseteq \{0,1\}^*$, nonce space $\mathcal{N} \subseteq \{0,1\}^*$, associated data space $\mathcal{A} \subseteq \{0,1\}^*$, message space $\mathcal{M} \subseteq \{0,1\}^*$, and ciphertext space $\mathcal{C} \subseteq \{0,1\}^*$.

1. $\mathsf{Kg} : \varnothing \to \mathcal{K}$ is a randomized algorithm that takes no input and returns a fresh secret key $K$.
2. $\mathsf{Enc} : (\mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}) \to (\mathcal{C} \cup \{\bot\})$ is a deterministic algorithm that takes a 4-tuple of a key $K$, nonce $N$, associated data $A$, and message $M$ and returns a ciphertext $C$ or an error (denoted by $\bot$).
3. $\mathsf{Dec} : (\mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C}) \to (\mathcal{M} \cup \{\bot\})$ is a deterministic algorithm that takes a 4-tuple of a key $K$, nonce $N$, associated data $A$, and ciphertext $C$ and returns a plaintext $M$ or an error (denoted by $\bot$).

We call the non-message inputs to $\mathsf{Enc}$—the key, nonce, and associated data—the *encryption context* and the non-ciphertext inputs to $\mathsf{Dec}$—the key, nonce, and associated data—the *decryption context*. And, for a given message, say that an encryption context is *valid* if $\mathsf{Enc}$ succeeds (i.e., does not output $\bot$). Similarly, for a given ciphertext, say that a decryption context is *valid* if $\mathsf{Dec}$ succeeds (i.e., does not output $\bot$).

For traditional AEAD correctness, we need $\mathsf{Enc}$ to be the inverse of $\mathsf{Dec}$. In other words, for any 4-tuple $(K, N, A, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}$, it holds that

$$\mathsf{Dec}(K, N, A, \mathsf{Enc}(K, N, A, M)) = M\ .$$

In addition, we impose *tidyness* [30], *ciphertext validity*, and *length uniformity* assumptions. Tidyness requires that for any 4-tuple $(K, N, A, C) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C}$, it holds that

$$\mathsf{Dec}(K, N, A, C) = M \neq \bot \implies \mathsf{Enc}(K, N, A, M) = C\ .$$

Ciphertext validity requires that for every ciphertext $C \in \mathcal{C}$ there exists at least one valid decryption context $(K, N, A) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A}$; that is $\mathsf{Dec}(K, N, A, C) \neq \bot$. Length uniformity requires that the length of a ciphertext depends only on the length of the message and length of the associated data.

Finally, for AEAD security, we use the traditional *privacy* and *authenticity* definitions [32, §3].

**Committing authenticated encryption.** A number of prior notions for committing AEAD have been proposed. In Figure 3 we provide the CMT-1 and CMT-3 games from Bellare and Hoang [5]. The FROB game from Farshim, Orlandi, and Rosie [22] adapted to the AEAD setting by Grubbs, Lu, and Ristenpart [24], is the same except that the final highlighted predicate is changed to "$K_1 = K_2$ or $N_1 \neq N_2$". The FROB game asks the adversary to produce a ciphertext that decrypts under two different keys with the same nonce. The

```
CMT-1(A):
((K_1, N_1, A_1), (K_2, N_2, A_2), C) ←$ A
M_1 ← AEAD.Dec(K_1, N_1, A_1, C)
M_2 ← AEAD.Dec(K_2, N_2, A_2, C)
// decryption success
If M_1 = ⊥ or M_2 = ⊥
    Return false
// commitment condition
If K_1 = K_2
    Return false
Return true
```

```
CMT-3(A):
((K_1, N_1, A_1), (K_2, N_2, A_2), C) ←$ A
M_1 ← AEAD.Dec(K_1, N_1, A_1, C)
M_2 ← AEAD.Dec(K_2, N_2, A_2, C)
// decryption success
If M_1 = ⊥ or M_2 = ⊥
    Return false
// commitment condition
If (K_1, N_1, A_1) = (K_2, N_2, A_2)
    Return false
Return true
```

Fig. 3: **(Left)** The CMT-1 game [5]. **(Right)** The CMT-3 game [5]. The differences are highlighted.

CMT-1 game is more permissive and removes the condition that the nonce be the same. The CMT-3 game is even more permissive and relaxes the different key condition to different keys, nonces, or associated data. Bellare and Hoang [5] show that CMT-3 implies CMT-1, which implies FROB. We will expand on these definitions with a more general framework next.

## 3 Granular Committing Encryption Definitions

We provide a more general framework for defining commitment security for encryption. As motivation, we observe that while the CMT-1 and the stronger CMT-3 notions provide good security goals for constructions, they do not precisely capture the *way* in which attacks violate security—which parts of the decryption context does the attacker need to control, which parts have been pre-selected by some other party, and which parts are known to the attacker.

These considerations are crucial for determining the exploitability of commitment vulnerabilities in practice. For instance, the vulnerability in Facebook attachment franking [20] exploited by Dodis et al. [16, §3] only works if the nonces are the same; and the key rotation attack described by Albertini et al. [2] only works with keys previously imported to the key management service. And, looking ahead, we propose a variant of the Subscribe with Google attack described by Albertini et al. [2] in which a malicious publisher provides a full decryption context only knowing the honestly published ciphertext.

We provide a more general framework for commitment security notions that more precisely captures attack settings. As we will see in subsequent sections, our definitions provide a clearer explanatory framework for vulnerabilities.

**Committing security framework.** We find it useful to expand the set of security notions to more granularly capture the ways in which the two decryption contexts are selected that generalizes context commitment security. In Figure 4 we detail the CMT[$\Sigma$] game, parameterized by a *setting* $\Sigma = (\mathsf{ts}, \mathsf{S}, \mathsf{P})$ that specifies a *target specifier* $\mathsf{ts}$, a *context selector* $\mathsf{S}$, and a *predicate* $\mathsf{P}$ (to be defined next.) The adversary helps compute a ciphertext and two decryption contexts

```
CMT[ts, S, P](𝒜):
$\mathrm{cat}_c \leftarrow\!\!\$\ \mathsf{S}$
$\mathrm{cat}_a \leftarrow\!\!\$\ \mathcal{A}(\mathsf{Reveal}_{\mathsf{ts}}(\mathrm{cat}_c))$
$\mathrm{cat} \leftarrow \mathsf{Merge}_{\mathsf{ts}}(\mathrm{cat}_c, \mathrm{cat}_a)$
If $\mathrm{cat} = \bot$:
    Return false
$(C, (K_1, N_1, A_1), (K_2, N_2, A_2)) \leftarrow \mathrm{cat}$
$M_1 \leftarrow \mathsf{AEAD.Dec}(K_1, N_1, A_1, C)$
$M_2 \leftarrow \mathsf{AEAD.Dec}(K_2, N_2, A_2, C)$
If $M_1 = \bot$ or $M_2 = \bot$:
    Return false
Return $\mathsf{P}((K_1, N_1, A_1), (K_2, N_2, A_2))$
```

| **Notion** | **Predicate** P |
|---|---|
| $\mathrm{CMT}_{\mathsf{k}}$ | $(K_1 \neq K_2)$ |
| $\mathrm{CMT}_{\mathsf{n}}$ | $(N_1 \neq N_2)$ |
| $\mathrm{CMT}_{\mathsf{a}}$ | $(A_1 \neq A_2)$ |
| $\mathrm{CMT}^*_{\mathsf{k}}$ | $(K_1 \neq K_2) \wedge (N_1, A_1) = (N_2, A_2)$ |
| $\mathrm{CMT}^*_{\mathsf{n}}$ | $(N_1 \neq N_2) \wedge (K_1, A_1) = (K_2, A_2)$ |
| $\mathrm{CMT}^*_{\mathsf{a}}$ | $(A_1 \neq A_2) \wedge (K_1, N_1) = (K_2, N_2)$ |

Fig. 4: **(Left)** The $\mathrm{CMT}[\Sigma]$ commitment security game, parameterized by $\Sigma = (\mathsf{ts}, \mathsf{S}, \mathsf{P})$, a target selector $\mathsf{ts}$, context selector $\mathsf{S}$, and predicate $\mathsf{P}$. **(Right)** Predicates for the permissive notions $\mathrm{CMT}_{\mathsf{k}}$, $\mathrm{CMT}_{\mathsf{n}}$, $\mathrm{CMT}_{\mathsf{a}}$ and restrictive notions $\mathrm{CMT}^*_{\mathsf{k}}$, $\mathrm{CMT}^*_{\mathsf{n}}$, $\mathrm{CMT}^*_{\mathsf{a}}$, where $\mathsf{ts} = \varnothing$.

$(C, (K_1, N_1, A_1), (K_2, N_2, A_2))$, what we call a *commitment attack instance (cat)*. The adversary wins if $C$ decrypts under both decryption contexts, and the two decryption contexts satisfy the predicate $\mathsf{P}$. The parameterization allows attack settings in terms of which portions of the commitment attack instance are attacker controlled versus chosen in some other way, and which of the latter are revealed to the attacker.

We now provide more details. A commitment attack instance is a tuple $(C, (K_1, N_1, A_1), (K_2, N_2, A_2))$ consisting of a ciphertext $C \in \mathcal{C}$; two keys $K_1, K_2 \in \mathcal{K}$; two nonces $N_1, N_2 \in \mathcal{N}$; and two associated data $A_1, A_2 \in \mathcal{A}$. A target specifier $\mathsf{ts}$ is a subset of labels $\{\mathsf{C}, \mathsf{k}_1, \mathsf{n}_1, \mathsf{a}_1, \mathsf{k}_2, \mathsf{n}_2, \mathsf{a}_2\} \times \{\cdot, \hat{\cdot}\}$. The left set labels the components of a commitment attack instance, called component labels, and the right set denotes whether the specified component is revealed to the adversary (no hat means revealed and hat means not revealed.) For example, $\mathsf{ts} = \{\mathsf{k}_1, \hat{\mathsf{k}}_2\}$ indicates the $K_1$ and $K_2$ in the context, and that $K_1$ is revealed to the attacker.

A context selector $\mathsf{S}$ is a randomized algorithm that takes no input and produces the challenger-defined elements of a commitment attack instance, denoted $\mathrm{cat}_c$, as specified by the target specifier $\mathsf{ts}$. The reveal function $\mathsf{Reveal}_{\mathsf{ts}}$ parameterized by $\mathsf{ts}$, takes a subset of a commitment attack instance and reveals the components that $\mathsf{ts}$ tells it to reveal; i.e., the specified components with no hat. The merge function $\mathsf{Merge}_{\mathsf{ts}}(\mathrm{cat}_c, \mathrm{cat}_a)$ parameterized by the target specifier $\mathsf{ts}$, takes two subsets of commitment attack instances $\mathrm{cat}_c$ (challenger-defined elements) and $\mathrm{cat}_a$ (adversary-defined elements) and works as follows. First, it checks for every component specified by $\mathsf{ts}$ that $\mathrm{cat}_c$ has a corresponding value. Second, it checks that for every component specified by $\mathsf{ts}$, if $\mathrm{cat}_a$ has a value, that it matches the value in $\mathrm{cat}_c$. If either of these checks fail, it outputs $\bot$. Otherwise, it returns their union $\mathrm{cat}_c \cup \mathrm{cat}_a$. Finally, the predicate $\mathsf{P}$ takes two decryption contexts output by $\mathsf{Merge}_{\mathsf{ts}}(\mathrm{cat}_c, \mathrm{cat}_a)$, and outputs $\mathsf{true}$ if they satisfy some criteria (e.g., that $K_1 \neq K_2$), and $\mathsf{false}$ otherwise.

We associate to a setting $\Sigma = (\mathsf{ts}, \mathsf{S}, \mathsf{P})$, AEAD $\Pi$, and adversary $\mathcal{A}$ the CMT advantage defined as

$$\mathbf{Adv}_{\Pi}^{\mathrm{CMT}[\Sigma]}(\mathcal{A}) := \Pr\left[\, \mathrm{CMT}[\Sigma](\mathcal{A}) \Rightarrow \mathsf{true} \,\right].$$

Taking a concrete security approach, we will track the running time used by $\mathcal{A}$ and provide explicit advantage functions. Adapting our notions to support asymptotic definitions of security is straightforward: in our discussions we will often say a scheme is $\mathrm{CMT}[\Sigma]$ secure as informal shorthand that no adversary can win the $\mathrm{CMT}[\Sigma]$ game with "good" probability using "reasonable" running time.

**Capturing CMT-1, CMT-3, and more via predicates.** To understand our definitional framework further, we can start by seeing how to instantiate it to coincide with prior notions. Let $\mathsf{ts} = \varnothing$ indicate the empty target selector, meaning that $\mathcal{A}$ chooses the ciphertext and two decryption contexts fully. Then the set of $\Sigma$ settings that use the empty target selector defines a family of security goals, indexed solely by predicates, which we denote by $\mathrm{CMT}[\mathsf{P}]$. This family includes CMT-1 by setting $\mathsf{P} := (K_1 \neq K_2)$ and CMT-3 by setting $\mathsf{P} := (K_1, N_1, A_1) \neq (K_2, N_2, A_2)$. Not all instances in this family are interesting: consider, for example, when $\mathsf{P}$ always outputs $\mathsf{true}$ or $\mathsf{false}$. Nevertheless, the flexibility here allows for more granular specification of adversarial ability. For instance, the predicate that requires $(K_1 \neq K_2) \wedge (N_1 = N_2)$ captures a setting like that of the Dodis et al. [16] attack against Facebook's message franking, which requires that both decryption contexts have the same nonce.

Three games of particular interest are those with predicates that focus on inequality of the three individual context components: $(K_1 \neq K_2)$, $(N_1 \neq N_2)$, and $(A_1 \neq A_2)$. For notational brevity, we let game $\mathrm{CMT}_{\mathsf{k}} := \mathrm{CMT}[\mathsf{P} = (K_1 \neq K_2)]$ and similarly $\mathrm{CMT}_{\mathsf{n}} := \mathrm{CMT}[\mathsf{P} = (N_1 \neq N_2)]$ and $\mathrm{CMT}_{\mathsf{a}} := \mathrm{CMT}[\mathsf{P} = (A_1 \neq A_2)]$. Then $\mathrm{CMT}_{\mathsf{k}}$ corresponds to CMT-1, but $\mathrm{CMT}_{\mathsf{n}}$ and $\mathrm{CMT}_{\mathsf{a}}$ are new. They are also orthogonal to CMT-1, in the sense that we can give schemes that achieve CMT-1 but not $\mathrm{CMT}_{\mathsf{a}}$ nor $\mathrm{CMT}_{\mathsf{n}}$ security (see Theorem 9 in the full version.) All three are, however, implied by being CMT-3 secure, and a scheme that simultaneously meets $\mathrm{CMT}_{\mathsf{k}}$, $\mathrm{CMT}_{\mathsf{n}}$, and $\mathrm{CMT}_{\mathsf{a}}$ also enjoys CMT-3 security (see Lemmas 7 and 8 in the full version.)

Note that $\mathrm{CMT}_{\mathsf{k}}$, $\mathrm{CMT}_{\mathsf{n}}$, and $\mathrm{CMT}_{\mathsf{a}}$ are *permissive*: as long as the relevant component is distinct across the two contexts, it does not matter whether the other components are distinct. Also, of interest are *restrictive* versions; for example, we can consider $\mathrm{CMT}^*_{\mathsf{k}} := \mathrm{CMT}[(K_1 \neq K_2) \wedge (N_1, A_1) = (N_2, A_2)]$ which requires that the nonces and associated data are the same. Similarly, we can define restrictive notions $\mathrm{CMT}^*_{\mathsf{n}}$ and $\mathrm{CMT}^*_{\mathsf{a}}$. Restrictive versions are useful as they correspond to attacks that have limited control over the decryption context. Interestingly, these restrictive notions are not equivalent to the corresponding permissive notions, nor does a scheme that simultaneously meets $\mathrm{CMT}^*_{\mathsf{k}}$, $\mathrm{CMT}^*_{\mathsf{n}}$, and $\mathrm{CMT}^*_{\mathsf{a}}$ achieve CMT-3 security (see Theorem 10 in the full version.)

**Targeted attacks.** Returning to settings with target specifier $\mathsf{ts} \neq \varnothing$, we can further increase the family of notions considered to capture situations where a portion of the context is pre-selected. For instance, in the key rotation example of Albertini et al. [2] mentioned earlier, we would have $\mathsf{ts} = \{\mathsf{k}_1, \mathsf{k}_2\}$ and $\mathsf{S} = \{K_1 \leftarrow\!\!\$\, \mathcal{K}; K_2 \leftarrow\!\!\$\, \mathcal{K}; \text{Return } (K_1, K_2)\}$ to indicate that the malicious sender has to use the two randomly generated keys.

However, not all targeted attack settings are interesting. For some target specifiers $\mathsf{ts}$, we can specify a context selector $\mathsf{S}$ such that no adversary can achieve non-zero advantage. In particular, if we have $\mathsf{ts} = \{\mathsf{C}, \mathsf{k}_1, \mathsf{n}_1, \mathsf{a}_1\}$ and have $\mathsf{S}$ pick ciphertext $C$ and context $(K_1, N_1, A_1)$ such that $\mathsf{AEAD.Dec}(K_1, N_1, A_1, C)$ returns $\bot$, then no adversary can win the game, making the associated security notion trivial (all schemes achieve it.)

**Hiding target components.** Finally, our game considers target specifiers $\mathsf{ts}$ that indicate that some values chosen by $\mathsf{S}$ should remain hidden from $\mathcal{A}$. For example, the Subscribe with Google attack described by Albertini et al. [2] can be reframed as a meddler-in-the-middle attack as follows. A publisher creates premium content $M_1$ and encrypts it using a context $(K_1, N_1, A_1)$ to get a ciphertext $C$. The ciphertext $C$ is published, but the context $(K_1, N_1, A_1)$ is hidden. A malicious third-party, only looking at the ciphertext $C$, tries to construct a valid decryption context $(K_2, N_2, A_2)$ and uses that to sell fake paywall bypasses. We can formalize this setting by having the target specifier $\mathsf{ts} = \{\mathsf{C}, \hat{\mathsf{k}}_1, \hat{\mathsf{n}}_1, \hat{\mathsf{a}}_1\}$, with the context selector $\mathsf{S}$ as

$$\begin{array}{l} K_1 \leftarrow\!\!\$\, \mathcal{K};\ N_1 \leftarrow\!\!\$\, \mathcal{N};\ A_1 \leftarrow\!\!\$\, \mathcal{A};\ M_1 \leftarrow\!\!\$\, \mathcal{M}; \\ \text{Return } (\mathsf{AEAD.Enc}(K_1, N_1, A_1, M_1), K_1, N_1, A_1) \end{array}$$

and with $\mathsf{Reveal}_{\mathsf{ts}}(C, K_1, N_1, A_1)$ outputting $C$.

**Context discoverability security.** Dodis et al [16, §5] and Albertini et al. [2, §3.3] have pointed out that traditional CMT games are analogous to collision-resistance for hash functions, in the sense that the goal is to find two different *encryption contexts* $(K_1, N_1, A_1, M_1)$ and $(K_2, N_2, A_2, M_2)$ such that they produce the same ciphertext $C$. Under this lens, CMT with targeting (and no hiding) is like second preimage resistance, and CMT with targeting and hiding is like preimage resistance. But, the analogy to preimage resistance is not perfect, since we are not asking for *any* preimage but rather one that is not the same as the original. Further, this restriction is unnecessary. Going back to the meddler-in-the-middle example above, it suffices for an on-path attacker to produce any valid context. Thus, we find it useful to define a new preimage resistance-inspired notion of commitment security.

In Figure 5 we define the game $\mathrm{CDY}[\mathsf{ts}, \mathsf{S}]$, parameterized by a setting $\Sigma = (\mathsf{ts}, \mathsf{S})$ that specifies a target specifier $\mathsf{ts}$ and a context selector $\mathsf{S}$. In more detail, a *discoverability attack instance (dat)* is a ciphertext and a decryption context $(C, (K, N, A))$. Here, a target specifier $\mathsf{ts}$ is a subset of $\{\mathsf{k}, \mathsf{n}, \mathsf{a}\} \times \{\cdot, \hat{\cdot}\}$ and a context selector $\mathsf{S}$ is a randomized algorithm that takes no input and produces a ciphertext and the elements of a decryption context specified by the target specifier $\mathsf{ts}$. The reveal function $\mathsf{Reveal}_{\mathsf{ts}}$ and the merge function $\mathsf{Merge}_{\mathsf{ts}}(\mathrm{dat}_c, \mathrm{dat}_a)$

```
CDY[ts, S](A):
$\text{dat}_c \leftarrow\!\!\$\ \mathsf{S}$
$\text{dat}_a \leftarrow\!\!\$\ \mathcal{A}(\mathsf{Reveal}_{\mathsf{ts}}(t))$
$\text{dat} \leftarrow \mathsf{Merge}_{\mathsf{ts}}(\text{dat}_c, \text{dat}_a)$
If $\text{dat} = \perp$:
    Return false
$(C, (K, N, A)) \leftarrow \text{dat}$
$M \leftarrow \mathsf{AEAD.Dec}(K, N, A, C)$
If $M = \perp$:
    Return false
Return true
```

```
CDY[{k, n}, S](A):
$(C, K, N) \leftarrow\!\!\$\ \mathsf{S}$
$A \leftarrow\!\!\$\ \mathcal{A}(C, K, N)$
$M \leftarrow \mathsf{AEAD.Dec}(K, N, A, C)$
If $M = \perp$:
    Return false
Return true
```

```
CDY[{k, a}, S](A):
$(C, K, A) \leftarrow\!\!\$\ \mathsf{S}$
$N \leftarrow\!\!\$\ \mathcal{A}(C, K, A)$
$M \leftarrow \mathsf{AEAD.Dec}(K, N, A, C)$
If $M = \perp$:
    Return false
Return true
```

Fig. 5: **(Left)** The CDY[ts, S] commitment security game, parameterized by a target specifier ts and a context selector S. **(Middle)** The variant of CDY[$\Sigma$] used in the definition of $\text{CDY}^*_{\mathsf{a}}$. **(Right)** The variant of CDY[$\Sigma$] used in the definition of $\text{CDY}^*_{\mathsf{n}}$.

work similarly to their CMT counterparts. Finally, the goal of the adversary is to produce *one* valid decryption context for the target ciphertext.

We associate to a setting $\Sigma = (\mathsf{ts}, \mathsf{S})$, AEAD scheme $\Pi$, and adversary $\mathcal{A}$ the CDY advantage defined as

$$\mathbf{Adv}_{\Pi}^{\text{CDY}[\Sigma]}(\mathcal{A}) = \Pr\left[\, \text{CDY}[\Sigma](\mathcal{A}) \Rightarrow \mathsf{true} \,\right].$$

**Restricted CDY and its variants.** To more accurately capture attack settings and to prove relations, we find it useful to define restricted variants of the CDY[$\Sigma$] game. A class of games of particular interest are ones that allow targeting under *any* context selector; we call this class *restricted CDY*. For a target specifier ts, let CDY$^*$[ts] be the game where the adversary is given a ciphertext and elements of a decryption context specified by ts, all selected arbitrarily, and needs to produce the remaining elements of a decryption context such that $\mathsf{AEAD.Dec}(K, N, A, C) \neq \perp$. Formally, for an AEAD scheme $\Pi$ and adversary $\mathcal{A}$, we define the CDY$^*$ advantage as

$$\mathbf{Adv}_{\Pi}^{\text{CDY}^*[\mathsf{ts}]}(\mathcal{A}) = \Pr\left[\, \text{for all } \mathsf{S},\ \text{CDY}[\mathsf{ts}, \mathsf{S}](\mathcal{A}) \Rightarrow \mathsf{true} \,\right].$$

In addition, we find it useful to define three specific variants of CDY$^*$ that allow targeting two-of-three components of a decryption context. Let $\text{CDY}^*_{\mathsf{a}}$ be the game where the adversary is given an arbitrary ciphertext $C$, key $K$, and nonce $N$, and has to produce associated data $A$ such that $\mathsf{AEAD.Dec}(K, N, A, C) \neq \perp$. Formally, for an AEAD scheme $\Pi$ and adversary $\mathcal{A}$, we define the $\text{CDY}^*_{\mathsf{a}}$ advantage as

$$\mathbf{Adv}_{\Pi}^{\text{CDY}^*_{\mathsf{a}}}(\mathcal{A}) = \Pr\left[\, \text{for all } \mathsf{S},\ \text{CDY}[\{\mathsf{k}, \mathsf{n}\}, \mathsf{S}](\mathcal{A}) \Rightarrow \mathsf{true} \,\right].$$

The $\text{CDY}^*_{\mathsf{k}}$ and $\text{CDY}^*_{\mathsf{n}}$ games are defined similarly where the adversary has to produce a valid key and nonce respectively such that decryption succeeds when the remaining inputs to decryption are pre-selected. Formally, for an AEAD scheme $\Pi$ and adversary $\mathcal{A}$, we define the $\mathrm{CDY}^*_{\mathsf{k}}$ and $\mathrm{CDY}^*_{\mathsf{n}}$ advantage as

$$\mathbf{Adv}_{\Pi}^{\mathrm{CDY}^*_{\mathsf{k}}}(\mathcal{A}) = \Pr[\,\text{for all } \mathsf{S},\ \mathrm{CDY}[\{\mathsf{n},\mathsf{a}\},\mathsf{S}](\mathcal{A}) \Rightarrow \mathsf{true}\,]\,,$$
$$\mathbf{Adv}_{\Pi}^{\mathrm{CDY}^*_{\mathsf{n}}}(\mathcal{A}) = \Pr[\,\text{for all } \mathsf{S},\ \mathrm{CDY}[\{\mathsf{k},\mathsf{a}\},\mathsf{S}](\mathcal{A}) \Rightarrow \mathsf{true}\,]\,.$$

Note that the context selector can only select *valid* ciphertexts, which sidesteps issues with formatting. Without this constraint, a context selector could select a ciphertext that has invalid padding for a scheme that requires valid padding, thereby making the notion trivial (all schemes achieve it.)

Furthermore, specific variants like $\mathrm{CDY}^*_{\mathsf{a}}$ may be trivial even with this constraint. For instance, if the ciphertext embeds the nonce, then one can pick some key $K$, some ciphertext $C$ embedding some nonce $N_1$, some other nonce $N_2$, then no $\mathrm{CDY}^*_{\mathsf{a}}$ adversary can pick associated data $A$ such that $C$ decrypts correctly under $(K, N_2, A)$. However, in the context of this restricted CDY notion, we think this is desired behavior and delegate capturing nuances like this to the unrestricted CDY notion (which can capture this by restricting to context selectors which ensure that the nonce embedded is the same as the nonce provided.)

**With context compression, CMT-3 implies restricted CDY.** A $\mathrm{CDY}[\Sigma]$ attack does not always imply a $\mathrm{CMT}[\Sigma]$ attack. Consider, for example, the "identity" AEAD that has $\mathsf{Enc}(K, N, A, M) \Rightarrow k \,\|\, n \,\|\, a \,\|\, m$ which has an immediate $\mathrm{CDY}[\Sigma]$ attack but is $\mathrm{CMT}[\Sigma]$ secure since a ciphertext can only be decrypted under one context.[4] However, continuing with the hash function analogy, we wonder if a "compression" assumption could make this implication hold. In Theorem 1 we show this statement for $\mathrm{CDY}^*[\mathsf{ts} = \varnothing]$ and CMT-3. And note that this generalizes to $\mathrm{CDY}^*[\mathsf{ts}]$ for any $\mathsf{ts}$ with an appropriate compression assumption. Notably, it holds for $\mathrm{CDY}^*_{\mathsf{a}}$ if we assume compression over associated data rather than the full context.

**Theorem 1.** *Fix some AEAD $\Pi$. Then for any adversary $\mathcal{A}$ that wins the $\mathrm{CDY}^*[\mathsf{ts} = \varnothing]$ game, we can give an adversary $\mathcal{B}$ such that*

$$\mathbf{Adv}_{\Pi}^{\mathrm{CDY}^*[\mathsf{ts}=\varnothing]}(\mathcal{A}) \le 2 \cdot \mathbf{Adv}_{\Pi}^{\mathrm{CMT\text{-}3}}(\mathcal{B}) + \mathrm{ProbBadCtx}_{\Pi}\,, \qquad (2)$$

*where $\mathrm{ProbBadCtx}_{\Pi}$ is the probability that a random decryption context, when used for encrypting a random message, is the only valid decryption context for the resulting ciphertext.*

*Proof.* This proof is adapted from Bellare and Rogaway [9, p.147], where they prove a similar theorem for hash functions. We construct an adversary $\mathcal{B}$ that randomly samples a context $(K_1, N_1, A_1)$, encrypts a random message to get a ciphertext $C$, then asks the CDY adversary $\mathcal{A}$ to produce a decryption context for $C$ to get $(K_2, N_2, A_2)$. This ciphertext generation can be viewed as a valid

[4] While the "identity" AEAD is not secure in the sense of privacy [32, §3], one can construct a secure counterexample by using a wide pseudorandom permutation [7].

```
$\mathcal{B}$:
$K_1 \leftarrow\!\!\$\ \mathcal{K}$; $N_1 \leftarrow\!\!\$\ \mathcal{N}$; $A_1 \leftarrow\!\!\$\ \mathcal{A}$
$M_1 \leftarrow\!\!\$\ \mathcal{M}$
$\mathrm{ctx}_1 \leftarrow (K_1, N_1, A_1)$
$C \leftarrow \Pi.\mathsf{Enc}(K_1, N_1, A_1, M_1)$
$\mathrm{ctx}_2 \leftarrow\!\!\$\ \mathcal{A}(C)$
If $\mathrm{ctx}_2 = \bot$:
    Return $\bot$
$(K_1, N_2, A_2) \leftarrow \mathrm{ctx}_2$
If $(K_1, N_1, A_1) = (K_2, N_2, A_2)$
    Return $\bot$
Return $(C, (K_1, N_1, A_1), (K_2, N_2, A_2))$
```

```
$\mathsf{S}$:
$K_1 \leftarrow\!\!\$\ \mathcal{K}$; $N_1 \leftarrow\!\!\$\ \mathcal{N}$; $A_1 \leftarrow\!\!\$\ \mathcal{A}$
$M_1 \leftarrow\!\!\$\ \mathcal{M}$
$C \leftarrow \Pi.\mathsf{Enc}(K_1, N_1, A_1, M_1)$
Return $C$
```

Fig. 6: Pseudocode for the CMT-3 adversary $\mathcal{B}$ and CDY$^*$ context selector $\mathsf{S}$, used in proof of Theorem 1.

CDY context selector $\mathsf{S}$ so $\mathcal{B}$ wins if the returned context is different from the one it sampled; i.e., $(K_1, N_1, A_1) \neq (K_2, N_2, A_2)$. The pseudocode for $\mathcal{B}$ and $\mathsf{S}$ is given in Figure 6 and the success probability is analyzed below.

Per the above discussion the advantage of $\mathcal{B}$ is

$$\mathbf{Adv}_{\Pi}^{\text{CMT-3}}(\mathcal{B}) = \Pr[(\mathcal{A}(C) \neq \bot) \wedge (\mathrm{ctx}_1 \neq \mathrm{ctx}_2)]\,, \tag{3}$$

where without loss of generality, we are assuming that $\mathcal{A}$ always produces a valid context or fails and produces $\bot$. But, before simplifying this equation, we need to define some terminology. First, let us define the set of valid decryption contexts for a ciphertext as

$$\Gamma(C) \coloneqq \{(K, N, A) : (\Pi.\mathsf{Dec}(K, N, A, C) \neq \bot)\}\,.$$

Now, for a given message $M$, let us also define the set of "bad" decryption contexts which when used for encrypting $M$, remain the only valid decryption context for the resulting ciphertext

$$\mathrm{BadCtxs}(M) \coloneqq \{(K, N, A) : |\Gamma(\Pi.\mathsf{Enc}(K, N, A, M))| = 1\}\,.$$

Finally, let us define the probability that a random decryption context is bad

$$\mathrm{ProbBadCtx}_{\Pi} \coloneqq \Pr\left[(K, N, A) \in \mathrm{BadCtxs}(M)\right],$$

over the choice $(K, N, A, M) \leftarrow\!\!\$\ (\mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M})$. Using this notation we can rewrite Equation 3, where the probabilities are over the choice $(K, N, A, M) \leftarrow\!\!\$\ (\mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M})$, as

$$\begin{aligned}\mathbf{Adv}_{\Pi}^{\text{CMT-3}}(\mathcal{B}) &= \Pr[(\mathcal{A}(C) \neq \bot) \wedge (\mathrm{ctx}_1 \neq \mathrm{ctx}_2)] \\ &\geq \Pr[(\mathcal{A}(C) \neq \bot) \wedge (\mathrm{ctx}_1 \neq \mathrm{ctx}_2) \wedge (\mathrm{ctx}_1 \notin \mathrm{BadCtxs}(M))]\,.\end{aligned}$$

Using conditional probability, we can rewrite this term as

$$\begin{aligned}&\Pr[\mathrm{ctx}_1 \neq \mathrm{ctx}_2 \mid (\mathcal{A}(C) \neq \bot) \wedge (\mathrm{ctx}_1 \notin \mathrm{BadCtxs}(m))] \\ &\qquad \cdot \Pr[(\mathcal{A}(C) \neq \bot) \wedge (\mathrm{ctx}_1 \notin \mathrm{BadCtxs}(m))]\,.\end{aligned}$$

```
$\mathcal{B}$:
$K_1 \leftarrow\!\$\ \mathcal{K}$; $N_1 \leftarrow\!\$\ \mathcal{N}$; $A_1 \leftarrow\!\$\ \mathcal{A}$
$M_1 \leftarrow\!\$\ \mathcal{M}$
$C \leftarrow \Pi.\mathsf{Enc}(K_1, N_1, A_1, M_1)$
$K_2 \leftarrow K_1 + 1$; $N_2 \leftarrow N_1 + 1$
$A_2 \leftarrow\!\$\ \mathcal{A}(C, K_2, N_2)$
If $A_2 = \bot$
    Return $\bot$
Return $(C, (K_1, N_1, A_1), (K_2, N_2, A_2))$
```

```
$\mathsf{S}$:
$K_1 \leftarrow\!\$\ \mathcal{K}$; $N_1 \leftarrow\!\$\ \mathcal{N}$; $A_1 \leftarrow\!\$\ \mathcal{A}$
$M_1 \leftarrow\!\$\ \mathcal{M}$
$C \leftarrow \Pi.\mathsf{Enc}(K_1, N_1, A_1, M_1)$
$K_2 \leftarrow K_1 + 1$; $N_2 \leftarrow N_1 + 1$
Return $(C, K_2, N_2)$
```

Fig. 7: Pseudocode for the CMT-3 adversary $\mathcal{B}$ and $\mathrm{CDY}^*_{\mathsf{a}}$ context selector $\mathsf{S}$, used in proof of Theorem 2.

Recall that if $\mathrm{ctx}_1 \notin \mathrm{BadCtxs}(m)$, then the adversary must choose one of at least two valid contexts, each of which are equally likely to be $\mathrm{ctx}_1$ (even conditioned on $C$). Thus the probably that it picks $\mathrm{ctx}_1$ is at most $1/2$, and so

$$\begin{aligned}\mathbf{Adv}^{\text{CMT-3}}_{\Pi}(\mathcal{B}) &\geq \frac{1}{2} \cdot \Pr[(\mathcal{A}(C) \neq \bot) \wedge (\mathrm{ctx}_1 \notin \mathrm{BadCtxs}(m))] \\ &\geq \frac{1}{2} \cdot (\Pr[\mathcal{A}(C) \neq \bot] - \Pr[\mathrm{ctx}_1 \in \mathrm{BadCtxs}(m)])\,.\end{aligned}$$

Putting it all together, we get that

$$\mathbf{Adv}^{\text{CMT-3}}_{\Pi}(\mathcal{B}) \geq \frac{1}{2} \cdot \left(\mathbf{Adv}^{\mathrm{CDY}^*[\mathsf{ts}=\varnothing]}_{\Pi}(\mathcal{A}) - \mathrm{ProbBadCtx}_{\Pi}\right),$$

and finally rearranging gives the desired result. □

**CMT-3 implies restricted variants of CDY.** We now show that if an attack against any of $\mathrm{CDY}^*_{\mathsf{k}}$, $\mathrm{CDY}^*_{\mathsf{n}}$, or $\mathrm{CDY}^*_{\mathsf{a}}$ implies an attack against CMT-3. The following theorem shows this for $\mathrm{CDY}^*_{\mathsf{a}}$, but it readily generalizes to $\mathrm{CDY}^*_{\mathsf{k}}$ and $\mathrm{CDY}^*_{\mathsf{n}}$.

**Theorem 2.** *Fix some AEAD $\Pi$ with key space $|\mathcal{K}| \geq 2$ and nonce space $|\mathcal{N}| \geq 2$. Then for any adversary $\mathcal{A}$ that wins the $\mathrm{CDY}^*_{\mathsf{a}}$ game, we can give an adversary $\mathcal{B}$ such that*

$$\mathbf{Adv}^{\mathrm{CDY}^*_{\mathsf{a}}}_{\Pi}(\mathcal{A}) = \mathbf{Adv}^{\text{CMT-3}}_{\Pi}(\mathcal{B})\,,$$

*and the runtime of $\mathcal{B}$ is that of $\mathcal{A}$.*

*Proof.* We prove this by constructing $\mathcal{B}$ such that it succeeds whenever $\mathcal{A}$ succeeds. The adversary $\mathcal{B}$ randomly samples a context $(K_1, N_1, A_1)$, encrypts a random message to get a ciphertext $C$, selects some other key $K_2$ and nonce $N_2$ and asks the $\mathrm{CDY}^*_{\mathsf{a}}$ adversary $\mathcal{A}$ to produce an associated data $A_2$ such that $(K_2, N_2, A_2)$ can decrypt $C$. This ciphertext and partial context construction can be viewed as a valid context selector $\mathsf{S}$. The pseudocode for the adversary $\mathcal{B}$ and the context selector $\mathsf{S}$ are given in Figure 7. And, notice that by construction, $\mathcal{B}$ wins whenever $\mathcal{A}$ succeeds. □

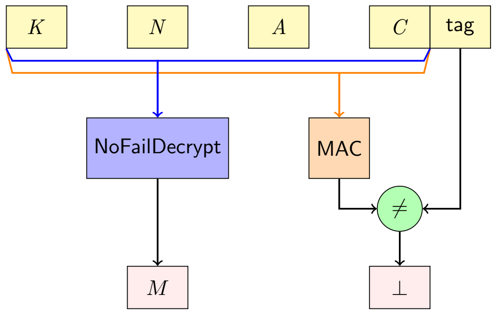

Fig. 8: Decryption structure of AEAD schemes which delegate their authenticity to a MAC. Should the MAC tag comparison fail, the routine outputs an error ($\perp$), otherwise a message is always output by NoFailDecrypt.

This approach of constructing $\mathcal{B}$ readily generalizes to $\mathrm{CDY}^*_{\mathsf{n}}$ and $\mathrm{CDY}^*_{\mathsf{k}}$. Further, notice that the $\mathcal{B}$ constructed in Figure 7 wins $\mathrm{CMT}_{\mathsf{k}}$ and $\mathrm{CMT}_{\mathsf{n}}$; and similar relations hold for adversaries $\mathcal{B}$ constructed from $\mathrm{CDY}^*_{\mathsf{n}}$ and $\mathrm{CDY}^*_{\mathsf{k}}$ adversaries. The following corollary captures these implications.

**Corollary 3.** *Fix some AEAD $\Pi$ with key space $|\mathcal{K}| \geq 2$, nonce space $|\mathcal{N}| \geq 2$, and associated data space $|\mathcal{A}| \geq 2$. Then the following three statements hold. First, for any adversary $\mathcal{A}_1$ that wins the $\mathrm{CDY}^*_{\mathsf{a}}$ game, we can give an adversary $\mathcal{B}_1$ such that*

$$\mathbf{Adv}_{\Pi}^{\mathrm{CDY}^*_{\mathsf{a}}}(\mathcal{A}_1) = \mathbf{Adv}_{\Pi}^{\mathrm{CMT}_{\mathsf{k}}}(\mathcal{B}_1) = \mathbf{Adv}_{\Pi}^{\mathrm{CMT}_{\mathsf{n}}}(\mathcal{B}_1)\,.$$

*Second, for any adversary $\mathcal{A}_2$ that wins the $\mathrm{CDY}^*_{\mathsf{n}}$ game, we can give an adversary $\mathcal{B}_2$ such that*

$$\mathbf{Adv}_{\Pi}^{\mathrm{CDY}^*_{\mathsf{n}}}(\mathcal{A}_2) = \mathbf{Adv}_{\Pi}^{\mathrm{CMT}_{\mathsf{k}}}(\mathcal{B}_2) = \mathbf{Adv}_{\Pi}^{\mathrm{CMT}_{\mathsf{a}}}(\mathcal{B}_2)\,.$$

*Third, for any adversary $\mathcal{A}_3$ that wins the $\mathrm{CDY}^*_{\mathsf{k}}$ game, we can give an adversary $\mathcal{B}_3$ such that*

$$\mathbf{Adv}_{\Pi}^{\mathrm{CDY}^*_{\mathsf{k}}}(\mathcal{A}_3) = \mathbf{Adv}_{\Pi}^{\mathrm{CMT}_{\mathsf{n}}}(\mathcal{B}_3) = \mathbf{Adv}_{\Pi}^{\mathrm{CMT}_{\mathsf{a}}}(\mathcal{B}_3)\,.$$

*And the runtimes of $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ are that of $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$, respectively.*

## 4 Context Discovery Attacks against AEAD

We show context discovery attacks on many AEAD schemes which delegate their authenticity to a non-preimage resistant MAC. Specifically, we show $\mathrm{CDY}^*_{\mathsf{a}}$ attacks on EAX [11], SIV [34], CCM [17], GCM [19], and OCB3 [28], and $\mathrm{CDY}^*_{\mathsf{n}}$ attacks on EAX [11] and GCM [19].

We say that an AEAD delegates its authenticity to a MAC if during decryption, a message is output whenever the MAC comparison succeeds. To formalize

```
OMAC$(K, M)$:
// Compute Constants
$L \leftarrow E_K(0^{128})$
$B \leftarrow 2 \cdot L$
// split into $n$-bit blocks
// & xor $B$ to the last block
Let $M_1, \ldots, M_m \leftarrow M$
$M_m \leftarrow M_m \oplus B$
// CBC-MAC Evaluation
$C_0 \leftarrow 0^{128}$
For $i = 1..m$:
    $C_i \leftarrow E_K(C_{i-1} \oplus M_i)$
Return $C_m$
```

```
EAX-Decrypt$(K, N, A, C)$:
// Separate the Tag
$C \parallel \mathsf{tag} \leftarrow C$
// Compute and Check Tag
$\mathcal{N} \leftarrow \mathsf{OMAC}(K, 0^{128} \parallel N)$
$\mathcal{H} \leftarrow \mathsf{OMAC}(K, 0^{127}1 \parallel A)$
$\mathcal{C} \leftarrow \mathsf{OMAC}(K, 0^{126}10 \parallel C)$
If $\mathsf{tag} \neq (\mathcal{N} \oplus \mathcal{H} \oplus \mathcal{C})$:
    Return $\perp$
// CTR Decryption
$r \leftarrow |C|/16$      // num blocks
For $i = 0..(r-1)$:
    $M_i \leftarrow C_i \oplus E_K(\mathcal{N} + i)$
Return $M$
```

```
$\mathcal{A}(C, K, N)$:
$C \parallel \mathsf{tag} \leftarrow C$
// Compute $\xi$
$\xi \leftarrow \mathsf{tag}$
$\xi \leftarrow \xi \oplus \mathsf{OMAC}_K(0^{128} \parallel N)$
$\xi \leftarrow \xi \oplus \mathsf{OMAC}_K(0^{126}10 \parallel C)$
// Reconstruct $A$ and Return
$A \leftarrow E_K^{-1}(\xi)$
$A \leftarrow A \oplus E_K(0^{127}1) \oplus (2 \cdot E_K(0^{128}))$
Return $(K, N, A)$
```

Fig. 9: **(Left)** Pseudocode for OMAC [11, Fig 1], used in EAX, with block-aligned inputs. **(Middle)** Pseudocode for EAX Mode [11] decryption with 128-bit tag, 128-bit nonce, and block-aligned messages and associated data. **(Right)** Pseudocode for an $\text{CDY}_{\mathsf{a}}^*$ attack on EAX.

this, we define NoFailDecrypt as a class of decryption algorithms that never fail. In other words, given a key, nonce, associated data, and ciphertext, they always produce a message. For example, ECB and CTR decryption are NoFailDecrypt algorithms.

With this terminology, we say that an AEAD delegates its authenticity to a MAC if it can be written as a combination of a MAC and a NoFailDecrypt algorithm such that if the MAC check fails, decryption fails; if instead the MAC check passes, then decryption outputs the result of NoFailDecrypt (which never fails). This structure is illustrated in Figure 8. As a concrete example, for EAX [11] (described in Figure 9), the MAC corresponds to checking the OMAC tag, and the NoFailDecrypt corresponds to the CTR decryption. In this section, we are particularly interested in schemes that compose this structure with a non-preimage resistant MAC like CMAC [18], GMAC [19, §6.4], or OMAC [11, Fig 1].

The $\text{CDY}_{\mathsf{a}}^*$ attacks we show on these schemes have the following outline. Following the definition of the game, the challenger provides the adversary with a ciphertext $C \parallel \mathsf{tag}$, a target key $K$, and a target nonce $N$, and asks it to find an associated data $A$ such that $\mathsf{Decrypt}(K, N, A, C \parallel \mathsf{tag}) \neq \perp$. Then, the adversary exploits the lack of preimage resistance to find an associated data $A$ such that $\mathsf{MAC}(K, N, A, C) = \mathsf{tag}$ and returns $A$. Since, in these schemes, the tag check passing guarantees decryption success, we get that decryption succeeds.

For EAX [11] and GCM [19], we also show $\text{CDY}_{\mathsf{n}}^*$ attacks. They proceed in a similar fashion to the $\text{CDY}_{\mathsf{a}}^*$ attacks but now the adversary finds a nonce $N$ such that $\mathsf{MAC}(K, N, A, C) = \mathsf{tag}$. But, when the nonce length is shorter than a block (which is always true with GCM, and may be true with EAX), the $\text{CDY}_{\mathsf{n}}^*$ attacks are slower than the $\text{CDY}_{\mathsf{a}}^*$ attacks.

The remainder of the section details the attacks on EAX. The attacks on SIV, CCM, GCM, and OCB3 are in Appendix B of the full version.

**$\mathrm{CDY}^*_{\mathsf{a}}$ and $\mathrm{CDY}^*_{\mathsf{n}}$ attacks on EAX.** We consider EAX over a 128-bit block cipher as defined in Bellare, Rogaway, and Wagner [11]. For simplicity, we restrict to 128-bit tag, 128-bit nonce,[5] and block-aligned messages and associated data. We note however that this is only to make the exposition simpler and is not necessary for the attack. Pseudocode for the scheme with these parameter choices is given in Figure 9.

Let's start by contextualizing the $\mathrm{CDY}^*_{\mathsf{a}}$ game. The challenger provides us with an $m$-block ciphertext $C = C_1 \cdots C_m \parallel \mathsf{tag}$, a 128-bit target key $K$, and a 96-bit target nonce $N$. And the goal is to find a 1-block associated data $A$ such that $\mathsf{EAX\text{-}Decrypt}(K, N, A, C) \neq \perp$. Notice from Figure 9 that decryption passing reduces to the tag check passing. In other words, we can rewrite the goal as finding an associated data $A$ such that

$$\mathsf{tag} = \mathsf{OMAC}_K(0^{128} \parallel N) \oplus \mathsf{OMAC}_K(0^{126}10 \parallel C) \oplus \mathsf{OMAC}_K(0^{127}1 \parallel A)\,. \quad (4)$$

We can rearrange terms to get

$$\mathsf{OMAC}_K(0^{127}1 \parallel A) = \mathsf{tag} \oplus \mathsf{OMAC}_K(0^{128} \parallel N) \oplus \mathsf{OMAC}_K(0^{126}10 \parallel C)\,.$$

Notice that the right-hand side is composed entirely of known terms, thus we can evaluate it to some constant $\xi$. Using the assumption that $A$ is 1-block, we can expand $\mathsf{OMAC}_K$ to get

$$E_K(E_K(0^{127}1) \oplus A \oplus (2 \cdot E_K(0^{128}))) = \xi\,.$$

Decrypting both sides under $K$, and solving for $A$ gives

$$A = E_K^{-1}(\xi) \oplus E_K(0^{127}1) \oplus (2 \cdot E_K(0^{128}))\,.$$

The full pseudocode for this attack is given in Figure 9.

This attack generalizes to other parameter choices. It works as is against an arbitrary-length message, an arbitrary-length tag, and an arbitrary-length nonce. In addition, this attack can also be adapted as a $\mathrm{CDY}^*_{\mathsf{n}}$ attack. We start by rewriting Equation 4 as

$$\mathsf{OMAC}_K(0^{128} \parallel N) = \mathsf{tag} \oplus \mathsf{OMAC}_K(0^{126}10 \parallel C) \oplus \mathsf{OMAC}_K(0^{127}1 \parallel C)\,,$$

and solving for $N$ as we did for $A$ above. Since $N$ is 1 block (128 bits), the reduction is similar, and the success probability remains one. If the nonce length was shorter, then assuming an idealized model like the ideal cipher model, the success probability reduces by a multiplicative factor of $2^{-f \cdot 128}$ where $f$ is the fraction of bytes we do not have control over. For example, if we only had control over 14 of the 16 bytes in an encoded block, then the success probability would reduce by $2^{-16}$.

This attack can also be adapted to provide partial control over the output plaintext. Notice that the output plaintext is a CTR decryption under the chosen

[5] EAX [11, Figure 4] supports an arbitrary length nonce; 128 bits (16 bytes) is the default in the popular Tink library [3], see [4].

```
SIV-1b-Decrypt$(K, C)$:
$\mathsf{c} \leftarrow 1^{n-64}01^{31}01^{31}$
$C_1 \parallel \mathsf{tag} \leftarrow C$
$I \leftarrow \mathsf{tag}$
$K_1 \parallel K_2 \leftarrow K$
// CTR Decryption
$\mathsf{ctr} \leftarrow I \,\&\, \mathsf{c}$
$M \leftarrow C_1 \oplus E_{K_2}(\mathsf{ctr})$
// IV Check
$I' \leftarrow \mathsf{CMAC}^*(K_1, M)$
If $I \neq I'$:
    Return $\perp$
Return $M$
```

```
CMAC*$(K, M)$:
$S \leftarrow \mathsf{CMAC}(K, 0^n)$
Return $\mathsf{CMAC}(K, S \oplus M)$

CMAC$(K, X)$:
$K_s \leftarrow 2 \cdot E_K(0^n)$
Return $E_K(K_s \oplus X)$
```

Fig. 10: **(Left)** Pseudocode for SIV Mode [34] decryption with an $n$-bit message and no associated data. **(Right)** Pseudocode for CMAC* [34] and CMAC [18] with an $n$-bit input.

key with the OMAC of the nonce as IV. Assuming an idealized model where the block cipher is an ideal cipher and OMAC is a random function, for every new choice of key and nonce, we get a random output plaintext. So, by trying $2^m$ key and nonce pairs, we can expect to control $m$ bits of the output plaintext.

## 5 Restrictive Commitment Attacks via k-Sum Problems

The previous section's $\text{CDY}^*_{\mathsf{a}}$ and $\text{CDY}^*_{\mathsf{n}}$ attacks against GCM, EAX, OCB3, SIV, and CCM immediately give rise to *permissive* $\text{CMT}_{\mathsf{k}}$ attacks against each scheme. This follows from our general result showing that $\text{CMT}_{\mathsf{k}}$ security implies $\text{CDY}^*_{\mathsf{a}}$ and $\text{CDY}^*_{\mathsf{n}}$ (Corollary 3). But this does not imply the ability to build restrictive $\text{CMT}^*_{\mathsf{k}}$, $\text{CMT}^*_{\mathsf{n}}$, or $\text{CMT}^*_{\mathsf{a}}$ attacks that require the non-adversarially controlled parts of the two decryption contexts to be identical (see Theorem 10 in the full version.)

Prior work has provided (in our terminology) $\text{CMT}^*_{\mathsf{k}}$ attacks for GCM [24,16], AES-GCM-SIV [35,29], ChaCha20/Poly1305 [24,29], XChaCha20/Poly1305 [29], and OCB3 [2]. An open question of practical interest [36] is whether there is a $\text{CMT}^*_{\mathsf{k}}$ attack against SIV. We resolve this open question, showing an attack that works in time about $2^{n/3}$. It requires new techniques related to the fast solution of $k$-sum problems, as we explain below.

**Attack on 1-block SIV.** We consider SIV over an $n$-bit block cipher (for $n \geq 64$) as defined in the draft NIST specification [34]. For ease of exposition, we restrict to the case of an $n$-bit message and no associated data, and describe how to generalize this to the multi-block case in Appendix D in the full version. Pseudocode for the scheme with these parameter choices is given in Figure 10.

Here, the $\text{CMT}^*_{\mathsf{k}}$ adversary seeks to produce a ciphertext $C = C_1 \| \mathsf{tag}$ and two $2n$-bit keys $K = K_1 \parallel K_2$ and $K' = K'_1 \parallel K'_2$ such that $\mathsf{SIV\text{-}Decrypt}(K, C) \neq \perp$ and $\mathsf{SIV\text{-}Decrypt}(K', C) \neq \perp$. Notice from Figure 10 that this reduces to two simultaneous IV checks passing which can be written as

$$\mathsf{tag} = \mathsf{CMAC}^*(K_1, C_1 \oplus E_{K_2}(\mathsf{tag}\ \&\ \mathsf{c})) = \mathsf{CMAC}^*(K_1', C_1 \oplus E_{K_2'}(\mathsf{tag}\ \&\ \mathsf{c}))$$

where $\mathsf{c} = 1^{n-64}01^{31}01^{31}$ is a constant specified by the SIV standard. Our attack strategy will be to choose $\mathsf{tag}$ arbitrarily, so we can treat this as a constant value. Towards solving for the remaining variable $C_1$, we can substitute in the definition of $\mathsf{CMAC}^*$ to get

$$\begin{aligned}\mathsf{tag} &= E_{K_1}((2 \cdot E_{K_1}(0^n)) \oplus E_{K_1}(2 \cdot E_{K_1}(0^n)) \oplus C_1 \oplus E_{K_2}(\mathsf{tag}\ \&\ \mathsf{c})) \\ &= E_{K_1'}((2 \cdot E_{K_1'}(0^n)) \oplus E_{K_1'}(2 \cdot E_{K_1'}(0^n)) \oplus C_1 \oplus E_{K_2'}(\mathsf{tag}\ \&\ \mathsf{c}))\,,\end{aligned}$$

which we can rearrange the two equalities by solving for the variable $C_1$, giving us the following:

$$\begin{aligned}C_1 &= E^{-1}_{K_1}(\mathsf{tag}) \oplus (2 \cdot E_{K_1}(0^n)) \oplus E_{K_1}(2 \cdot E_{K_1}(0^n)) \oplus E_{K_2}(\mathsf{tag}\ \&\ \mathsf{c}) \\ &= E^{-1}_{K_1'}(\mathsf{tag}) \oplus (2 \cdot E_{K_1'}(0^n)) \oplus E_{K_1'}(2 \cdot E_{K_1'}(0^n)) \oplus E_{K_2'}(\mathsf{tag}\ \&\ \mathsf{c})\,. \qquad (5)\end{aligned}$$

The above implies that it suffices now to find $K_1, K_2, K_1', K_2'$ that satisfy Equation 5. To ease notation, we define four helper functions, one for each term:

$$\begin{aligned}F_1(K_1) &\coloneqq E^{-1}_{K_1}(\mathsf{tag}) \oplus 2 \cdot E_{K_1}(0^n) \oplus E_{K_1}(2 \cdot E_{K_1}(0^n))\,, \\ F_2(K_2) &\coloneqq E_{K_2}(\mathsf{tag}\ \&\ \mathsf{c})\,, \\ F_3(K_1) &\coloneqq E^{-1}_{K_1'}(\mathsf{tag}) \oplus 2 \cdot E_{K_1'}(0^n) \oplus E_{K_1'}(2 \cdot E_{K_1}(0^n))\,, \\ F_4(K_2') &\coloneqq E_{K_2'}(\mathsf{tag}\ \&\ \mathsf{c})\,,\end{aligned}$$

and recast Equation 5 as a 4-sum problem

$$F_1(K_1) \oplus F_2(K_2) \oplus F_3(K_1') \oplus F_4(K_2') = 0\,.$$

If these were independent random functions, then we could directly apply Wagner's k-tree algorithm [38] for finding a 4-way collision (also referred to as the generalized birthday problem). But even modeling $E$ as an ideal cipher, the functions are neither random nor independent. For example, $F_1(x) = F_3(x)$ always.

Towards resolving this, we first ensure that the keys $K_1$, $K_2$, $K_1'$, and $K_2'$ are domain separated. This can be easily arranged: see Figure 11 for the pseudocode of our $\mathrm{CMT}^*_\mathsf{k}$ adversary $\mathcal{A}$ against SIV. We now turn to lower bounding $\mathcal{A}$'s advantage, which consists of two primary steps.

The first is that we argue that, in $\mathrm{CMT}^*_\mathsf{k}$ when running our adversary against SIV, the helper-function outputs are statistically close to uniform. Then, we show that Wagner's approach works for not-too-biased values.

We observe that $F_2$ and $F_4$ trivially behave as independent random functions in the ideal cipher model for $E$. The analysis for $F_1$ and $F_3$ is more involved. We use the following lemma, which bounds the distinguishing advantage between a uniform $n$-bit string and the output of a single query to either $F_1$ or $F_3$.

```
$\underline{\mathcal{A}():}$
$\mathsf{c} \leftarrow 1^{n-64}01^{31}01^{31}$
// Arbitrarily pick a tag
$\mathsf{tag} \leftarrow\!\$\ \{0,1\}^n \setminus \{0^n\}$
// Define helper functions
Def $F_1(K_1) \leftarrow E^{-1}_{K_1}(\mathsf{tag}) \oplus 2 \cdot E_{K_1}(0^n) \oplus E_{K_1}(2 \cdot E_{K_1}(0^n))$
Def $F_2(K_2) \leftarrow E_{K_2}(\mathsf{tag}\ \&\ \mathsf{c})$
Def $F_3(K_1) \leftarrow E^{-1}_{K'_1}(\mathsf{tag}) \oplus 2 \cdot E_{K'_1}(0^n) \oplus E_{K'_1}(2 \cdot E_{K_1}(0^n))$
Def $F_4(K'_2) \leftarrow E_{K'_2}(\mathsf{tag}\ \&\ \mathsf{c})$
// Generate lists
For $i = 1, \ldots, q$:
    $x \leftarrow \mathsf{encode}_{128-2}(i)$
    // Domain separate the keys
    $K_1 \leftarrow \mathsf{00} \parallel x$;  $K_2 \leftarrow \mathsf{01} \parallel x$;  $K'_1 \leftarrow \mathsf{10} \parallel x$;  $K'_2 \leftarrow \mathsf{11} \parallel x$
    // Query a row
    $L_1[i] \leftarrow F_1(K_1)$;  $L_2[i] \leftarrow F_2(K_2)$;  $L_3[i] \leftarrow F_3(K'_1)$;  $L_4[i] \leftarrow F_4(K'_2)$
// Find an 4-way collision using Wagner's k-tree algorithm [38]
res $\leftarrow \mathcal{A}.\mathsf{fourWayCollision}(L_1, L_2, L_3, L_4)$
If res $= \varnothing$:
    Return $\perp$
// Repackage the collision into ciphertext and keys
$(x_1, x_2, x_3, x_4) \leftarrow$ res
$C_1 \leftarrow F_1(x_1) \oplus F_2(x_2)$
$K_1 \leftarrow \mathsf{00} \parallel x_1$;  $K_2 \leftarrow \mathsf{01} \parallel x_2$;  $K'_1 \leftarrow \mathsf{10} \parallel x_3$;  $K'_2 \leftarrow \mathsf{11} \parallel x_4$
Return $C_1 \parallel \mathsf{tag}, K_1 \parallel K_2, K'_1 \parallel K'_2$
```

Fig. 11: Pseudocode for $\mathrm{CMT}^*_{\mathsf{k}}$ attack on SIV-1b. The fourWayCollision subroutine is defined in Appendix C in the full version.

**Lemma 4.** *Let* $\mathsf{tag} \in \{0,1\}^n \setminus \{0^n\}$ *and* $\sigma$ *be an* $n$*-bit random permutation with inverse* $\sigma^{-1}$ *and* $U$ *be the uniform random variable over* $n$ *bit strings. Define* $n$*-bit random variables (over the choice of* $\sigma$*)*

$$A \coloneqq \sigma^{-1}(\mathsf{tag}), \qquad B \coloneqq 2 \cdot \sigma(0^n), \qquad C \coloneqq \sigma(2 \cdot \sigma(0^n)),$$

*where* $\cdot$ *denotes multiplication in* $\mathrm{GF}(2^n)$*. Then no adversary that makes one query to a procedure* $P$ *can distinguish between* $P \mapsto (U,U,U)$ *and* $P \mapsto (A,B,C)$ *with probability greater than* $6 \cdot 2^{-n}$*.*

The proof proceeds by constructing identical-until-bad games and applying the *fundamental lemma of game playing* [10] to discern the distinguishing advantage. The proof appears in Appendix C in the full version.

We combine this with the following technical statement about applying Wagner's k-tree algorithm [38] to almost-random lists.

**Theorem 5.** *Let* $L$ *be a list of* $\ell$ *4-tuples* $x = (x_1, x_2, x_3, x_4)$*, where each entry* $x$ *is distinguishable from an 4-tuple of independent uniformly random values with probability at most* $\xi$*. Let* $L_1$*,* $L_2$*,* $L_3$*, and* $L_4$ *be lists of 1-index* $(x_1)$*, 2-index* $(x_2)$*, 3-index* $(x_3)$*, and 4-index* $(x_4)$ *elements of* $L$ *respectively. Then Wagner's k-tree algorithm [38] finds a solution* $(y_1, y_2, y_3, y_4) \in L_1 \times L_2 \times L_3 \times L_4$ *such* *that*

$$y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 0\,,$$

*with probability at least*

$$(1 - \ell \cdot \xi)\left(1 - \exp\left(-\frac{\ell^2 \cdot 2^{-n/3}}{8}\right)\right)\left(1 - \exp\left(1 - \frac{\ell^4 \cdot 2^{-4n/3}}{8} - \frac{2}{\ell^4 \cdot 2^{-4n/3}}\right)\right),$$

*and time at most*

$$20\ell + 4\ell^2 \cdot 2^{-n/3} + 4\mathsf{Sort}(\ell) + 2\mathsf{Sort}((1/2)\ell^2 \cdot 2^{-n/3})\,,$$

*where* $\mathsf{Sort}(k)$ *denotes the time to sort a list of* $k$ *items.*

The proof proceeds by analyzing the algorithm step-by-step and at each step applying Chernoff bounds [23] to compute a lower bound on the success probability. The proof appears in Appendix C in the full version.

With Lemma 4 and Theorem 5, we can now prove a lower bound on the advantage of the $\mathrm{CMT}^*_{\mathsf{k}}$ adversary in Figure 11.

**Theorem 6.** *Let* $\mathcal{A}$ *be the* $\mathrm{CMT}^*_{\mathsf{k}}$ *adversary against SIV over an* $n$*-bit ideal cipher* $E$*, detailed in Figure 11. It makes* $10q$ *queries to* $E$ *and takes at most*

$$35q + 4q^2 \cdot 2^{-n/3} + 4\mathsf{Sort}(q) + 2\mathsf{Sort}((1/2)q^2 \cdot 2^{-n/3}) + 11\,,$$

*time, where* $\mathsf{Sort}(k)$ *is the cost of sorting a list of* $k$ *items. Then the advantage*

$$\mathbf{Adv}^{\mathrm{CMT}^*_{\mathsf{k}}}_{\mathrm{SIV}}(\mathcal{A}) \geq \left(1 - 8q \cdot 2^{-n}\right)\left(1 - \exp\left(-\frac{q^2 \cdot 2^{-n/3}}{8}\right)\right)$$
$$\left(1 - \exp\left(1 - \frac{q^4 \cdot 2^{-4n/3}}{8} - \frac{2}{q^4 \cdot 2^{-4n/3}}\right)\right). \qquad (6)$$

*Proof.* By construction, the adversary $\mathcal{A}$ (Figure 11) wins whenever it finds a collision, so it suffices to lower bound this probability. First, the domain separation over the keys ensures that the two helper functions never query the ideal cipher with the same key. This, by the properties of the ideal cipher, ensures independence of the outputs. Second, $F_2$ and $F_4$ call the ideal cipher only once on a fixed output under a new key each invocation, so their outputs are indistinguishable from an $n$-bit uniform random value. Third, $F_1$ and $F_3$ call the ideal cipher three times under the same key each invocation. However, applying Lemma 4 gives us that their outputs are distinguishable from an $n$-bit uniform random value with probability at most $6 \cdot 2^{-n}$. So, by the union bound, a row of outputs $(F_1(K_1), F_2(K_2), F_3(K_1'), F_4(K_2'))$ is distinguishable from four independent, uniformly random outputs with probability at most $8 \cdot 2^{-n}$. Then, Theorem 5 tells us that the function $\mathsf{fourWayCollision}$ called by $\mathcal{A}$ finds a collision with probability at least that of Equation 6.

It remains to analyze the cost of the adversary $\mathcal{A}$. First, it costs 2 operations to initialize $\mathsf{c}$ and $\mathsf{tag}$. Second, since each loop iteration costs 15 operations, the loop costs $15q$ operations. Third, from Theorem 5, finding a 4-way collision on four lists of size $q$ using Wagner's k-tree algorithm [38] costs at most

$$20q + 4q^2 \cdot 2^{-n/3} + 4\mathsf{Sort}(q) + 2\mathsf{Sort}((1/2)q^2 \cdot 2^{-n/3})$$

operations. Fourth, repackaging the collision and returning costs 9 operations. So, the runtime is at most

$$35q + 4q^2 \cdot 2^{-n/3} + 4\mathsf{Sort}(q) + 2\mathsf{Sort}((1/2)q^2 \cdot 2^{-n/3}) + 11\,.$$

Finally, since each loop iteration makes 10 ideal cipher queries, the algorithm makes $10q$ queries. □

In the following corollary, we show that when the adversary makes approximately $2^{n/3}$ queries, it can win $\mathrm{CMT}^*_{\mathsf{k}}$ against SIV with high probability, taking time approximately $2^{n/3}$.

**Corollary 7.** *Let $\mathcal{A}$ be the $\mathrm{CMT}^*_{\mathsf{k}}$ adversary against SIV over an $n$-bit ideal cipher $E$, detailed in Figure 11 with $q = 10 \cdot 2^{n/3}$. It makes $100 \cdot 2^{n/3}$ queries to $E$ and takes at most*

$$750 \cdot 2^{n/3} + 4\mathsf{Sort}(10 \cdot 2^{n/3}) + 2\mathsf{Sort}(50 \cdot 2^{n/3}) + 11\,,$$

*time, where $\mathsf{Sort}(n)$ is the cost of sorting a list of $n$ items. Then*

$$\mathbf{Adv}^{\mathrm{CMT}^*_{\mathsf{k}}}_{\mathrm{SIV}}(\mathcal{A}) \geq \left(1 - 80 \cdot 2^{-2n/3}\right)\left(1 - \exp\left(-12.5 \cdot 2^{n/3}\right)\right)\left(1 - \exp\left(-1249\right)\right).$$

## 6 Related Work

Key commitment for authenticated encryption was introduced in Farshim, Orlandi, and Rosie [22] through full robustness (FROB), which in turn was inspired by key robustness notions in the public key setting by Abdalla, Bellare, and Neven [1] and refined by Farshim et al. [21]. The FROB game asks that a ciphertext only be able to decrypt under a single key. However, the FROB game was defined for randomized authenticated encryption. Grubbs, Lu, and Ristenpart [24] adapted the FROB game to work with associated data, where they ask that a ciphertext only be able to decrypt under a single key (with no constraints on the associated data.) This notion was further generalized by Bellare and Hoang [5] to the nonce-based setting, with their committing security 1 (CMT-1) definition. The CMT-1 game asks that a ciphertext only be able to decrypt under a single key (with no constraints on the nonce nor the associated data.)

The real-world security implications of key commitment were first highlighted by Dodis et al. [16] where they exploited the lack of key commitment when encrypting attachments in Facebook Messenger's message franking protocol [20] to send abusive images that cannot be reported. Albertini et al. [2] generalized this attack from images to other file formats and called attention to more settings where lack of key commitment can be exploited to defeat integrity. While both these attacks targeted integrity, Len, Grubbs, and Ristenpart [29] introduced partitioning oracle attacks and showed how to use them for password guessing attacks by exploiting lack of key commitment to obtain large speedups over standard dictionary attacks, endangering confidentiality.

Proposals for constructing key committing ciphers also started in the Farshim, Orlandi, and Rosie paper [22] where they showed that single-key Encrypt-then-MAC, Encrypt-and-MAC, and MAC-then-Encrypt constructions produce key committing ciphers, when the MAC is collision-resistant. Grubbs, Lu, and Ristenpart [24] showed that the Encode-then-Encipher construction [8] was key committing. Dodis et al. [16] proposed a faster compression function-based key committing AEAD construction termed *encryptment*, and also discussed the closely related Duplex construction [12], which is also key committing. Albertini et al. [2] formally analyzed the folklore padding zeroes and key hashing transforms and showed that they produce key committing AEAD at a lower performance cost than prior constructions. Bellare and Hoang [5] constructed key committing variants of GCM and GCM-SIV termed CAU-C1 and CAU-SIV-C1, and generic transforms UtC and RtC that can be used to turn unique-nonce secure and nonce-reuse secure AEAD schemes respectively into key committing AEAD schemes.

The potential risk of delegating authenticity of an AEAD entirely to a non-collision-resistant MAC is folklore. Farshim, Orlandi, and Rosie [22] who introduced the notion of committing AEAD also cautioned against using non-collision-resistant MACs and CBC-MAC in particular.

On February 7, 2023, NIST announced the selection of the Ascon family for lightweight cryptography standardization [37]. The finalist version of Ascon [15] specifies two AEAD parameter sets ASCON-128 and ASCON-128A. Both parameter sets specify a 128 bit tag, which by the birthday bound, upper bounds the committing security at 64 bits. But, since the underlying algorithm is a variant of the Duplex construction with a 320-bit permutation, and the same specification specifies parameters a hash function with 128-bit collision resistance, one can specify an AEAD with 128-bit committing security by tweaking parameters.

**Concurrent work.** In independent and concurrent work made public very recently, Chan and Rogaway [14] introduced a new definitional framework for committing AE. Their goal is to capture multiple different types of commitment attacks—what they call *misattributions*, or an adversary being able to construct distinct pairs $(K, N, A, M)$ and $(K', N', A', M')$ that both "explain" a single ciphertext $C$—in a unified way. Their main definition only captures commitment to an entire $(K, N, A, M)$ tuple; but in [14, Appendix A], they briefly describe an extension to only require commitments to a subset of the values.

The extended version of their framework is similar to our CMT[$\Sigma$] definition. While both frameworks aim to capture granular win conditions beyond CMT-3, they are orthogonal. Their framework models the multi-key setting with many randomly chosen unknown-to-the-adversary, known-to-the-adversary, and chosen-by-the-adversary keys. While our CMT[$\Sigma$] captures the distinction between *permissive* and *restrictive* notions, and settings that impose restrictions on the nonce and associated data. We also introduce the notion of context discoverability and describe its relation to CMT[$\Sigma$].

Chan and Rogaway [14] also independently observed that AEAD with non-preimage resistant MACs are vulnerable to commitment attacks and show attacks on GCM and OCB3 similar to the ones we give in Section 4.

**Acknowledgments**

We thank the anonymous reviewers of Eurocrypt 2023 for their feedback. Sanketh thanks Giacomo Pope for helpful discussions. This work was supported in part by NSF grant CNS #2120651, and the NSF Graduate Research Fellowship under Grant No. DGE-2139899.

## References

1. Abdalla, M., Bellare, M., Neven, G.: Robust encryption. In: Micciancio, D. (ed.) Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings. Lecture Notes in Computer Science, vol. 5978, pp. 480–497. Springer (2010). https://doi.org/10.1007/978-3-642-11799-2_28
2. Albertini, A., Duong, T., Gueron, S., Kölbl, S., Luykx, A., Schmieg, S.: How to abuse and fix authenticated encryption without key commitment. USENIX Security 2022 (2022), https://ia.cr/2020/1456
3. Ambrosin, M., et al.: Tink (2021), https://github.com/google/tink/releases/tag/v1.6.1
4. Ambrosin, M., et al.: Tink EAX key manager (2021), https://github.com/google/tink/blob/v1.6.1/java_src/src/main/java/com/google/crypto/tink/aead/AesEaxKeyManager.java#L115-L116
5. Bellare, M., Hoang, V.T.: Efficient schemes for committing authenticated encryption. In: Dunkelman, O., Dziembowski, S. (eds.) 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. Lecture Notes in Computer Science, vol. 13276, pp. 845–875. Springer (2022). https://doi.org/10.1007/978-3-031-07085-3_29
6. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. J. Cryptol. **21**(4), 469–491 (2008). https://doi.org/10.1007/s00145-008-9026-x
7. Bellare, M., Ristenpart, T., Rogaway, P., Stegers, T.: Format-preserving encryption. In: Jr., M.J.J., Rijmen, V., Safavi-Naini, R. (eds.) Selected Areas in Cryptography, 16th Annual International Workshop, SAC 2009, Calgary, Alberta, Canada, August 13-14, 2009, Revised Selected Papers. Lecture Notes in Computer Science, vol. 5867, pp. 295–312. Springer (2009). https://doi.org/10.1007/978-3-642-05445-7_19
8. Bellare, M., Rogaway, P.: Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In: Okamoto, T. (ed.) 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. Lecture Notes in Computer Science, vol. 1976, pp. 317–330. Springer (2000). https://doi.org/10.1007/3-540-44448-3_24, https://cseweb.ucsd.edu/~mihir/papers/ee.pdf

9. Bellare, M., Rogaway, P.: Introduction to Modern Cryptography (2005), https://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf
10. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) 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. Lecture Notes in Computer Science, vol. 4004, pp. 409–426. Springer (2006). https://doi.org/10.1007/11761679_25
11. Bellare, M., Rogaway, P., Wagner, D.A.: The EAX mode of operation. In: Roy, B.K., Meier, W. (eds.) Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers. Lecture Notes in Computer Science, vol. 3017, pp. 389–407. Springer (2004). https://doi.org/10.1007/978-3-540-25937-4_25, https://web.cs.ucdavis.edu/~rogaway/papers/eax.pdf
12. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Duplexing the sponge: Single-pass authenticated encryption and other applications. In: Miri, A., Vaudenay, S. (eds.) Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7118, pp. 320–337. Springer (2011). https://doi.org/10.1007/978-3-642-28496-0_19
13. Black, J., Rogaway, P., Shrimpton, T.: Black-box analysis of the block-cipher-based hash-function constructions from PGV. In: Yung, M. (ed.) Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2442, pp. 320–335. Springer (2002). https://doi.org/10.1007/3-540-45708-9_21
14. Chan, J., Rogaway, P.: On committing authenticated-encryption. In: European Symposium on Research in Computer Security. pp. 275–294. Springer (2022), https://ia.cr/2022/1260
15. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1.2: Submission to nist (may 2021), https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/ascon-spec-final.pdf
16. Dodis, Y., Grubbs, P., Ristenpart, T., Woodage, J.: Fast message franking: From invisible salamanders to encryptment. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10991, pp. 155–186. Springer (2018). https://doi.org/10.1007/978-3-319-96884-1_6
17. Dworkin, M.: Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality. NIST Special Publication 800-38C (2004). https://doi.org/10.6028/NIST.SP.800-38C
18. Dworkin, M.: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. NIST Special Publication 800-38B (2005). https://doi.org/10.6028/NIST.SP.800-38B

19. Dworkin, M.: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D (2017). https://doi.org/10.6028/NIST.SP.800-38D
20. Facebook: Messenger secret conversations: Technical whitepaper (2017), https://about.fb.com/wp-content/uploads/2016/07/messenger-secret-conversations-technical-whitepaper.pdf
21. Farshim, P., Libert, B., Paterson, K.G., Quaglia, E.A.: Robust encryption, revisited. In: Kurosawa, K., Hanaoka, G. (eds.) Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7778, pp. 352–368. Springer (2013). https://doi.org/10.1007/978-3-642-36362-7_22
22. Farshim, P., Orlandi, C., Rosie, R.: Security of symmetric primitives under incorrect usage of keys. IACR Trans. Symmetric Cryptol. **2017**(1), 449–473 (2017). https://doi.org/10.13154/tosc.v2017.i1.449-473
23. Goemans, M.: Chernoff bounds, and some applications (2015), https://math.mit.edu/~goemans/18310S15/chernoff-notes.pdf
24. Grubbs, P., Lu, J., Ristenpart, T.: Message franking via committing authenticated encryption. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III. Lecture Notes in Computer Science, vol. 10403, pp. 66–97. Springer (2017). https://doi.org/10.1007/978-3-319-63697-9_3
25. Harkins, D.: Synthetic Initialization Vector (SIV) Authenticated Encryption Using the Advanced Encryption Standard (AES). Request for Comments - Informational (2008), https://datatracker.ietf.org/doc/rfc5297/
26. Kilian, J., Rogaway, P.: How to protect DES against exhaustive key search (an analysis of DESX). J. Cryptol. **14**(1), 17–35 (2001). https://doi.org/10.1007/s001450010015
27. Krawczyk, H.: The OPAQUE Asymmetric PAKE Protocol. Tech. rep. (October 2019), https://datatracker.ietf.org/doc/draft-krawczyk-cfrg-opaque/03/
28. Krovetz, T., Rogaway, P.: The OCB Authenticated-Encryption Algorithm. Request for Comments - Informational (2014), https://datatracker.ietf.org/doc/rfc7253/
29. Len, J., Grubbs, P., Ristenpart, T.: Partitioning oracle attacks. In: Bailey, M., Greenstadt, R. (eds.) 30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021. pp. 195–212. USENIX Association (2021), https://ia.cr/2020/1491
30. Namprempre, C., Rogaway, P., Shrimpton, T.: Reconsidering generic composition. In: Nguyen, P.Q., Oswald, E. (eds.) Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8441, pp. 257–274. Springer (2014). https://doi.org/10.1007/978-3-642-55220-5_15
31. Ristenpart, T., Shacham, H., Shrimpton, T.: Careful with composition: Limitations of indifferentiability and universal composability. IACR Cryptol. ePrint Arch. p. 339 (2011), http://ia.cr/2011/339
32. Rogaway, P.: Authenticated-encryption with associated-data. In: Atluri, V. (ed.) Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002. pp. 98–107. ACM (2002). https://doi.org/10.1145/586110.586125

33. Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem **4004**, 373–390 (2006). `https://doi.org/10.1007/11761679_23`
34. Rogaway, P., Shrimpton, T.: The SIV Mode of Operation for Deterministic Authenticated-Encryption (Key Wrap) and Misuse-Resistant Nonce-Based Authenticated-Encryption (2007), `https://web.cs.ucdavis.edu/~rogaway/papers/siv.pdf`, draft 0.32
35. Schmieg, S.: Invisible Salamanders in AES-GCM-SIV (2020), `https://keymaterial.net/2020/09/07/invisible-salamanders-in-aes-gcm-siv/`
36. Sophie, indistinguishable from random noise (@SchmiegSophie): (2020), `https://web.archive.org/web/20200909134511/https://twitter.com/SchmiegSophie/status/1303690812933382148`, via Twitter.
37. Team, N.L.C.: NIST announces the selection of the Ascon family for lightweight cryptography standardization (feb 2023), `https://www.nist.gov/news-events/news/2023/02/lightweight-cryptography-standardization-process-nist-selects-ascon`
38. Wagner, D.A.: A generalized birthday problem. In: Yung, M. (ed.) Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2442, pp. 288–303. Springer (2002), `https://people.eecs.berkeley.edu/~daw/papers/genbday.html`
39. Wiener, M.J.: The full cost of cryptanalytic attacks. J. Cryptol. **17**(2), 105–124 (2004). `https://doi.org/10.1007/s00145-003-0213-5`