Constructing Committing and Leakage-Resilient Authenticated Encryption

Patrick Struck[1] and Maximiliane Weishäupl[2]

[1] Universität Konstanz, Konstanz, Germany, patrick.struck@uni-konstanz.de
[2] Universität Regensburg, Regensburg, Germany, maximiliane.weishaeupl@ur.de

Abstract. The main goal of this work is to construct authenticated encryption (AE) that is both committing and leakage-resilient. As a first approach for this we consider generic composition as a well-known method for constructing AE schemes. While the leakage resilience of generic composition schemes has already been analyzed by Barwell et al. (Asiacrypt'17), for committing security this is not the case. We fill this gap by providing a separate analysis of the generic composition paradigms with respect to committing security, giving both positive and negative results: By means of a concrete attack, we show that Encrypt-then-MAC is not committing. Furthermore, we prove that Encrypt-and-MAC is committing, given that the underlying schemes satisfy security notions we introduce for this purpose. We later prove these new notions achievable by providing schemes that satisfy them. MAC-then-Encrypt turns out to be more difficult due to the fact that the tag is not outputted alongside the ciphertext as it is done for the other two composition methods. Nevertheless, we give a detailed heuristic analysis of MAC-then-Encrypt with respect to committing security, leaving a definite result as an open task for future work. Our results, in combination with the fact that only Encrypt-then-MAC yields leakage-resilient AE schemes, show that one cannot obtain AE schemes that are both committing and leakage-resilient via generic composition. As a second approach for constructing committing and leakage-resilient AE, we develop a generic transformation that turns an arbitrary AE scheme into one that fulfills both properties—though only a slightly weakened form of leakage resilience. The transformation relies on a keyed function that is both binding, i.e., it is hard to find key-input pairs that result in the same output, leakage-resilient pseudorandom and unpredictable.

Keywords: Authenticated Encryption · Committing Security · Leakage Resilience

*__Errata:__ We noticed that the version published in ToSC 2024(1) contains an error. The transform given there does not achieve the claimed leakage resilience security notion. In this version, we give a different transform that restores the initial claim for a slightly weaker leakage resilience security notion. A more detailed errata about the mistake and fix is given in Section 4. The results in Section 3 are unaffected.

We thank the authors of [BMOS17] for helpful feedback regarding their leakage security notions. We further thank the anonymous reviewers for the valuable feedback. Patrick Struck acknowledges support by the Hector Foundation II. Work of Maximiliane Weishäupl was funded by the German Federal Ministry of Education and Research (BMBF) under the project Quant-ID (16KISQ111).

1 Introduction

Authenticated encryption (AE) with associated data is the modern day version of what cryptography was historically: a method to secure the communication between two parties. Classically, secure communication is understood to entail confidentiality and authenticity. The former prevents anyone except the rightful receiver of a ciphertext to recover the message from it, while the latter ensures that the ciphertext originates indeed from the claimed sender and was not manipulated during transmission. In an AE scheme, the ciphertext is generated by encryption of the message under a context (K, N, A), consisting of a key K, a nonce N, and associated data A. To achieve the above security goals, both the message and associated data need to be authenticated, whereas confidentiality is required only for the message.

Besides the standard security notions—confidentiality and authenticity—there are several, more advanced security notions. This encompasses, for instance, security against related-key attacks [FKOS22] which enables the adversary to obtain ciphertexts generated by keys that exhibit a certain relation, anonymous AE [CR19] where ciphertexts have to conceal the nonce to provide anonymity of the communicating parties, leakage-resilient AE [BMOS17] where the adversary can obtain extra information via side-channels, and committing security [BH22] which prevents adversaries from finding ciphertexts that decrypt under more than one key. These advanced security notions are typically analyzed isolated and not in conjunction. However, depending on the use-case, a joint consideration of multiple security notions might be sensible. As there is little analysis on the interplay between the notions, unwanted effects might occur: While there are transformations, which one applies to an AE scheme to obtain certain security properties, concatenating them might not have the desired outcome. A successive application of two or more transformations could result in a scheme that is not secure with respect to all of the security notions, but—in the worst case—only the notion corresponding to the last transformation that was applied. This stems from the fact that these transformations are usually proven to achieve the desired security property while maintaining the standard AE security, i.e., the last transformation might subvert the additional guarantees from the previous ones. On the other hand, it is possible that a transformation achieves security with respect to additional notions besides the one it was developed for. This would make further changes to the scheme redundant and might result in unnecessary overhead. In total, deepening our understanding of the relation between advanced security notions brings various benefits with it.

Two areas that have gained a lot of attention are leakage resilience [BMOS17, DJS19, DEM$^+$20, BPPS17, BGP$^+$19, DM21, DM19, MOSW15, BGPS21] and committing security [BH22, CR22, MLGR23, KSW23, DMVA23], though they were never considered in conjunction. We focus on developing AE schemes that fulfill both properties. Firstly, having such schemes allows deployment in different scenarios even if each of it requires only one of the properties. Secondly, we think that this concrete combination of security notions has also practical relevance: Considering key-recovery attacks on embedded devices, both leakage-resilience and committing security are necessary properties. More precisely, consider an AE scheme that is either leakage-resilient or committing but not both. Using side-channel attacks, an adversary might be able extract the secret key if the scheme is only committing secure. If the scheme is only leakage-resilient an adversary could apply a *Partitioning Oracle Attack* [LGR21] to learn the key: the adversary crafts ciphertexts that are valid for a subset of keys. If decryption succeeds, it knows that the target key is within this set and the attack continues using smaller and smaller subsets until it finds the key.

Towards the goal of developing schemes that fulfill both notions, we start by considering generic composition as a well-known method of constructing AE schemes. Here, an AE scheme is obtained by "gluing together" a symmetric encryption (SE) scheme and a message authentication code (MAC) such that security of the AE scheme follows from security of the

two underlying components. There are three general methods: Encrypt-and-MAC (E&M), Encrypt-then-MAC (EtM), and MAC-then-Encrypt (MtE). A common misconception is that only EtM is secure and the other two approaches (E&M and MtE) are not. While initial work [BN00] indeed showed that only EtM is secure in general, later work [NRS14] provided a more fine-grained analysis. Here, associated data was taken into account as an input to the MAC and encryption schemes based on random IVs or nonces were considered. This led to the development of several secure AE schemes derived from the three classical composition methods: eight A-schemes that are composed of an IV-based encryption scheme and a vector-MAC; and three N-schemes which rely on nonce-based encryption for the underlying scheme and again a vector-MAC. Throughout this work, we will mainly focus on the latter schemes N1, N2, and N3, which follow the E&M, EtM, and MtE approach, respectively. The three N-schemes have been analyzed both in the leakage setting [BMOS17] and against related-key attacks [FKOS22]. However, they have yet to be analyzed with respect to their committing security.

1.1 Contribution

We first study the committing security of generic composition.[1] We show that N2 (and EtM in general) does not achieve committing security. The fact that N2 authenticates the ciphertext together with the circumstance that colliding ciphertexts are easy to find—due to encryption schemes being reversible by design—enables a generic attack. For N1 (following E&M), we prove that committing security can be reduced to corresponding collision resistance properties of the underlying encryption and MAC.[2] We call these properties wCR and CR and show that they are achievable security notions by providing instances that fulfill them—namely, the symmetric encryption scheme and MAC underlying Slae [DJS19]. To check this, we first note that Slae follows the FGHF′ construction and hence one can further break down the notions wCR and CR to the functions $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$, and $\mathcal{F}'$. We then prove that the instantiations for these functions used in Slae fulfill the desired properties. Our results for N3 (following MtE) are not that definite as the ones for N1 and N2: The main difficulty lies in the fact that the tag gets encrypted alongside the message, whereas for N1 and N2, the tag gets appended to the ciphertext—thus any committing attack against N1 and N2 requires identical tags whereas those against N3 do not. We give an heuristic analysis, narrowing down the possible attacks and discussing a number of common attack strategies. However, a complete analysis of MtE with respect to committing security is still an open task and left for future work. Nevertheless, we can conclude that committing and leakage-resilient AE cannot be built via generic composition, as the only leakage-resilient generic composition method is Encrypt-then-MAC [BMOS17], which we prove to be not secure with respect to committing security. As a second approach for building committing and leakage-resilient AE, we turn towards generic transformations as a way to achieve the desired security properties. We observe that none of the existing transformations from the literature are suitable for our purpose. The problem is that all of them hash the key, hence the leakage resilience of a scheme might be lost after its application if the hash function leaks information. To address this problem, we develop a generic transformation that turns an arbitrary AE scheme, a hash function, and a keyed function into an AE scheme that is both leakage-resilient and committing. Note that for leakage-resilience, we consider the LAE notion from Barwell et al. [BMOS17], however, slightly weakened adopting the approach by Shrimpton and Terashima [ST13] for AE

[1] We focus on the N-schemes, though it is easy to see that the results also apply to the A-schemes: In case of committing security, the difference between unique nonces and random IVs becomes obsolete as the adversary can choose them at will.

[2] In order to get a non-trivial bound for the symmetric encryption scheme, the message length needs to be of sufficient length (e.g., 128 bits). In light of real-world committing attacks, like the *Facebook message franking attack* [DGRW18], this is a reasonable assumption.

security. At the core of the transformation is the keyed function which needs to be both pseudorandom and unpredictable under leakage as well as binding. We show existence of such a function by proving that the sponge-based function used in SLAE fulfills the binding property—pseudorandomness and unpredictability under leakage was already proven in [DJS19]. For the underlying AE scheme and hash function we rely on default AE security and collision resistance, respectively.

In total, our consideration of committing security and leakage resilience of AE schemes in conjunction, yields both negative and positive results. On the negative side, our results indicate that committing security and leakage resilience do not work well together for generic composition. On the positive side, we develop a simple generic transformation that achieves both security notions, thereby meeting our goal of building committing and leakage-resilient AE.

1.2 Related Work

Generic composition was initially introduced in [BN00] as a method to construct AE schemes and refined in [NRS14]. It is, especially from a theoretical point of view, an important method and was studied in various settings [FKOS22, BPP18, Ber23, BMOS17, KS20].

The necessity of committing security for AE schemes is due to real-world attacks that exploit the lack thereof [DGRW18, LGR21, ADG+22]. The first committing security notions for AE schemes were formalized in [BH22]. Later works [CR22, MLGR23] provided a more fine-grained framework for committing security notions. Several works show committing attacks against various AE schemes: GCM/AES-GCM-SIV [BH22], GCM/OCB [CR22], CCM/EAX/SIV [MLGR23], AEZ [CFGI+23], and the NIST LWC finalists [KSW23]. However, there are also positive results: CAU [BH22], CTX [CR22], the sponge-based NIST LWC finalists ASCON, ISAP, and SCHWAEMM [KSW23], SPONGEWRAP [DFG23], and a sponge-based AE scheme based on SHAKE [DMVA23] were proven to be committing.

Leakage-resilient cryptography was formalized in [DP08], following the "Only Computation Leaks Information" paradigm [MR04]. Security notions for authenticated encryption incorporating leakage were developed in [BMOS17], which also analyzed the leakage resilience of generic composition, showing inherent weaknesses for Encrypt-and-MAC and MAC-then-Encrypt. Several works developed AE schemes that are designed to be leakage-resilient: ISAP [DEM+17, DEM+20], SLAE/FGHF′ [DJS19, KS20], TEDT [BGP+19], ROMULUS-T [IKM+21], and SIVAT [BMOS17]. Furthermore, there are results concerning the leakage resilience of existing AE modes: PYJAMASK, PHOTON-BEETLE, ASCON, SPOOK, ISAP, and TEDT, are analyzed in [BBC+20], which shows different results regarding what parts need to be protected against leakage. Similarly, [VCS22] shows that out of the NIST LWC finalists, three modes (ASCON, ISAP, and ROMULUS-T) are advantageous when hardening their implementations against leakage. Lastly, [GPPS20] is worth highlighting, which provides a methodology that allows to analyze the leakage resilience of duplex sponges, e.g., ISAP, ASCON, GIBBON, and TEDTSPONGE—two of which were also shown to be committing.

Overall, there are many results with respect to either notion—for ASCON/ISAP even for both[3] individually—we are not aware of any work explicitly targeting both of them.

2 Preliminaries

Notation. By $\{0,1\}^*$, we denote the set of bit strings with arbitrary length. The empty bit string is written as ε. For a bit string X and an integer y, $\lceil X \rceil_y$ (resp. $\lfloor X \rfloor_y$) denote the leftmost (resp. rightmost) y bits of X. Concatenation of bit strings X and Y is written

[3] ROMULUS was also analyzed with respect to both leakage resilience and committing security [VCS22, KSW23], though for different variants (ROMULUS-N and ROMULUS-T).

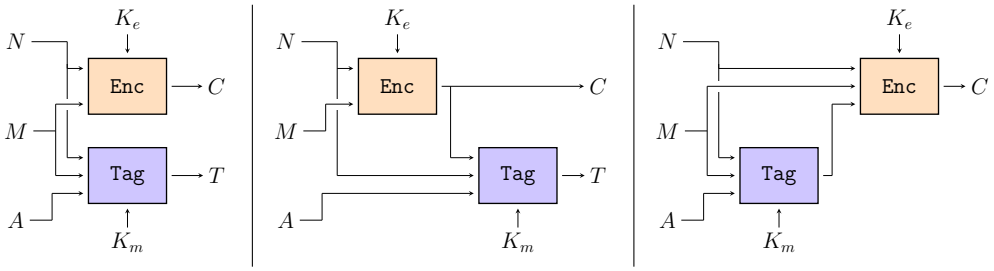

Figure 1: The AE schemes N1 (left), N2 (middle), and N3 (right) [NRS14] in terms of an underlying symmetric encryption scheme $(\mathtt{Enc}, \mathtt{Dec})$ and MAC $(\mathtt{Tag}, \mathtt{Ver})$.

as $X \parallel Y$. We use game-based proofs [BR06] and write $\mathsf{G}(\mathcal{A}) \to x$, to denote that the output of game G, when played by $\mathcal{A}$, is x. By $\mathcal{A}^{\mathsf{G}} \to x$, we denote that $\mathcal{A}$ outputs x, when playing game G. Throughout this work, we write IV for some public, fixed initialization vector used in the constructions.

Primitives. The focus of this work is on authenticated encryption (AE) scheme with associated data, which is a pair of two algorithms $(\mathtt{Enc}, \mathtt{Dec})$. The encryption algorithm $\mathtt{Enc}\colon \mathcal{K}\times\mathcal{N}\times\mathcal{A}\times\mathcal{M} \to \mathcal{C}$ takes a key K, a nonce N, associated data A, and a message M as input, and outputs a ciphertext C. The decryption algorithm $\mathtt{Dec}\colon \mathcal{K}\times\mathcal{N}\times\mathcal{A}\times\mathcal{C} \to \mathcal{M}\cup\{\perp\}$ takes a key K, a nonce N, associated data A, and a ciphertext C as input, and outputs a message M or a special symbol $\perp$. The key space, nonce space, associated data space, message space, and ciphertext space are denoted by the sets $\mathcal{K}$, $\mathcal{N}$, $\mathcal{A}$, $\mathcal{M}$, and $\mathcal{C}$, respectively. Throughout this work, we consider these sets to be bit strings of certain length, more precisely, $\mathcal{K} = \{0,1\}^\kappa$, $\mathcal{N} = \{0,1\}^\nu$, $\mathcal{A} = \{0,1\}^*$, $\mathcal{M} = \{0,1\}^*$, and $\mathcal{C} = \{0,1\}^* \times \{0,1\}^\tau$. We abuse notation and write either (C, T) or $C \parallel T$. The length of a message is denoted by m. *Correctness* of an AE scheme means that for any $(K, N, A, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}$, we have $\mathtt{Dec}(K, N, A, \mathtt{Enc}(K, N, A, M)) = M$. All schemes that we consider satisfy the *tidyness* property [NRS14]: it holds that $M = \mathtt{Dec}(K, N, A, C)$ implies that $C = \mathtt{Enc}(K, N, A, M)$. A symmetric encryption scheme is defined equivalently, except that there is no associated data and decryption does not return $\perp$. Following [MLGR23], the triple (K, N, A) is called a *context*.

A message authentication code (MAC) consists of two algorithms $(\mathtt{Tag}, \mathtt{Ver})$. The tagging algorithm $\mathtt{Tag}\colon \mathcal{K} \times \mathcal{X} \to \{0,1\}^\tau$ takes as input a key K and an element X. It returns a tag T of size $\{0,1\}^\tau$. The verification algorithm $\mathtt{Ver}\colon \mathcal{K} \times \mathcal{X} \times \{0,1\}^\tau \to \{0,1\}$ takes as input a key K, an element X, and a tag T and outputs 1 indicating that the input is valid, or otherwise 0. Throughout this work, we consider $\mathcal{K} = \{0,1\}^\kappa$ and $\mathcal{X} = \{0,1\}^*$. *Correctness* of a MAC means that for any $(K, X) \in \mathcal{K} \times \mathcal{X}$, it holds that $\mathtt{Ver}(K, X, \mathtt{Tag}(K, X)) = 1$.

The N-schemes [NRS14] construct an AE scheme from a symmetric encryption scheme and a MAC. Figure 1 illustrates the schemes N1, N2, and N3, while their pseudocodes are provided in Figure 14.

Sponges. Sponges are a tool to construct primitives. They rely on an n-bit state S that is constantly updated by applying a permutation $\pi\colon \{0,1\}^n \to \{0,1\}^n$ or a transformation $\rho\colon \{0,1\}^n \to \{0,1\}^n$ to it. A P-Sponge is centered around a permutation whereas a T-Sponge is centered around a transformation. In this work, we focus on the latter. In between two invocations of the permutation/transformation the sponge can *absorb* an input by XORing it to the first r bits of the state S or *squeeze* an output by outputting the first r bits of the state S. In the simplest setting—which is all we need in this work—the sponge will first absorb the whole input by constantly absorbing it r bits at a time and

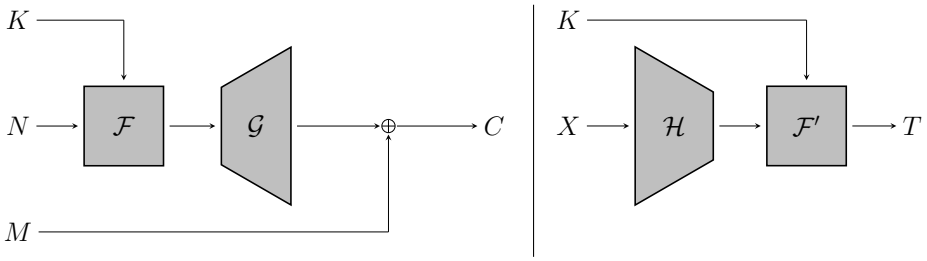

Figure 2: The symmetric encryption scheme $\textsc{Se}[\mathcal{F}, \mathcal{G}]$ (left), composed of a function $\mathcal{F}$ and a pseudorandom generator $\mathcal{G}$, and the MAC $\textsc{Mac}[\mathcal{H}, \mathcal{F}']$ (right), composed of a hash function $\mathcal{H}$ and a function $\mathcal{F}'$.

then squeeze the output r bits at a time. The capacity $c = n - r$ is the part of the state that is neither modified by the input during the absorption phase nor outputted in the squeezing phase. For the sponge-based proofs, the underlying transformation is assumed to be ideal, i.e., a random transformation to which the adversary gets oracle access.

In this work, we consider the sponge-based AE scheme SLAE [DJS19], more precisely, its underlying components. SLAE follows the generic construction FGHF′ [DJS19, KS20], which constructs an AE scheme (via N2) from a symmetric encryption scheme $\textsc{Se}[\mathcal{F}, \mathcal{G}]$ and a MAC $\textsc{Mac}[\mathcal{H}, \mathcal{F}']$ (cf. Figure 2), which in turn are composed of functions $\mathcal{F}$ and $\mathcal{F}'$, a PRG $\mathcal{G}$, and a hash function $\mathcal{H}$. The AE scheme SLAE is obtained by instantiating $\mathcal{F}/\mathcal{F}'$, $\mathcal{G}$, and $\mathcal{H}$ with the sponge-based primitives SLFUNC, SPRG, and SVHASH, respectively. The primitives SLFUNC, SPRG, and SVHASH are illustrated in Figure 3, while their pseudocodes are given in Figure 16. The pseudocodes of the composed encryption scheme SLENC and the MAC SLMAC are provided in Figure 15.

3 Committing Security and Generic Composition

In this section, we analyze the committing security of the different generic composition paradigms. In Section 3.1, we analyze the three N-schemes N1, N2, and N3, given in [NRS14]. We continue with the generic encryption scheme $\textsc{Se}[\mathcal{F}, \mathcal{G}]$ and MAC $\textsc{Mac}[\mathcal{H}, \mathcal{F}']$ [DJS19, KS20] in Section 3.2, and its sponge-based instantiation in Section 3.3.

3.1 Committing Security of the N-Schemes

In this section, we analyze the committing security of N1, N2, and N3. We focus on the strongest form of committing security, which we define below. It asks the adversary to output two different contexts and messages that encrypt to the same ciphertext.

Definition 1. Let $\mathrm{AE} = (\mathtt{Enc}, \mathtt{Dec})$ be an authenticated encryption scheme and the game CMT be defined as in Figure 4. For any adversary $\mathcal{A}$, its CMT advantage is defined as

$$\mathbf{Adv}^{\mathsf{CMT}}_{\mathrm{AE}}(\mathcal{A}) := \Pr[\mathsf{CMT}(\mathcal{A}) \to 1] \,.$$

3.1.1 N2 (Encrypt-then-MAC)

The following theorem shows that the N2 construction does not achieve any committing security. Due to the fact that the N2 construction authenticates the ciphertext—compared to the N1 and N3 construction which both authenticate the message—there is a generic attack exploiting the fact that colliding ciphertexts are easy to find for symmetric encryption schemes: We obtain two messages by decrypting an arbitrary ciphertext under different

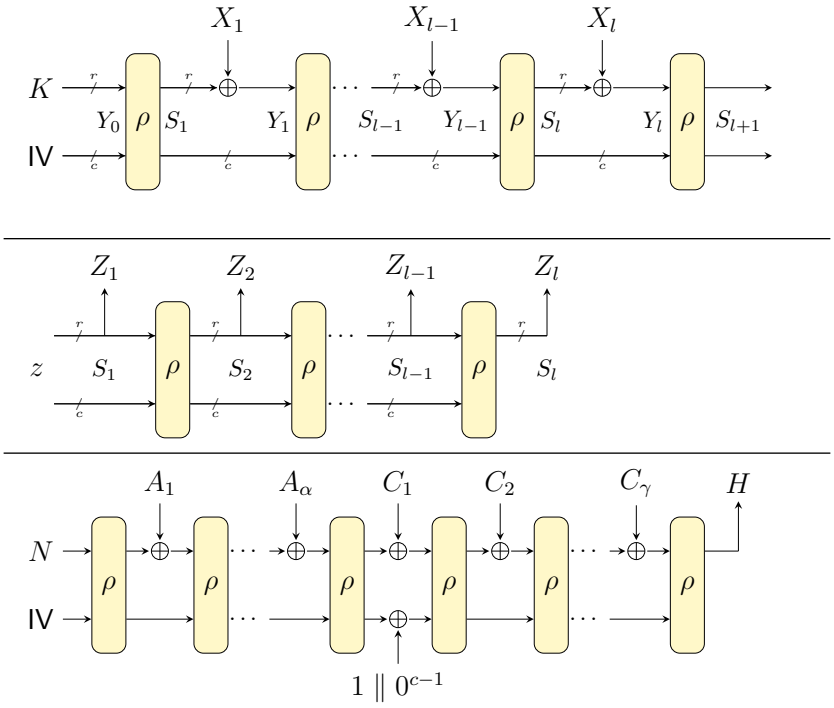

Figure 3: Illustration of the sponge-based primitives SLFUNC, SPRG, and SVHASH.

Game CMT

$(K, N, A, M), (\overline{K}, \overline{N}, \overline{A}, \overline{M}) \leftarrow \mathcal{A}()$
if $(K, N, A) = (\overline{K}, \overline{N}, \overline{A})$
  return 0
$(C, T) \leftarrow \mathtt{Enc}(K, N, A, M)$
$(\overline{C}, \overline{T}) \leftarrow \mathtt{Enc}(\overline{K}, \overline{N}, \overline{A}, \overline{M})$
return $((C, T) = (\overline{C}, \overline{T}))$

Figure 4: Security game CMT for authenticated encryption schemes.

keys and the same nonce, and then authenticate said ciphertext, the nonce, and arbitrary associated data using some key for the MAC.

Theorem 1. *Let* SE *be a symmetric encryption scheme and* MAC *be a MAC. Let further* N2[SE, MAC] *be the authenticated encryption scheme obtained via the N2 construction using* SE *and* MAC*. Then there exists an adversary* $\mathcal{A}$ *such that*

$$\mathbf{Adv}^{\mathsf{CMT}}_{\mathrm{N2[SE,MAC]}}(\mathcal{A}) = 1\,.$$

Proof. We construct a CMT adversary $\mathcal{A}$ against N2[SE, MAC] as shown in Figure 5. It picks an encryption key K_e, a MAC key K_m, a nonce N, associated data A, and a message M at random from the respective sets and computes the ciphertext $C \leftarrow \text{SE.}\mathtt{Enc}(K_e, N, M)$. After sampling a different encryption key $\overline{K}_e$ from $\mathcal{K}\backslash\{K_e\}$ at random, $\mathcal{A}$ computes $\overline{M} \leftarrow \text{SE.}\mathtt{Dec}(\overline{K}_e, N, C)$.[4] Note that $\text{SE.}\mathtt{Enc}(\overline{K}_e, N, \overline{M}) = C$ as we assume the symmetric

[4]Note that the messages M and $\overline{M}$ might be identical which does not affect the attack as the encryption keys are guaranteed to be distinct.

Adversary $\mathcal{A}$

$K_e, N, M \leftarrow\!\!\$\ \mathcal{K} \times \mathcal{N} \times \mathcal{M}$
$C \leftarrow \text{SE.}\mathtt{Enc}(K_e, N, M)$
$\overline{K}_e \leftarrow\!\!\$\ \mathcal{K} \backslash \{K_e\}$
$\overline{M} \leftarrow \text{SE.}\mathtt{Dec}(\overline{K}_e, N, C) \quad /\!\!/$ by tidyness: $C = \text{SE.}\mathtt{Enc}(\overline{K}_e, N, \overline{M})$
$(K_m, A) \leftarrow\!\!\$\ \mathcal{K} \times \mathcal{A}$
return $((K_e, K_m), N, A, M), ((\overline{K}_e, K_m), N, A, \overline{M})$

Figure 5: Our generic adversary breaking committing security for any instantiation of N2.

encryption scheme to be tidy. Lastly the adversary sets $K \leftarrow (K_e, K_m)$ and $\overline{K} \leftarrow (\overline{K}_e, K_m)$, and outputs $(K, N, A, M), (\overline{K}, N, A, \overline{M})$. Then $\mathcal{A}$ is successful as $K_e \neq \overline{K}_e$ implies $K \neq \overline{K}$ and

$$\begin{aligned}
&\text{N2}[\text{SE}, \text{MAC}].\mathtt{Enc}(K, N, A, M) \\
&= \text{SE.}\mathtt{Enc}(K_e, N, M) \parallel \text{MAC.}\mathtt{Tag}(K_m, (N, A, \text{SE.}\mathtt{Enc}(K_e, N, M))) \\
&= C \parallel \text{MAC.}\mathtt{Tag}(K_m, (N, A, C)) \\
&= \text{SE.}\mathtt{Enc}(\overline{K}_e, N, \overline{M}) \parallel \text{MAC.}\mathtt{Tag}(K_m, (N, A, \text{SE.}\mathtt{Enc}(\overline{K}_e, N, \overline{M}))) \\
&= \text{N2}[\text{SE}, \text{MAC}].\mathtt{Enc}(\overline{K}, N, A, \overline{M})\,.
\end{aligned}$$

This finishes the proof. □

3.1.2 N1 (Encrypt-and-MAC)

We first define two new security notions which enable us to show committing security of N1: CR and wCR. The former requires the adversary to find a collision of a MAC, i.e., two tuples of keys and inputs that differ yet result in the same tag. The latter is similar but defined for symmetric encryption and of a more restrictive nature. Instead of asking the adversary to find any collision, it is required to find *distinct* keys and *identical* nonce-message pairs that result in the same ciphertext. While there are no tidy encryption schemes that satisfy unrestricted collision resistance—as our attack against N2 illustrates—it turns out that this restricted form is achievable as we will show later.

Definition 2. Let $\text{MAC} = (\mathtt{Tag}, \mathtt{Ver})$ be a message authentication code and the game CR be defined as in Figure 6. For any adversary $\mathcal{A}$, its CR advantage is defined as

$$\mathbf{Adv}^{\mathsf{CR}}_{\text{MAC}}(\mathcal{A}) \coloneqq \Pr[\mathsf{CR}(\mathcal{A}) \to 1]\,.$$

Definition 3. Let $\text{SE} = (\mathtt{Enc}, \mathtt{Dec})$ be a symmetric encryption scheme and the game wCR be defined as in Figure 6. For any adversary $\mathcal{A}$, its wCR advantage is defined as

$$\mathbf{Adv}^{\mathsf{wCR}}_{\text{SE}}(\mathcal{A}) \coloneqq \Pr[\mathsf{wCR}(\mathcal{A}) \to 1]\,.$$

Note that, while the tag length (in game CR) is fixed, the ciphertext length (in game wCR) is controlled by the adversary through the choice of the messages. Thus, for very short messages the game wCR can be won easily.

The theorem below shows that the N1 construction achieves the strongest form of committing security, i.e., CMT, given that the underlying encryption scheme and MAC achieve our new security notions wCR and CR.

Game CR	Game wCR
$(K, X), (\overline{K}, \overline{X}) \leftarrow \mathcal{A}()$	$(K, N, M), (\overline{K}, \overline{N}, \overline{M}) \leftarrow \mathcal{A}()$
if $(K, X) = (\overline{K}, \overline{X})$	**if** $K = \overline{K} \vee (N, M) \neq (\overline{N}, \overline{M})$
return 0	**return** 0
$T \leftarrow \mathtt{Tag}(K, X)$	$C \leftarrow \mathtt{Enc}(K, N, M)$
$\overline{T} \leftarrow \mathtt{Tag}(\overline{K}, \overline{X})$	$\overline{C} \leftarrow \mathtt{Enc}(\overline{K}, \overline{N}, \overline{M})$
return $(T = \overline{T})$	**return** $(C = \overline{C})$

Figure 6: Security game CR for MACs and wCR for symmetric encryption schemes.

Theorem 2. *Let* Se *be a symmetric encryption scheme and* Mac *be a MAC. Let further* N1[Se, Mac] *be the authenticated encryption scheme obtained via the N1 construction using* Se *and* Mac*. Then for any adversary* $\mathcal{A}$ *there exist adversaries* $\mathcal{B}$ *and* $\mathcal{C}$ *such that*

$$\mathbf{Adv}^{\mathsf{CMT}}_{\text{N1[Se,Mac]}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{wCR}}_{\text{Se}}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{CR}}_{\text{Mac}}(\mathcal{C})\,.$$

Proof. Let $\mathcal{A}$ be a CMT adversary against N1[Se, Mac] and denote his output by $((K, N, A, M), (\overline{K}, \overline{N}, \overline{A}, \overline{M}))$, where $K = (K_e, K_m)$ with K_e and K_m being the encryption key and Mac key, respectively. The same for $\overline{K}$ with encryption key $\overline{K}_e$ and MAC key $\overline{K}_m$. Consider E to be the event that $K_e \neq \overline{K}_e$ and $(N, M) = (\overline{N}, \overline{M})$. In the following, we abbreviate N1[Se, Mac] with N1 as Se and Mac are clear from the context.

We construct a wCR adversary $\mathcal{B}$ against Se. It runs $((K, N, A, M), (\overline{K}, \overline{N}, \overline{A}, \overline{M})) \leftarrow \mathcal{A}()$ and outputs $((K_e, N, M), (\overline{K}_e, \overline{N}, \overline{M}))$. If $\mathcal{A}$ is successful, we obtain $(K, N, A, M) \neq (\overline{K}, \overline{N}, \overline{A}, \overline{M})$ and

$$\text{N1}.\mathtt{Enc}(K, N, A, M) = \text{N1}.\mathtt{Enc}(\overline{K}, \overline{N}, \overline{A}, \overline{M})\,. \tag{1}$$

By definition of the N1 construction, the encryption of the scheme N1 is given by

$$\text{N1}.\mathtt{Enc}(K, N, A, M) = (\text{Se}.\mathtt{Enc}(K_e, N, M), \text{Mac}.\mathtt{Tag}(K_m, (N, A, M)))\,. \tag{2}$$

Then we can conclude from Eq. (1) and Eq. (2) that

$$\text{Se}.\mathtt{Enc}(K_e, N, M) = \text{Se}.\mathtt{Enc}(\overline{K}_e, \overline{N}, \overline{M})\,.$$

Next assume that, additionally to $\mathcal{A}$ being successful, event E holds. Then adversary $\mathcal{B}$ succeeds as $K_e \neq \overline{K}_e$ and $(N, M) = (\overline{N}, \overline{M})$, but their respective encryptions under Se agree. This yields

$$\Pr[\mathsf{E} \wedge \mathsf{CMT}(\mathcal{A}) \to 1] \leq \Pr[\mathsf{wCR}(\mathcal{B}) \to 1]\,.$$

Next, we construct a CR adversary $\mathcal{C}$ against Mac. It runs $((K, N, A, M), (\overline{K}, \overline{N}, \overline{A}, \overline{M})) \leftarrow \mathcal{A}()$ and outputs $((K_m, N, A, M), (\overline{K}_m, \overline{N}, \overline{A}, \overline{M}))$. If $\mathcal{A}$ succeeds, we can conclude from Eq. (1) and Eq. (2) that

$$\text{Mac}.\mathtt{Tag}(K_m, (N, A, M)) = \text{Mac}.\mathtt{Tag}(\overline{K}_m, (\overline{N}, \overline{A}, \overline{M}))\,.$$

Assume that additionally to $\mathcal{A}$ being successful, event $\neg\mathsf{E}$ holds, that means either $K_e = \overline{K}_e$ or $(N, M) \neq (\overline{N}, \overline{M})$. If $(N, M) \neq (\overline{N}, \overline{M})$ holds, then $(K_m, N, A, M), (\overline{K}_m, \overline{N}, \overline{A}, \overline{M})$ are two different inputs for Mac. The same can be deduced if $K_e = \overline{K}_e$, as $\mathcal{A}$ being successful implies that $(K, N, A, M) \neq (\overline{K}, \overline{N}, \overline{A}, \overline{M})$ and since the encryption keys agree, the difference must come from the inputs to Mac. Then adversary $\mathcal{C}$ wins the game wCR as it outputs two different tuples that generate the same tag. Thus we obtain

$$\Pr[\neg\mathsf{E} \wedge \mathsf{CMT}(\mathcal{A}) \to 1] \leq \mathbf{Adv}^{\mathsf{CR}}_{\text{Mac}}(\mathcal{C})\,. \tag{3}$$

Combining these results we obtain

$$\begin{aligned}\mathbf{Adv}^{\mathsf{CMT}}_{\mathrm{N1[SE,MAC]}}(\mathcal{A}) &= \Pr[\mathsf{CMT}(\mathcal{A}) \to 1] \\ &= \Pr[\mathsf{E} \wedge \mathsf{CMT}(\mathcal{A}) \to 1] + \Pr[\neg\mathsf{E} \wedge \mathsf{CMT}(\mathcal{A}) \to 1] \\ &\leq \Pr[\mathsf{wCR}(\mathcal{B}) \to 1] + \Pr[\mathsf{CR}(\mathcal{C}) \to 1] \\ &= \mathbf{Adv}^{\mathsf{wCR}}_{\mathrm{SE}}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{CR}}_{\mathrm{MAC}}(\mathcal{C})\,.\end{aligned}$$

This concludes the proof. □

3.1.3 N3 (MAC-then-Encrypt)

The core feature that differentiates N3 from the other two N-schemes is that the tag is no longer appended to the ciphertext, but instead is encrypted alongside the message. This means—at least in theory—that a valid committing attack against N3 can have $T \neq \overline{T}$ and $C = \overline{C}$ at the same time, which is impossible for both N1 and N2. It turns out that this case is the problematic one for proving security of N3.

We start by considering the case that the underlying encryption scheme and MAC fulfill the security properties wCR and CR security, respectively, which we defined for the security proof of N1. While it is not clear that these security properties are fitting for N3 as well, they seem to be a good starting point—in order to prove classical security, coinciding assumptions are required for N1 and N3.

Under the assumption that the wCR and CR properties hold for SE and MAC, we can show that any CMT attack with $T = \overline{T}$ is ruled out, leaving only attacks for which $T \neq \overline{T}$ holds. This can be checked as follows: Assume for sake of contradiction that $\mathcal{A}$ is an adversary that wins the game CMT and its outputs $((K_e, K_m), N, A, M)$ and $((\overline{K}_e, \overline{K}_m), \overline{N}, \overline{A}, \overline{M})$ fulfill $T = \overline{T}$. If $K_e = \overline{K}_e$ holds, at least one of the inputs to MAC must differ by definition of CMT. This contradicts the fact that MAC is CR-secure. If $K_e \neq \overline{K}_e$ and $(N, M) \neq (\overline{N}, \overline{M})$ holds, we also obtain a contradiction to MAC fulfilling CR security. Lastly, the case that $K_e \neq \overline{K}_e$ and $(N, M) = (\overline{N}, \overline{M})$ cannot occur either, as it would contradict the wCR security of SE.

We can further restrict the scope of possible attacks: As $T \neq \overline{T}$ must hold, SE has two different messages as input (namely $M \parallel T \neq \overline{M} \parallel \overline{T}$). By correctness, this implies that $(K_e, N) \neq (\overline{K}_e, \overline{N})$ must hold, as otherwise $\mathrm{SE}.\mathtt{Dec}(K_e, N, C)$ would have to equal both $M \parallel T$ and $\overline{M} \parallel \overline{T}$. This leaves us with attacks for which $T \neq \overline{T}$, i.e., $(K_m, N, A, M) \neq (\overline{K}_m, \overline{N}, \overline{A}, \overline{M})$ and $(K_e, N) \neq (\overline{K}_e, \overline{N})$, holds. While this is not enough to prove security of N3 in general, the restriction of possible attacks might aid security proofs for certain schemes. One example where this is the case, is N3[SLENC, SLMAC], i.e., SLAE but built following N3 instead if N2: In Section 3.3, we will see that the scheme's underlying components fulfill wCR and CR, hence the above argument can be applied to N3[SLENC, SLMAC]. This implies that any attack must contain different tags while yielding the same ciphertext. By construction of N3[SLENC, SLMAC] the tag is computed as output of the function SLMAC, which is based on a random function. Therefore, it is highly unlikely that a pre-chosen tag can be hit by choosing the inputs fittingly. Furthermore, by construction of SLENC, the key stream consists of several r-bit blocks generated by SPRG on input $\mathrm{SLFUNC}(K_e, N)$. In particular, as the tag is inputted into SLENC as the last message block, one needs to find $(K, N), (\overline{K}, \overline{N})$ such that the corresponding last r-bits blocks $(R, \overline{R})$ from the key stream satisfy $R \oplus T = \overline{R} \oplus \overline{T}$. However, both the computation of the key streams R, $\overline{R}$ and the tags T, $\overline{T}$ depend on the respective nonce and message. Thus after either the tag or the key stream is computed, the choices for the remaining one are heavily restricted and hence $R \oplus T = \overline{R} \oplus \overline{T}$ is unlikely to occur.

For the rest of this analysis we drop the assumptions wCR and CR and play through a number of general attack strategies.

1. A naive strategy in CMT attacks is to generate the target ciphertext C from randomly sampled (K, N, A, M) and then look for a second different input tuple $(\overline{K}, \overline{N}, \overline{A}, \overline{M})$ that encrypts to C as well. This approach is what [MLGR23] introduced as context discovery and several of the attacks against the NIST LWC finalists are of this form [KSW23]. Below, we show that context discovery attacks on the AE scheme boil down to similar attacks on the underlying encryption scheme and MAC.

   (a) One possibility for an attack is to invert the ciphertext under $\overline{K}_e$ and $\overline{N}$, which yields $\overline{M} \parallel \overline{T}$. This leaves the adversary with the task of finding $\overline{K}_m$ and $\overline{A}$ such that $\textsc{Mac}(\overline{K}_m, \overline{N}, \overline{A}, \overline{M}) = \overline{T}$. Following the context discovery notions defined for AE schemes [MLGR23], we refer to the above as a (K_m, A)-discovery attack. While there are MACs for which this is not possible (e.g. SLMAC), there are also ones that allow this attack (e.g., the MACs underlying ELEPHANT [BCDM21] and MINALPHER [STA+15] as well as the MAC CHASKEY [MMV+14]).

   (b) Furthermore, one can try to reach the target ciphertext by first computing a second tag $\overline{T}$ using $(\overline{K}_m, \overline{N}, \overline{A}, \overline{M})$. This leaves the task of finding $\overline{K}_e$ such that $\textsc{Se}.\texttt{Enc}(\overline{K}_e, \overline{N}, \overline{M} \parallel \overline{T}) = C$. We call this a K_e-discovery attack. At first glance, this might look like a key recovery attack, where the adversary gets message-ciphertext pairs. Note, however, that the adversary only needs to find a key that satisfies this property for one particular message-ciphertext pair. This differentiates it from a key recovery attack and leaves the possibility for secure scheme that allow for that—though we are not aware of such an example.

2. Another strategy is trying to produce a collision, i.e., instead of fixing a target ciphertext and trying to match it with the second tuple, one varies the inputs of SE simultaneously. As message, nonce, associated data, and MAC key already need to be chosen to compute the tag as input for the encryption scheme, the collision attack boils down to finding K_e and $\overline{K}_e$ such that $\textsc{Se}.\texttt{Enc}(K_e, N, M \parallel T) = \textsc{Se}.\texttt{Enc}(\overline{K}_e, \overline{N}, \overline{M} \parallel \overline{T})$.

Overall, the results for N3 are of a heterogeneous nature: On the one hand, we identify a possible proof strategy for schemes with wCR and CR security, on the other hand we discuss a number of attacks. This indicates that—at least until further analysis is carried out—an individual treatment is necessary for each scheme. However, the above results might give a first indication to whether the scheme under consideration is committing.

3.2 Committing Security for Symmetric Encryption and MACs

Having established the positive results for N1, the question is whether there are any schemes that satisfy the required security notions wCR and CR. In this section we analyze the generic constructions $\textsc{Se}[\mathcal{F}, \mathcal{G}]$ and $\textsc{Mac}[\mathcal{H}, \mathcal{F}']$. We show that the wCR security of $\textsc{Se}[\mathcal{F}, \mathcal{G}]$ reduces to its underlying components $\mathcal{F}$ and $\mathcal{G}$. Likewise, CR security of $\textsc{Mac}[\mathcal{H}, \mathcal{F}']$ reduces to $\mathcal{H}$ and $\mathcal{F}'$. We first give the security notions required in this section, starting with the definition of collision resistance of a hash function and a PRG.

Definition 4. Let $\mathcal{H}\colon \{0,1\}^* \to \{0,1\}^w$ be a hash function with output length w. For any adversary $\mathcal{A}$, its CR advantage is defined as

$$\mathbf{Adv}^{\mathsf{CR}}_{\mathcal{H}}(\mathcal{A}) \coloneqq \Pr[\mathcal{H}(X_1) = \mathcal{H}(X_2) \wedge X_1 \neq X_2 \mid (X_1, X_2) \leftarrow \mathcal{A}()]\,.$$

Definition 5. Let $\mathcal{G}\colon \{0,1\}^\sigma \to \{0,1\}^*$ be a pseudorandom generator with associated seed space $\{0,1\}^\sigma$. For any adversary $\mathcal{A}$, its CR advantage is defined as

$$\mathbf{Adv}^{\mathsf{CR}}_{\mathcal{G}}(\mathcal{A}) \coloneqq \Pr[\mathcal{G}(z_1) = \mathcal{G}(z_2) \wedge z_1 \neq z_2 \mid (z_1, z_2) \leftarrow \mathcal{A}()]\,.$$

Game pbind	Game bind	Game wbind
$(K, X), (\overline{K}, \overline{X}) \leftarrow \mathcal{A}()$	$(K, X), (\overline{K}, \overline{X}) \leftarrow \mathcal{A}()$	$(K, X), (\overline{K}, \overline{X}) \leftarrow \mathcal{A}()$
if $(K, X) = (\overline{K}, \overline{X})$	**if** $(K, X) = (\overline{K}, \overline{X})$	**if** $K = \overline{K} \vee X \neq \overline{X}$
return 0	**return** 0	**return** 0
$Y \leftarrow \mathrm{F}(K, X)$	$Y \leftarrow \mathrm{F}(K, X)$	$Y \leftarrow \mathrm{F}(K, X)$
$\overline{Y} \leftarrow \mathrm{F}(\overline{K}, \overline{X})$	$\overline{Y} \leftarrow \mathrm{F}(\overline{K}, \overline{X})$	$\overline{Y} \leftarrow \mathrm{F}(\overline{K}, \overline{X})$
return $(\lceil Y \rceil_k = \lceil \overline{Y} \rceil_k)$	**return** $(Y = \overline{Y})$	**return** $(Y = \overline{Y})$

Figure 7: Security games pbind, bind, and wbind.

Remark 1. An injective pseudorandom generator trivially is collision-resistant as no collision exists. Unlike hash functions, injectivity seems likely for PRGs as they map a short seed into a larger pseudorandom bit string. However, collision resistance is not implied by a secure PRG: take any secure PRG and hardcode two seeds $z_1 \neq z_2$ to a fixed output Z.

Next, we define the so-called binding security of a keyed function. More precisely, we give three variants of it: pbind, bind, and wbind. The first, pbind, is taken from [BH22]—note that they call this notion bind since they do not make the same distinction as we do—and requires the adversary to find two key-input pairs for which the outputs "partially" agree, i.e., on their first k bits. The second, bind, is the the same except that the whole outputs have to agree. The third, wbind, bears similarities with wCR: the adversary needs to find *distinct* key and *identical* inputs for which the outputs agree.

Definition 6. Let $\mathrm{F} \colon \mathcal{K} \times \{0,1\}^x \to \{0,1\}^y$ be a function and the games pbind, bind, wbind be defined as in Figure 7. For any adversary $\mathcal{A}$, its pbind, bind, and wbind advantages are defined as

$$\mathbf{Adv}^{\mathsf{X}}_{\mathrm{F}}(\mathcal{A}) \coloneqq \Pr[\mathsf{X}(\mathcal{A}) \to 1]\,,$$

where $\mathsf{X} \in \{\mathsf{pbind}, \mathsf{bind}, \mathsf{wbind}\}$.

These security notions can be ordered hierarchically with respect to their strength, namely $\mathbf{Adv}^{\mathsf{wbind}}_{\mathrm{F}}(\mathcal{C}) \leq \mathbf{Adv}^{\mathsf{bind}}_{\mathrm{F}}(\mathcal{D}) \leq \mathbf{Adv}^{\mathsf{pbind}}_{\mathrm{F}}(\mathcal{B})$. Details are given in Appendix B.6.

The theorem below shows that wCR security of the encryption scheme $\textsc{Se}[\mathcal{F}, \mathcal{G}]$ reduces to binding security and collision resistance of $\mathcal{F}$ and $\mathcal{G}$, respectively. The details are given in Appendix B.1.

Theorem 3. *Let $\mathcal{F} \colon \{0,1\}^\kappa \times \{0,1\}^\nu \to \{0,1\}^\sigma$ be a function family and $\mathcal{G} \colon \{0,1\}^\sigma \to \{0,1\}^*$ be a pseudorandom generator. Then for any* wCR *adversary $\mathcal{A}$ against $\textsc{Se}[\mathcal{F}, \mathcal{G}]$, there exists a* wbind *adversary $\mathcal{B}$ against $\mathcal{F}$ and a* CR *adversary $\mathcal{C}$ against $\mathcal{G}$ such that*

$$\mathbf{Adv}^{\mathsf{wCR}}_{\textsc{Se}[\mathcal{F},\mathcal{G}]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{wbind}}_{\mathcal{F}}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{CR}}_{\mathcal{G}}(\mathcal{C})\,.$$

The binding security of $\textsc{Mac}[\mathcal{H}, \mathcal{F}']$ follows from the collision resistance of $\mathcal{H}$ and binding security of $\mathcal{F}'$. This is formalized in the following theorem, which is proven in Appendix B.2.

Theorem 4. *Let $\mathcal{H} \colon \{0,1\}^* \to \{0,1\}^w$ be a hash function and $\mathcal{F}' \colon \{0,1\}^\kappa \times \{0,1\}^w \to \{0,1\}^\tau$ be a function family. Then for any* CR *adversary $\mathcal{A}$ against $\textsc{Mac}[\mathcal{H}, \mathcal{F}']$, there exists a* CR *adversary $\mathcal{B}$ against $\mathcal{H}$ and a* bind *adversary $\mathcal{C}$ against $\mathcal{F}'$ such that*

$$\mathbf{Adv}^{\mathsf{CR}}_{\textsc{Mac}[\mathcal{H},\mathcal{F}']}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{CR}}_{\mathcal{H}}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{bind}}_{\mathcal{F}'}(\mathcal{C})\,.$$

The following composition theorem shows committing security of the N1 construction, when instantiating it with $\textsc{Se}[\mathcal{F}, \mathcal{G}]$ and $\textsc{Mac}[\mathcal{H}, \mathcal{F}']$ which in turn are instantiated with $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$, and $\mathcal{F}'$. It follows by combining the previous results and is proven in Appendix B.3.

Theorem 5. *Let* $\mathcal{F}\colon \{0,1\}^\kappa \times \{0,1\}^\nu \to \{0,1\}^\sigma$, $\mathcal{F}'\colon \{0,1\}^\kappa \times \{0,1\}^w \to \{0,1\}^\tau$ *be function families,* $\mathcal{G}\colon \{0,1\}^\sigma \to \{0,1\}^*$ *be a pseudorandom generator, and* $\mathcal{H}\colon \{0,1\}^* \to \{0,1\}^w$ *be a hash function. Then for any* CMT *adversary* $\mathcal{A}$ *against* $\mathrm{N1}[\textsc{Se}[\mathcal{F},\mathcal{G}],\textsc{Mac}[\mathcal{H},\mathcal{F}']]$*, there exists a* wbind *adversary* $\mathcal{B}$ *against* $\mathcal{F}$*, a* CR *adversary* $\mathcal{C}$ *against* $\mathcal{G}$*, a* CR *adversary* $\mathcal{D}$ *against* $\mathcal{H}$ *and a* bind *adversary* $\mathcal{E}$ *against* $\mathcal{F}'$ *such that*

$$\mathbf{Adv}^{\mathsf{CMT}}_{\mathrm{N1}[\textsc{Se}[\mathcal{F},\mathcal{G}],\textsc{Mac}[\mathcal{H},\mathcal{F}']]}(\mathcal{A}) \le \mathbf{Adv}^{\mathsf{wbind}}_{\mathcal{F}}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{CR}}_{\mathcal{G}}(\mathcal{C}) + \mathbf{Adv}^{\mathsf{CR}}_{\mathcal{H}}(\mathcal{D}) + \mathbf{Adv}^{\mathsf{bind}}_{\mathcal{F}'}(\mathcal{E})\,.$$

3.3 An Instantiation from Sponges

Having established Theorem 5, we now give concrete bounds for the sponge-based instantiations of $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$, and $\mathcal{F}'$. Recall that n is the size of the sponge state while c is its capacity.

The theorem below bounds the pbind, bind, and wbind advantages of the sponge-based function SLFUNC. Its proof can be found in Appendix B.4.

Theorem 6. *Let* SLFUNC *be the sponge-based function as displayed in Figure 3. Then for any adversary* $\mathcal{A}$ *making* q *queries to* ρ*, it holds that*

$$\mathbf{Adv}^{\mathsf{pbind}}_{\textsc{SlFunc}}(\mathcal{A}) \le \frac{q^2-q}{2^{c+1}} + \frac{q^2-q}{2^{k+1}} \quad \text{and} \quad \mathbf{Adv}^{\mathsf{wbind}}_{\textsc{SlFunc}}(\mathcal{A}) \le \frac{q^2-q}{2^{n+1}}\,.$$

In particular, for $k = n$ *we have*

$$\mathbf{Adv}^{\mathsf{bind}}_{\textsc{SlFunc}}(\mathcal{A}) \le \frac{q^2-q}{2^{c+1}} + \frac{q^2-q}{2^{n+1}}\,.$$

Next, we give a bound on the collision-resistance of SPRG, the proof is given in Appendix B.5.

Theorem 7. *Let* $\textsc{SPrg}\colon \{0,1\}^n \to \{0,1\}^m$ *be the sponge-based pseudorandom generator as displayed in Figure 3. Then for any adversary* $\mathcal{A}$ *making* q *queries to* ρ*, it holds that*

$$\mathbf{Adv}^{\mathsf{CR}}_{\textsc{SPrg}}(\mathcal{A}) \le \frac{q^2-q}{2^{m+1}}\,.$$

Collision resistance of the sponge-based hash function SVHASH has been shown in [DJS19]. We recall this result below.

Theorem 8 ([DJS19])**.** *Let* SVHASH *be the hash function as displayed in Figure 3 with output length* w*. Then for any adversary* $\mathcal{A}$ *making* q *queries to* ρ*, it holds that*

$$\mathbf{Adv}^{\mathsf{CR}}_{\textsc{SvHash}}(\mathcal{A}) \le \frac{q(q-1)}{2^{w+1}} + \frac{q(q+2)}{2^{c-1}}\,.$$

Using Theorem 5 together with Theorem 6, Theorem 7, and Theorem 8, we obtain the committing security of $\mathrm{N1}[\textsc{SlEnc},\textsc{SlMac}]$.

Theorem 9. *Let* SLFUNC, SPRG, *and* SVHASH *be the function family, PRG, and hash function described in Figure 3, respectively. Let further* $\mathrm{N1}[\textsc{SlEnc},\textsc{SlMac}]$ *be the AE scheme constructed via N1 construction from* SLENC *and* SLMAC*, which in turn are constructed from* SLFUNC, SPRG, *and* SVHASH *as described in Figure 15. Then for any* CMT *adversary* $\mathcal{A}$ *against* AE*, making* q *queries to* ρ*, it holds that*

$$\mathbf{Adv}^{\mathsf{CMT}}_{\mathrm{N1}[\textsc{SlEnc},\textsc{SlMac}]}(\mathcal{A}) \le \frac{q^2-q}{2^n} + \frac{q^2-q}{2^{m+1}} + \frac{q^2-q}{2^{w+1}} + \frac{q^2+2q}{2^{c-1}} + \frac{q^2-q}{2^{c+1}}\,.$$

Remark 2. Note that, in order to obtain a reasonable bound, the length of the messages outputted by $\mathcal{A}$ should satisfy $m \ge \min\{w, c\}$. Otherwise, $\frac{q^2-q}{2^{m+1}}$ becomes the dominant term, which can be trivial for very small m. However, this is a non-restrictive requirement, as real-world committing attacks [DGRW18] usually have this property.

4 Committing Security and Leakage Resilience

In this section, we develop a generic transformation that turns an arbitrary AE scheme into an AE scheme that is both leakage-resilient and committing. We start with the required background on leakage-resilient security notions in Section 4.1 followed by the transformation in Section 4.2.

4.1 Leakage Security Notions

For the leakage model, we use [BMOS17] which is based on [DP08], following the "Only Computation Leaks Information" assumption [MR04]. In this model, the adversary obtains challenge oracles that do not leak—representing the goal of the adversary—and leakage oracles that do leak—representing the power of the adversary to obtain side-channel leakage. For the leakage oracles, the adversary can choose a leakage function from some predetermined set of leakage functions. Alongside the output of the functionality that the leakage oracle represents, the adversary receives the evaluation of the leakage function. In case the scheme is composed of different components, the leakage of the composed scheme is the composition of the leakage from the individual components, i.e., for a primitive C composed of A and B, the leakage set of C is $\mathcal{L}_C = \mathcal{L}_A \times \mathcal{L}_B$ for $\mathcal{L}_A$ and $\mathcal{L}_B$ the leakage sets of A and B, respectively. An assumption that we are making is that comparison of values is leak-free, for instance, when a given tag is compared with the correct, recomputed tag. This assumption is made for SLAE and its generic construction FGHF′ [DJS19, KS20]. Methods to achieve this are presented in [DM21].

Our target is LAE security as defined in [BMOS17]. In the notion, the adversary gets four oracles: Two challenge oracles which either implement the real encryption and decryption or their idealized counterparts, i.e., outputting random ciphertext (for the encryption oracle) and rejecting any query (for the decryption oracle). In addition, the adversary gets two leakage oracles, one for encryption and one for decryption, which always implement the real algorithm. The leakage oracle takes a leakage function as additional input whose output models the side-channel leakage that the adversary receives. Classical security for AE schemes is obtained by discarding the leakage oracles for the adversary. The security notion is formalized below.

Definition 7 (LAE Security)**.** Let $\mathrm{AE} = (\mathtt{Enc}, \mathtt{Dec})$ be an authenticated encryption scheme with associated data and the game LAE be as defined in Figure 8. For any nonce-respecting adversary $\mathcal{A}$ that never forwards or repeats queries to or from the oracles Enc and Dec and only makes encryption and decryption queries containing leakage functions in the set $\mathcal{L}_{AE}$, describing the leakage sets for authenticated encryption, its corresponding LAE advantage is given by

$$\mathbf{Adv}^{\mathsf{LAE}}_{\mathrm{AE}}(\mathcal{A}, \mathcal{L}_{AE}) \coloneqq \left|\Pr[\mathcal{A}^{\mathsf{LAE}} \to 1 \mid b = 1] - \Pr[\mathcal{A}^{\mathsf{LAE}} \to 1 \mid b = 0]\right| .$$

Barwell et al. [BMOS17] mention that their LAE notion can be slightly weakened by following the approach for AE security due to Shrimpton and Terashima [ST13]. Here, a ciphertext consists of the actual ciphertext C along with some, so-called, *recovery information* R. One can think of this recovery information as the tag—in case of AEAD schemes via generic composition methods. Shrimpton and Terashima [ST13] require only C to be indistinguishable from random but not R. We adopt this approach and refer to the resulting notion as wLAE (with security game wLAE).

In addition to LAE/wLAE security, we need LPRF security which corresponds to the standard PRF security enhanced with leakage: The adversary gets a challenge oracle, which implements either the real function or a random function. On top of that, the adversary gets a leakage oracle, which always implements the real function that also returns leakage, based on the leakage function queried by the adversary. The definition of LPRF security is given below.

Game LAE	**oracle** $\mathsf{Enc}(N, A, M)$	**oracle** $\mathsf{Dec}(N, A, C)$
$b \leftarrow\!\!\$\ \{0,1\}$	$C \leftarrow \mathtt{Enc}(K, N, A, M)$	**if** $b = 0$
$K \leftarrow\!\!\$\ \mathcal{K}$	**if** $b = 0$	**return** $\perp$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc},\mathsf{LEnc},\mathsf{Dec},\mathsf{LDec}}()$	**return** $\overline{C} \leftarrow\!\!\$\ \{0,1\}^{\lvert C\rvert}$	$M \leftarrow \mathtt{Dec}(K, N, A, C)$
return $(b' = b)$	**return** C	**return** M

oracle $\mathsf{LEnc}(N, A, M, L)$	**oracle** $\mathsf{LDec}(N, A, C, L)$
$\Lambda \leftarrow L(K, N, A, M)$	$\Lambda \leftarrow L(K, N, A, C)$
$C \leftarrow \mathtt{Enc}(K, N, A, M)$	$M \leftarrow \mathtt{Dec}(K, N, A, C)$
return (C, Λ)	**return** (M, Λ)

Figure 8: Security game LAE.

Definition 8 (LPRF Security)**.** Let $\mathrm{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a function family indexed by $\mathcal{K}$ over the domain $\mathcal{X}$ and let the game LPRF be as defined in Figure 9. For any adversary $\mathcal{A}$ that never forwards or repeats queries to or from the oracle F and only queries leakage functions in the set $\mathcal{L}_F$, describing the leakage set for the function, its corresponding LPRF advantage is given by

$$\mathbf{Adv}^{\mathsf{LPRF}}_{\mathrm{F}}(\mathcal{A}, \mathcal{L}_F) \coloneqq \left|\Pr[\mathcal{A}^{\mathsf{LPRF}} \Rightarrow 1 \mid b = 1] - \Pr[\mathcal{A}^{\mathsf{LPRF}} \Rightarrow 1 \mid b = 0]\right|.$$

Game LPRF	**oracle** $\mathsf{F}(X)$	**oracle** $\mathsf{LF}(X, L)$
$b \leftarrow\!\!\$\ \{0,1\}$	**if** $b = 0$	$\Lambda \leftarrow L(K, X)$
$K \leftarrow\!\!\$\ \mathcal{K}$	**return** $Y \leftarrow\!\!\$\ \mathcal{Y}$	$Y \leftarrow \mathrm{F}(K, X)$
$b' \leftarrow \mathcal{A}^{\mathsf{F},\mathsf{LF}}()$	**else**	**return** (Y, Λ)
return $(b' = b)$	**return** $\mathrm{F}(K, X)$	

Figure 9: Security game LPRF.

Finally, we require LUF security as defined in [DJS19]. This notion grants the adversary a challenge oracle Guess, where the adversary tries to predict the output of the function. In addition, $\mathcal{A}$ receives an oracle F which provides the adversary the output of the function for queried inputs and an oracle Lkg which evaluates the function on an input chosen by the adversary but returns *only* the leakage. To avoid trivial wins, the adversary does not win if it predicts the output of an input queried to F, though it can obtain the leakage for an output it later tries to predict (meaning that queries can be repeated across Guess and Lkg). Below we give the formal definition.

Definition 9 (LUF Security)**.** Let $\mathrm{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a function family over the domain $\mathcal{X}$ and indexed by $\mathcal{K}$, and the game LUF be as defined in Figure 10. Then, for any adversary $\mathcal{A}$, querying only leakage functions in the set L_F, its corresponding LUF advantage is given by

$$\mathbf{Adv}^{\mathsf{LUF}}_{\mathrm{F}}(\mathcal{A}, \mathcal{L}_F) \coloneqq \Pr[\mathsf{LUF}(\mathcal{A}) \to 1].$$

4.2 A Generic Transformation

Our results so far show that one can construct committing authenticated encryption via generic composition, more precisely by following the Encrypt-and-MAC paradigm.

Game LUF	**oracle** Guess$(X, \overline{Y})$	**oracle** F(X)	**oracle** Lkg(X, L)
win $\leftarrow 0$	$Y \leftarrow \mathrm{F}(K, X)$	$\mathcal{S} \leftarrow\cup X$	$\Lambda \leftarrow L(K, X)$
$\mathcal{S} \leftarrow \emptyset$	**if** $X \notin \mathcal{S} \wedge Y = \overline{Y}$	$Y \leftarrow \mathrm{F}(K, X)$	**return** Λ
$K \leftarrow\$ \mathcal{K}$	win $\leftarrow 1$	**return** Y	
$\mathcal{A}^{\mathsf{Guess},\mathsf{F},\mathsf{Lkg}}()$	**return** $(Y = \overline{Y})$		
return win			

Figure 10: Security game LUF.

However, our analysis also revealed that the Encrypt-then-MAC approach can never yield committing authenticated encryption. This presents a stark contrast to the results that are known in the realm of leakage-resilient AE schemes via generic composition: Barwell et al. [BMOS17] showed that only the Encrypt-then-MAC paradigm yields leakage-resilient AE schemes whereas the other two approaches suffer from inherent weaknesses. The key problem is that in both Encrypt-and-MAC and MAC-then-Encrypt, adversaries can obtain decryption leakage even for invalid ciphertexts, as the ciphertext needs to be decrypted in order to be validated. In contrast, Encrypt-then-MAC schemes can validate the ciphertext *before* decrypting it, thereby guaranteeing that decryption can leak only for valid ciphertexts.

In total, our results on the committing security of the generic composition methods combined with the existing ones on their leakage-resilience, expose that we cannot build AE schemes with both properties from generic composition. On our quest for committing and leakage-resilient AE schemes, we hence turn towards transformations that achieve the security notions.

To the best of our knowledge, there are two transformations that achieve CMT security: the combination of UtC and HtE from [BH22] and the CTX construction from [CR22]. Other transformations PaddingZeros, KeyHashing, and CAU-C1 [BH22] only achieve weaker forms of committing security.

Both constructions are shown to achieve CMT security while maintaining the security of the underlying AE scheme.[5] Here, however, security is to be understood only with respect to classical AE security. Thus, when applying the transformation to any AE scheme that achieves LAE security, it is unclear whether the resulting scheme still achieves LAE security. Note that both transformations feed the key K as input into a hash function H. If the hash function is not leakage-resilient, the adversary might be able to learn the key which would render the scheme insecure. In particular, we are not aware of any leakage-resilient hash function. While sponge constructions can achieve leakage resilience by reducing the rate to a minimum [DEM+17, DEM+20, DJS19], for hash functions such a change seems impractical.

This raises the question whether there are other transformations without this drawback. In the following we answer this question in the affirmative:

Our transform EtC (for Encrypt-then-Commit), turns an arbitrary AEAD scheme into an AEAD scheme that is both leakage-resilient—in the sense of wLAE—and achieves CMT security. This transformation is shown in Figure 11 and visualized in Figure 12. The session key K' is computed by feeding key and nonce into a keyed function F, while the commitment P stems from hashing nonce, associated data, and ciphertext, and then taking the result and the key as input to F—note that we deploy a domain separation for these two invocations of F. While this transform requires an arbitrary-length ciphertext to be hashed, the ciphertext expansion is the same as for the existing transform UtC*. Lastly, observe that the difference between LAE and wLAE security for EtC lies in the

[5] Note that [BH22] also provides a transformation RtC that maintains nonce-misuse resistance.

$\textsc{EtC}[\mathrm{H},\mathrm{F},\textsc{Ae}].\texttt{Enc}(K,N,A,M)$	$\textsc{EtC}[\mathrm{H},\mathrm{F},\textsc{Ae}].\texttt{Dec}(K,N,A,(P^*,C))$
$K' \leftarrow \mathrm{F}(K, N \parallel 0)$	$P \leftarrow \mathrm{F}(K, \mathrm{H}(N,A,C) \parallel 1)$
$C \leftarrow \textsc{Ae}.\texttt{Enc}(K', N, A, M)$	**if** $P \neq P^*$
$P \leftarrow \mathrm{F}(K, \mathrm{H}(N,A,C) \parallel 1)$	**return** $\perp$
return (P, C)	$K' \leftarrow \mathrm{F}(K, N \parallel 0)$
	$M \leftarrow \textsc{Ae}.\texttt{Dec}(K', N, A, C)$
	return M

Figure 11: Transformation EtC.

commitment P, which is appended to the ciphertext: for LAE, it has to be unpredictable and indistinguishable from random whereas for wLAE it merely has to be unpredictable.

Remark and Errata. The proceedings version of this paper [SW24] contains another transformation called UtC* which is shown in Figure 13. It turns out that this transform—while achieving CMT security—does not yield leakage resilient AE schemes as we claimed in [SW24]. More precisely, the following forgery attack is possible: Let Ae be an AEAD scheme, F be a function, and H be a hash function. Further, let $\textsc{UtC}^*[\mathrm{H},\mathrm{F},\textsc{Ae}]$ be the resulting AEAD scheme after applying the UtC* transform. Consider the set of leakage functions $L = L_{\mathrm{H}} \times L_{\mathrm{F}} \times L_{\textsc{Ae}}$, where $L_{\textsc{Ae}}(K,\cdot,\cdot,\cdot) = K$, i.e., leakage of the AEAD scheme leaks the entire key. For the UtC*-transformed scheme, the leakage function leaks the session key K', which is derived from the context as $(P,K') \leftarrow \mathrm{F}(K,\mathrm{H}(N,A))$. Consider an adversary $\mathcal{A}$ that picks an arbitrary tuple (N,A,M) of nonce, associated data, and message, and queries it to its leakage encryption oracle. The response will be some ciphertext (P,C) and the leakage $\Lambda_{\textsc{Ae}}$ provides the adversary with the session key $K' = \mathrm{F}(K,\mathrm{H}(N,A))$. Using K', $\mathcal{A}$ can pick an arbitrary message $\overline{M}$ (this one needs to be different than the queried message M) and compute $\overline{C} \leftarrow \textsc{Ae}.\texttt{Enc}(K',N,A,\overline{M})$. By outputting $(P,\overline{C})$, $\mathcal{A}$ has successfully forged a ciphertext. The problem of the transform is that an adversary which obtains K' via leakage can forge arbitrary ciphertext for K'. Since K' is determined by the nonce and the associated data, this allows to forge any ciphertext for the context (K,N,A).

The claims regarding confidentiality in the presence of leakage are unaffected. It seems plausible that requiring LAE security of the underlying AEAD scheme might be sufficient as a fix—the aforementioned attack would definitely be ruled out by that. However, the transform would then not turn an arbitrary AEAD scheme into one that is both leakage-resilient *and* committing but rather turn a leakage-resilient one into one that is committing while *maintaining* its leakage resilience. Further, we do not know how to prove it since there are certain query combinations that result in forbidden queries for the reduction. That is why we are pursuing an alternative route, which allows us to provide a transform that does *not* require leakage resilience of the underlying scheme—as was the initial goal. For this, we make two main changes: Firstly, we include the ciphertext in the computation of the commitment P and, secondly, we consider a slightly weakened form of leakage resilience wLAE as described in Section 4.1. These changes allow us to circumvent the forbidden queries and give a proof for our new transform EtC.

The theorem below establishes the security of the EtC transform.

Theorem 10. *Let* Ae *be an authenticated encryption scheme,* $\mathrm{F}\colon \mathcal{K}\times\{0,1\}^x \to \{0,1\}^k \times \mathcal{K}$ *be a function, and* $\mathrm{H}\colon \{0,1\}^* \to \{0,1\}^x$ *be a hash function with associated leakage sets* $\mathcal{L}_{\textsc{Ae}}$, $\mathcal{L}_{\mathrm{F}}^0 \cup \mathcal{L}_{\mathrm{F}}^1$, *and* $\mathcal{L}_{\mathrm{H}}$, *respectively.*[6] *Let further* $\textsc{EtC}[\mathrm{H},\mathrm{F},\textsc{Ae}]$ *be the authenticated encryption scheme resulting from* H, F, *and* Ae *via the transformation shown in Figure 11 with*

[6]Note that $\mathcal{L}_{\mathrm{F}}^0$ and $\mathcal{L}_{\mathrm{F}}^1$ refer to the leakage for the first and second application of F in EtC, respectively.

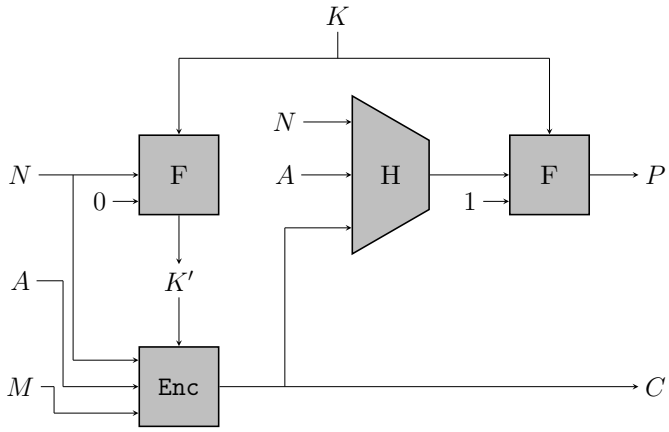

Figure 12: Visualization of the ETC transform.

$\text{UTC}^*[\text{H},\text{F},\text{AE}].\texttt{Enc}(K,N,A,M)$	$\text{UTC}^*[\text{H},\text{F},\text{AE}].\texttt{Dec}(K,N,A,(P^*,C,T))$
$X \leftarrow \text{H}(N,A)$	$X \leftarrow \text{H}(N,A)$
$(P,K') \leftarrow \text{F}(K,X)$	$(P,K') \leftarrow \text{F}(K,X)$
$(C,T) \leftarrow \texttt{Enc}(K',N,A,M)$	**if** $P^* \neq P$
return (P,C,T)	**return** $\bot$
	return $\texttt{Dec}(K',N,A,(C,T))$

Figure 13: Our modified transform UTC*.

associated leakage set $\mathcal{L} = \mathcal{L}_{\text{H}} \times \mathcal{L}_{\text{F}}^0 \times \mathcal{L}_{\text{F}}^1 \times \mathcal{L}_{\text{AE}}$*. Then for any adversary* $\mathcal{A}$ *there exist adversaries* $\mathcal{B}$ *and* $\mathcal{C}$*, such that*

$$\mathbf{Adv}^{\mathsf{CMT}}_{\text{ETC}[\text{H},\text{F},\text{AE}]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{bind}}_{\text{F}}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{CR}}_{\text{H}}(\mathcal{C})\,,$$

and for any nonce-respecting adversary $\mathcal{A}$*, making* q *queries to its challenge oracles, there exist adversaries* $\mathcal{A}_i$ *for* $i \in \{0,\ldots,3\}$*, such that*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{wLAE}}_{\text{ETC}[\text{H},\text{F},\text{AE}]}(\mathcal{A},\mathcal{L}) \leq\ & 2\mathbf{Adv}^{\mathsf{CR}}_{\text{H}}(\mathcal{A}_0) + 2\mathbf{Adv}^{\mathsf{LUF}}_{\text{F}}(\mathcal{A}_1,\mathcal{L}^1_{\text{F}}) \\ & + \mathbf{Adv}^{\mathsf{LPRF}}_{\text{F}}(\mathcal{A}_2,\mathcal{L}^0_{\text{F}}) + q\mathbf{Adv}^{\mathsf{AE}}_{\text{AE}}(\mathcal{A}_3)\,.\end{aligned}$$

Proof. We start by showing the first part of the statement. Let G be the CMT game instantiated with the scheme $\text{ETC}[\text{H},\text{F},\text{AE}]$ that is composed of the hash function H, the function F, and the authenticated encryption scheme AE. Let further $\widetilde{\mathsf{G}}$ be like G, except that $\mathcal{A}$ loses if its outputs constitute a collision in the hash function H. By a game hopping argument, we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{CMT}}_{\text{ETC}[\text{H},\text{F},\text{AE}]}(\mathcal{A}_0) &= \Pr[\mathsf{G}(\mathcal{A}) \to 1] \\ &\leq |\Pr[\mathsf{G}(\mathcal{A}) \to 1] - \Pr[\widetilde{\mathsf{G}}(\mathcal{A}) \to 1]| + \Pr[\widetilde{\mathsf{G}}(\mathcal{A}) \to 1]\,.\end{aligned}$$

The games G and $\widetilde{\mathsf{G}}$ are identical until a collision of H, occurs. We construct $\mathcal{C}$ that runs $\mathcal{A}$ to obtain (K,N,A,M), $(\overline{K},\overline{N},\overline{A},\overline{M})$ and simply outputs (N,A,C), $(\overline{N},\overline{A},\overline{C})$ for $C = \text{AE}.\texttt{Enc}(K,N,A,M)$ and $\overline{C} = \text{AE}.\texttt{Enc}(\overline{K},\overline{N},\overline{A},\overline{M})$. It then holds that

$$|\Pr[\mathsf{G}(\mathcal{A}) \to 1] - \Pr[\widetilde{\mathsf{G}}(\mathcal{A}) \to 1]| \leq \mathbf{Adv}^{\mathsf{CR}}_{\text{H}}(\mathcal{C})\,.$$

Let $\mathcal{A}$ be an adversary winning the game $\widetilde{\mathsf{G}}$. This means that $\mathcal{A}$ outputs $(K, N, A, M) \neq (\overline{K}, \overline{N}, \overline{A}, \overline{M})$, which fulfill

$$\begin{aligned}\textsc{EtC}[\mathrm{H}, \mathrm{F}, \textsc{Ae}].\mathtt{Enc}(K, N, A, M) &= (\mathrm{F}(K, \mathrm{H}(N, A, C)||1), C) \\ &= (\mathrm{F}(\overline{K}, \mathrm{H}(\overline{N}, \overline{A}, \overline{C})||1), \overline{C}) \\ &= \textsc{EtC}[\mathrm{H}, \mathrm{F}, \textsc{Ae}].\mathtt{Enc}(\overline{K}, \overline{N}, \overline{A}, \overline{M})\,.\end{aligned}$$

We construct an adversary $\mathcal{B}$, which runs $\mathcal{A}$—obtaining $(K, N, A, M), (\overline{K}, \overline{N}, \overline{A}, \overline{M})$—and outputs $(K, \mathrm{H}(N, A, C))$ and $(\overline{K}, \mathrm{H}(\overline{N}, \overline{A}, \overline{C}))$. Using the above, we get that $\mathcal{B}$'s output satisfies $\mathrm{F}(K, \mathrm{H}(N, A, C) \parallel 1) = \mathrm{F}(\overline{K}, \mathrm{H}(\overline{N}, \overline{A}, \overline{C} \parallel 1))$. It remains to argue that its output is valid, i.e., $(K, \mathrm{H}(N, A, C) \parallel 1) \neq (\overline{K}, \mathrm{H}(\overline{N}, \overline{A}, \overline{C}) \parallel 1)$. In case that $\mathcal{A}$'s output satisfies $K \neq \overline{K}$, this holds trivially. In the case that $\mathcal{A}$'s output satisfies $K = \overline{K}$ it must hold that $(N, A) \neq (\overline{N}, \overline{A})$ and $\mathrm{H}(N, A, C) \neq \mathrm{H}(\overline{N}, \overline{A}, \overline{C})$, as otherwise $\mathcal{A}$ would not be an adversary winning the game $\widetilde{\mathsf{G}}$. Thus, in both cases, we have that $\mathcal{B}$'s output is valid and conclude with

$$\Pr[\widetilde{\mathsf{G}}(\mathcal{A}) \to 1] = \mathbf{Adv}^{\mathsf{bind}}_{\mathrm{F}}(\mathcal{B})\,.$$

Collecting the above finishes the first part of the proof.

Next, we prove the second part of the statement. The overall proof strategy follows by consecutive game hops, using the following games:

- G_0: game wLAE
- G_1: Dec is replaced by $\perp$ and LDec is modified to reject all queries excepts the ones forwarded from LEnc; this means the output will be $\perp$ together with the leakage of verifying P via F
- G_2: the output K' of F is replaced by random values—note that P is still computed via F as before
- G_3: Enc is changed to output uniform ciphertexts C
- G_4: LDec is changed back to no longer reject non-forwarded queries (note that this corresponds to the ideal version of LAE)

To deal with hash collisions, we consider variants of these games, where the oracles reject queries (outputting $\perp$) if there is a hash collision. We indicate these variants using a tilde, e.g., $\widetilde{\mathsf{G}}_i$. The exact sequence of game hops for the proof is thus

$$\mathsf{G}_0 \to \widetilde{\mathsf{G}}_0 \to \widetilde{\mathsf{G}}_1 \to \widetilde{\mathsf{G}}_2 \to \widetilde{\mathsf{G}}_3 \to \widetilde{\mathsf{G}}_4 \to \mathsf{G}_4$$

meaning we first switch to the version where hash collisions are rejected ($\mathsf{G}_0 \to \widetilde{\mathsf{G}}_0$), then do the game hops according to the description above ($\widetilde{\mathsf{G}}_0 \to \widetilde{\mathsf{G}}_1 \to \widetilde{\mathsf{G}}_2 \to \widetilde{\mathsf{G}}_3 \to \widetilde{\mathsf{G}}_4$), and finally switch back to not reject queries leading to hash collisions ($\widetilde{\mathsf{G}}_4 \to \mathsf{G}_4$). In the following, we analyze the individual game hops.

Game Hop $\mathsf{G}_0 \to \widetilde{\mathsf{G}}_0$:

We start by giving a bound for the first summand. The games G_0 and $\widetilde{\mathsf{G}}_0$ are identical until a collision of H occurs. We construct $\mathcal{A}_0$ that outputs the tuples (N, A, C), $(\overline{N}, \overline{A}, \overline{C})$ by $\mathcal{A}$ that yielded a hash collision. It then holds that

$$|\Pr[\mathcal{A}^{\mathsf{G}_0} \to 1] - \Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_0} \to 1]| \leq \mathbf{Adv}^{\mathsf{CR}}_{\mathrm{H}}(\mathcal{A}_0)\,.$$

Note that in the following we do not describe the check for hash collisions, in which case the adversaries simply reject a query; we thus can assume that the hash values in the next hops are always unique.

Game Hop $\widetilde{\mathsf{G}}_0 \to \widetilde{\mathsf{G}}_1$:

To bound $|\Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_0} \to 1] - \Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_1} \to 1]|$, we construct the following LUF adversary $\mathcal{A}_1$ against F. For queries (N, A, M) to Enc by $\mathcal{A}$, the adversary $\mathcal{A}_1$ queries $N \parallel 0$ to F, resulting in K'. This key is used to compute the ciphertext C locally. Additionally, $\mathrm{H}(N, A, C) \parallel 1$ is queried to F, which returns the commitment P. Then (P, C) is returned to $\mathcal{A}$. For queries $(N, A, M, (L_{\mathrm{H}}, L_{\mathrm{F}}^0, L_{\mathrm{F}}^1, L_{\mathrm{AE}}))$ to LEnc, we repeat these steps with the difference that $N \parallel 0$ and $\mathrm{H}(N, A, C) \parallel 1$ are queried to F *and* Lkg. This latter gives the leakage for the two applications of the PRF (Λ_{F}^0,Λ_{F}^1); the leakage of the AE scheme (Λ_{AE}) and the hash function (Λ_{H}) are computed locally by $\mathcal{A}_1$, which then returns $((P, C), (\Lambda_{\mathrm{H}}, \Lambda_{\mathrm{F}}^0, \Lambda_{\mathrm{F}}^1, \Lambda_{\mathrm{AE}}))$. For queries $(N, A, (P, C))$ to Dec, $\mathcal{A}_1$ queries $(\mathrm{H}(N, A, C) \parallel 1, P)$ to Guess, which outputs either 0 or 1. If the output is 0, $\mathcal{A}_1$ returns $\perp$ to $\mathcal{A}$; if it is 1, $\mathcal{A}_1$ queries $N \parallel 0$ to F, which outputs K'. Using the derived key K', $\mathcal{A}_1$ can compute the AE decryption locally and returns M to $\mathcal{A}$. For queries $(N, A, (P, C), (L_{\mathrm{H}}, L_{\mathrm{F}}, L_{\mathrm{AE}}))$ to LDec, $\mathcal{A}_1$ returns $(\perp, (\perp, \perp, \Lambda_{\mathrm{F}}^1, \perp))$ if the query was not forwarded from LEnc, where Λ_{F}^1 is obtained by querying $\mathrm{H}(N, A, C) \parallel 1$ to Lkg. Otherwise, since the query is forwared from LEnc, $\mathcal{A}$ can look up the leakage for the two evaluations of F and the hash function from the corresponding LEnc. It then computes the leakage of the evaluation of AE together with M locally and returns $(M, (\Lambda_{\mathrm{H}}, \Lambda_{\mathrm{F}}^0, \Lambda_{\mathrm{F}}^1, \Lambda_{\mathrm{AE}}))$ to $\mathcal{A}_1$.

It is left to show, that $\mathcal{A}$ being able to differentiate $\widetilde{\mathsf{G}}_1$ and $\widetilde{\mathsf{G}}_0$ implies that $\mathcal{A}_1$ wins the game LUF. In order to differentiate LDec from $\widetilde{\mathsf{G}}_0$ and $\widetilde{\mathsf{G}}_1$, $\mathcal{A}$ needs to make a query $(N, A, (P, C))$, which is *not* forwarded from LEnc but for which $\mathrm{ETC}.\mathtt{Dec}$ does not return $\perp$—in particular this implies that $\mathrm{F}(K, \mathrm{H}(N, A, C) \parallel 1) = P$ has to hold. For a query of this form, LDec in $\widetilde{\mathsf{G}}_1$ returns $(\perp, (\perp, \perp, \Lambda_{\mathrm{F}}^1, \perp))$, while LDec in $\widetilde{\mathsf{G}}_0$ does not. In particular, $\mathcal{A}_1$ can then win the game LUF by querying $(\mathrm{H}(N, A, C) \parallel 1, P)$ to Guess. Note that $\mathcal{A}_1$ does not win the game LUF if $\mathrm{H}(N, A, C) \parallel 1$ was queried to F before. However, we can exclude this case: Recall that a query of $\mathrm{H}(N, A, C) \parallel 1$ to F could have only happened after a query of $\mathcal{A}$ to LEnc or Enc.

Assume (N, A, M) was queried to Enc or LEnc with output $(\overline{P}, C)$.[7] This means that $\mathrm{H}(N, A, C) \parallel 1$ was queried to F. Note that, since we assume that $N, A, (P, C)$ was not forwarded from $\mathsf{Enc}/\mathsf{LEnc}$, $\overline{P} \neq P$ must hold. However, then $\mathrm{F}(K, \mathrm{H}(N, A, C) \parallel 1) = \overline{P} \neq P$ contradicts our assumption.

Similarly, to distinguish Dec from $\perp$, $\mathcal{A}$ needs to make a query $(N, A, (P, C))$—which was not forwarded from any oracle—and for which $\mathrm{F}(K, \mathrm{H}(N, A, C) \parallel 1) = P$, as otherwise Dec simply returns $\perp$. Just as discussed above for LDec, $\mathcal{A}_1$ then wins game LUF by querying $(\mathrm{H}(N, A, C) \parallel 1, P)$ to Guess.

In total we have shown

$$|\Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_0} \to 1] - \Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_1} \to 1] \leq \mathbf{Adv}_{\mathrm{F}}^{\mathrm{LUF}}(\mathcal{A}_1)\,.$$

Note that for the following hops, we do not describe how queries to Dec are handled, as Dec can easily be simulated by always returning $\perp$.

Game Hop $\widetilde{\mathsf{G}}_1 \to \widetilde{\mathsf{G}}_2$:

To bound $|\Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_1} \to 1] - \Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_2} \to 1]$, we construct the following LPRF adversary $\mathcal{A}_2$ against F. For queries (N, A, M) to Enc by $\mathcal{A}$, the adversary $\mathcal{A}_2$ queries $N \parallel 0$ to F, resulting in a session key K'. With this key, $\mathrm{AE}.\mathtt{Enc}(K', N, A, M)$ is then computed locally and $\mathrm{H}(N, A, C) \parallel 1$ is queried to F, which returns the commitment P. Then, (P, C) is returned to $\mathcal{A}$. Regarding queries to the leakage oracles, we proceed as

[7] We drop the leakage functions and returned leakage here, as our argument is independent of them.

follows: For queries to LEnc, we use LF to obtain the leakage of F applications and compute the leakage of the other components locally. For queries to LDec, we return $(\bot, (\bot, \bot, \Lambda^1_{\mathrm{F}}, \bot))$, where Λ^1_{F} is obtained from LF on input $\mathrm{H}(N, A, C) \parallel 1$, for all queries that are not forwarded from LEnc, and for the ones that are, we proceed as for LEnc, i.e., obtain leakage for F from LF and simulate the rest locally. Whenever $\mathcal{A}$ outputs a bit, $\mathcal{A}_2$ outputs the same.

Note that the LPRF adversary $\mathcal{A}_2$ makes no forbidden queries, due to $\mathcal{A}$ being nonce-respecting and the fact that queries cannot be forwarded between Enc and LDec. Further, $\mathcal{A}_2$ perfectly simulates $\widetilde{\mathsf{G}}_1$ or $\widetilde{\mathsf{G}}_2$ for $\mathcal{A}$—depending on the value of the secret bit in the LPRF game. Since the games only difference is whether F or a random function is used, $\mathcal{A}_2$ wins the game LPRF if $\mathcal{A}$ can distinguish $\widetilde{\mathsf{G}}_1$ and $\widetilde{\mathsf{G}}_2$. Thus, we have shown

$$|\Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_1} \to 1] - \Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_2} \to 1]| = \mathbf{Adv}^{\mathsf{LPRF}}_{\mathrm{H}}(\mathcal{A}_2)\,.$$

Game Hop $\widetilde{\mathsf{G}}_2 \to \widetilde{\mathsf{G}}_3$:

Next, we bound the game hop between G_2 and G_3. Due to the fact that EtC uses different session keys—computed from the context—for the underlying AE scheme Ae, we cannot simply reduce to the security of Ae. To accommodate for that, we do a hybrid argument over the distinct session keys.[8] For this, we define a sequence of hybrid games $\mathsf{H}_0, \ldots, \mathsf{H}_q$, for q the number of nonces that $\mathcal{A}$ queries to Enc,[9] as follows: In H_i, the first i distinct nonces are answered as in $\widetilde{\mathsf{G}}_3$, while the remaining queries are handled as in $\widetilde{\mathsf{G}}_2$. Observe that $\mathsf{H}_0 = \widetilde{\mathsf{G}}_2$ and $\mathsf{H}_q = \widetilde{\mathsf{G}}_3$, which yields

$$\begin{aligned}|\Pr[\mathcal{A}_1^{\widetilde{\mathsf{G}}_2} \to 1] - \Pr[\mathcal{A}_1^{\widetilde{\mathsf{G}}_3} \to 1]| &= |\Pr[\mathcal{A}_1^{\mathsf{H}_0} \to 1] - \Pr[\mathcal{A}_1^{\mathsf{H}_q} \to 1]| \\ &\le \sum_{i=1}^{q} |\Pr[\mathcal{A}_1^{\mathsf{H}_{i-1}} \to 1] - \Pr[\mathcal{A}_1^{\mathsf{H}_{i-1}} \to 1]|\,.\end{aligned}$$

For each $i = 1, \ldots, q$, we construct an AE adversary $\mathcal{D}_i$ to bound the game hop between H_{i-1} and H_i. First, the adversary $\mathcal{D}_i$ samples a key K to locally simulate the oracles LEnc, and LDec for $\mathcal{A}$.

For $(N_1, \ldots, N_q)$ the q distinct nonces, $\mathcal{D}_i$ proceeds as follows: Consider the j-th query to be a query to Enc of the form (N_j, A_j, M_j)—this is wlog as any query to Dec simply returns $\bot$. If $j < i$, $\mathcal{D}_i$ samples C_j randomly, computes $P_j \leftarrow \mathrm{F}_K(\mathrm{H}(N_j, A_j, C_j) \parallel 1)$ and sends it to $\mathcal{A}$. If $j = i$, $\mathcal{D}_i$ invokes its own encryption oracle on (N_i, A_i, M_i) to obtain C_i, which is sent to $\mathcal{A}_1$ together with $P_i \leftarrow \mathrm{F}_K(\mathrm{H}(N, A, C) \parallel 1)$. If $j > i$, K'_j is sampled randomly, $\textsc{Ae}.\mathtt{Enc}(K'_j, N_j, A_j, M_j)$ is computed locally by $\mathcal{D}_i$ as is $P_j \leftarrow \mathrm{F}_K(\mathrm{H}(N, A, C) \parallel 1)$. Then (P_j, C_j) is returned to $\mathcal{A}_1$.

For each query $(N, A, M, L = (L_{\mathrm{H}}, L^0_{\mathrm{F}}, L^1_{\mathrm{F}}, L_{\mathrm{AE}}))$ to LEnc, $\mathcal{D}_i$ computes $K' \leftarrow \mathrm{F}(K, N \parallel 0)$, $C \leftarrow \textsc{Ae}.\mathtt{Enc}(K', N, A, M)$, $P \leftarrow \mathrm{F}(K, \mathrm{H}(N, A, C) \parallel 1)$, and the leakage locally. It sends (P, C) and $(\Lambda_{\mathrm{H}}, \Lambda^0_{\mathrm{F}}, \Lambda^1_{\mathrm{F}}, \Lambda_{\mathrm{AE}})$ to $\mathcal{A}$. Queries to LDec are handled analogously with C being used instead of M,[10] but only if they are forwarded from LEnc—otherwise $\bot$ is returned.

We observe that $\mathcal{D}_i$ simulates H_{i-1} or H_i depending on the secret bit from the AE game, respectively. Note further that $\mathcal{D}_i$ only makes a single encryption query. This follows from the fact that $\mathcal{D}_i$ makes an encryption query for the ith nonce N_i queried by $\mathcal{A}$ which is nonce-respecting and therefore does not repeat this nonce.

[8] In [BH22], the same problem occurs, though they resolve this using multi-user security [BT16].
[9] Note that by the nonce-respecting property, the nonces are distinct.
[10] Note, however, that $P^* = P$ needs to be checked.

Furthermore, we observe that any encryption/decryption query by $\mathcal{D}_i$ stems from a query by $\mathcal{A}$. If any of the queries by $\mathcal{D}_i$ would be prohibited, we would immediately get that $\mathcal{A}$ made queries that are forbidden. Thus, we can conclude

$$\begin{aligned}|\Pr[\mathcal{A}^{\mathsf{H}_{i-1}} \to 1] - \Pr[\mathcal{A}^{\mathsf{H}_{i-1}} \to 1]| &= |\Pr[\mathcal{D}_i^{\mathsf{AE}} \to 1 \mid b = 1] - \Pr[\mathcal{D}_i^{\mathsf{AE}} \to 1 \mid b = 0]| \\ &= \mathbf{Adv}_{\mathrm{AE}}^{\mathsf{AE}}(\mathcal{D}_i)\,.\end{aligned}$$

We define $\mathcal{A}_3$ to be the adversary that picks i from $\{1, \ldots, q\}$ randomly and then acts like $\mathcal{D}_i$. By a standard hybrid argument, we get

$$\begin{aligned}|\Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_2} \to 1] - \Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_3} \to 1]| &\leq \sum_{i=1}^{q} |\Pr[\mathcal{A}^{\mathsf{H}_{i-1}} \to 1] - \Pr[\mathcal{A}^{\mathsf{H}_{i-1}} \to 1]| \\ &\leq q\mathbf{Adv}_{\mathrm{AE}}^{\mathsf{AE}}(\mathcal{A}_3)\,.\end{aligned}$$

Game Hop $\widetilde{\mathsf{G}}_3 \to \widetilde{\mathsf{G}}_4$:

This hop reverts some of the changes done when switching from $\widetilde{\mathsf{G}}_0$ to $\widetilde{\mathsf{G}}_1$. The same argument as above then yields that there exists an adversary $\mathcal{A}_1$ such that

$$|\Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_3} \to 1] - \Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_4} \to 1] \leq \mathbf{Adv}_{\mathrm{F}}^{\mathrm{LUF}}(\mathcal{A}_1)\,.$$

Game Hop $\widetilde{\mathsf{G}}_4 \to \mathsf{G}_4$:

Note that adversary $\mathcal{A}$ can not distinguish $\widetilde{\mathsf{G}}_4$ and G_4 except if there is a collision of the hash function. Then, the encryption oracle (or the decryption oracle, respectively) of $\widetilde{\mathsf{G}}_4$ outputs $\perp$, whereas for G_4 this is not the case. Then we can construct $\mathcal{A}_0$ that outputs the tuples (N, A, C), $(\overline{N}, \overline{A}, \overline{C})$ by $\mathcal{A}$ that yielded a hash collision. Hence, we obtain

$$|\Pr[\mathcal{A}^{\widetilde{\mathsf{G}}_4} \to 1] - \Pr[\mathcal{A}^{\mathsf{G}_4} \to 1]| \leq \mathbf{Adv}_{\mathrm{H}}^{\mathsf{CR}}(\mathcal{A}_0)\,.$$

Collecting everything concludes the proof. □

Remark 3. Note that the proof above does not rely on the unforgeability of the underlying AEAD scheme. It seems plausible that the transform might also work when the underlying component is just a symmetric encryption scheme and unforgeability then follows from the added function F. However, the details would still need to be checked.

5 Conclusion and Instantiation/Implementation Aspects

Our analysis reveals that the generic composition paradigms are not suitable for constructing AE schemes that are simultaneously committing and leakage-resilient. However, we identify another way to obtain such schemes by means of a generic transformation. The latter can be applied to any secure AE scheme, i.e., it does not require any committing or leakage resilience guarantees on the scheme to begin with. In particular, this allows to apply the EtC transformation to AE schemes built from generic composition without needing the underlying symmetric encryption scheme and MAC to fulfill binding or leakage resilience properties. While we just require standard AE security for the scheme that is to be transformed, the transformation uses a function F that needs to be binding, an LPRF, and an LUF. A possible instantiation for this is the sponge-based function SLFunc (cf. Figure 3) which was shown to be an LPRF and an LUF in [DJS19] and binding in Theorem 6. Note that, in order to obtain a good bound on the LPRF/LUF security, the rate r needs to be very small and is typically set to 1. Further, for any practical instantiation,

one needs to choose a concrete non-invertible permutation for ρ. Following [DJS19], a candidate can be obtained using KECCAK-P: define $\rho(x) = \text{KECCAK-P}(x) \oplus x$, where the additional XOR ensures non-invertibility. An alternative candidate is the tagging algorithm of the leakage-resilient MAC given in [BMOS17][11] which is an LPRF but has yet to be analyzed with respect to LUF and binding security. Bellare and Hoang [BH22] provide a construction of a binding function which they call "Counter-then-XOR" (CTX). The CTX construction, which relies on a block-cipher, is also an alternative but not yet analyzed in the leakage setting.

Coming back to SLAE, which was used as an example throughout this work, we can now provide a committing and leakage-resilient variant: This is achieved by applying the ETC transformation, using SVHASH and SLFUNC, to SLAE, where $\rho(x) = \text{KECCAK-P}(x) \oplus x$ as described above. While SLAE is leakage-resilient to start with, due to a very small rate in the underlying function SLFUNC, this is not a necessary prerequisite for applying the ETC transformation. Thus, it is even possible to use SLAE with a bigger rate in SLFUNC as input to ETC, while the two additional SLFUNC instances used in the transformation deploy $r = 1$ to ensure the leakage-resilient guarantees.

Finally, recall that we assume the comparison of values to be leak-free. This assumption is crucial for Theorem 10, when the given P^* is compared with the recomputed P. Any implementation of ETC needs to make sure that this assumption is not violated, e.g., by hardening this step with proper counter measures like masking [CJRR99]. Otherwise, the results can be misused, as was the case in [USS+20], which used the FGHF′ construction [DJS19, KS20] but did not take into account the leak-free assumption—this was pointed out in [BMPS21]. Dobraunig and Mennink [DM21] also provide methods on how to achieve this assumption, which are often readily available in the implementation anyway.

References

[ADG+22] Ange Albertini, Thai Duong, Shay Gueron, Stefan Kölbl, Atul Luykx, and Sophie Schmieg. How to abuse and fix authenticated encryption without key commitment. In Kevin R. B. Butler and Kurt Thomas, editors, *USENIX Security 2022*, pages 3291–3308. USENIX Association, August 2022.

[BBC+20] Davide Bellizia, Olivier Bronchain, Gaëtan Cassiers, Vincent Grosso, Chun Guo, Charles Momin, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Mode-level vs. implementation-level physical security in symmetric cryptography - A practical guide through the leakage-resistance jungle. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 369–400. Springer, Cham, August 2020.

[BCDM21] Tim Beyne, Yu Long Chen, Christoph Dobraunig, and Bart Mennink. Elephant. Technical report, National Institute of Standards and Technology, 2021. Available at https://csrc.nist.gov/projects/lightweight-cryptography/finalists.

[Ber23] Francesco Berti. Reconsidering generic composition: The modes A10, A11 and A12 are insecure. In Leonie Simpson and Mir Ali Rezazadeh Baee, editors, *ACISP 23*, volume 13915 of *LNCS*, pages 157–176. Springer, Cham, July 2023.

[BGP+19] Francesco Berti, Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. TEDT: a leakage-resistant AEAD mode. *IACR TCHES*, 2020(1):256–320, 2019.

[11] This MAC is inspired by the one given in [MOSW15], which, while achieving unpredictability, is not pseudorandom in the leakage setting (LPRF). This makes it unsuitable for the UTC* transformation.

[BGPS21] Francesco Berti, Chun Guo, Thomas Peters, and François-Xavier Standaert. Efficient leakage-resilient MACs without idealized assumptions. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part II*, volume 13091 of *LNCS*, pages 95–123. Springer, Cham, December 2021.

[BH22] Mihir Bellare and Viet Tung Hoang. Efficient schemes for committing authenticated encryption. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 845–875. Springer, Cham, May / June 2022.

[BMOS17] Guy Barwell, Daniel P. Martin, Elisabeth Oswald, and Martijn Stam. Authenticated encryption in the face of protocol and side channel leakage. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 693–723. Springer, Cham, December 2017.

[BMPS21] Olivier Bronchain, Charles Momin, Thomas Peters, and François-Xavier Standaert. Improved leakage-resistant authenticated encryption based on hardware AES coprocessors. *IACR TCHES*, 2021(3):641–676, 2021.

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Tatsuaki Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 531–545. Springer, Berlin, Heidelberg, December 2000.

[BPP18] Francesco Berti, Olivier Pereira, and Thomas Peters. Reconsidering generic composition: The tag-then-encrypt case. In Debrup Chakraborty and Tetsu Iwata, editors, *INDOCRYPT 2018*, volume 11356 of *LNCS*, pages 70–90. Springer, Cham, December 2018.

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

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Berlin, Heidelberg, May / June 2006.

[BT16] Mihir Bellare and Björn Tackmann. The multi-user security of authenticated encryption: AES-GCM in TLS 1.3. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 247–276. Springer, Berlin, Heidelberg, August 2016.

[CFGI+23] Yu Long Chen, Antonio Flórez-Gutiérrez, Akiko Inoue, Ryoma Ito, Tetsu Iwata, Kazuhiko Minematsu, Nicky Mouha, Yusuke Naito, Ferdinand Sibleyras, and Yosuke Todo. Key committing security of AEZ and more. In *ToSC 2023*, 2023.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 398–412. Springer, Berlin, Heidelberg, August 1999.

[CR19] John Chan and Phillip Rogaway. Anonymous AE. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part II*, volume 11922 of *LNCS*, pages 183–208. Springer, Cham, December 2019.

[CR22] John Chan and Phillip Rogaway. On committing authenticated-encryption. In Vijayalakshmi Atluri, Roberto Di Pietro, Christian Damsgaard Jensen, and Weizhi Meng, editors, *ESORICS 2022, Part II*, volume 13555 of *LNCS*, pages 275–294. Springer, Cham, September 2022.

[DEM+17] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, and Thomas Unterluggauer. ISAP – towards side-channel secure authenticated encryption. *IACR Trans. Symm. Cryptol.*, 2017(1):80–105, 2017.

[DEM+20] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, Robert Primas, and Thomas Unterluggauer. ISAP v2.0. *IACR Trans. Symm. Cryptol.*, 2020(S1):390–416, 2020.

[DFG23] Jean Paul Degabriele, Marc Fischlin, and Jérôme Govinden. The indifferentiability of the duplex and its practical applications. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part VIII*, volume 14445 of *LNCS*, pages 237–269. Springer, Singapore, December 2023.

[DGRW18] Yevgeniy Dodis, Paul Grubbs, Thomas Ristenpart, and Joanne Woodage. Fast message franking: From invisible salamanders to encryptment. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 155–186. Springer, Cham, August 2018.

[DJS19] Jean Paul Degabriele, Christian Janson, and Patrick Struck. Sponges resist leakage: The case of authenticated encryption. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part II*, volume 11922 of *LNCS*, pages 209–240. Springer, Cham, December 2019.

[DM19] Christoph Dobraunig and Bart Mennink. Leakage resilience of the duplex construction. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 225–255. Springer, Cham, December 2019.

[DM21] Christoph Dobraunig and Bart Mennink. Leakage resilient value comparison with application to message authentication. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 377–407. Springer, Cham, October 2021.

[DMVA23] Joan Daemen, Silvia Mella, and Gilles Van Assche. Committing authenticated encryption based on SHAKE. *IACR Cryptol. ePrint Arch.*, 2023:1494, 2023.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th FOCS*, pages 293–302. IEEE Computer Society Press, October 2008.

[FKOS22] Sebastian Faust, Juliane Krämer, Maximilian Orlt, and Patrick Struck. On the related-key attack security of authenticated encryption schemes. In Clemente Galdi and Stanislaw Jarecki, editors, *SCN 22*, volume 13409 of *LNCS*, pages 362–386. Springer, Cham, September 2022.

[GPPS20] Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Towards low-energy leakage-resistant AE from the duplex sponge. *IACR Trans. Symm. Cryptol.*, 2020(1):6–42, 2020.

[IKM+21] Tetsu Iwata, Mustafa Khairallah, Kazuhiko Minematsu, Thomas Peyrin, and Chun Guo. Romulus. Technical report, National Institute of Standards and Technology, 2021. Available at https://csrc.nist.gov/projects/lightweight-cryptography/finalists.

[KS20] Juliane Krämer and Patrick Struck. Leakage-resilient authenticated encryption from leakage-resilient pseudorandom functions. In Guido Marco Bertoni and Francesco Regazzoni, editors, *COSADE 2020*, volume 12244 of *LNCS*, pages 315–337. Springer, Cham, April 2020.

[KSW23] Juliane Krämer, Patrick Struck, and Maximiliane Weishäupl. Committing AE from sponges: Security analysis of the NIST LWC finalists. Cryptology ePrint Archive, Report 2023/1525, 2023.

[LGR21] Julia Len, Paul Grubbs, and Thomas Ristenpart. Partitioning oracle attacks. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 195–212. USENIX Association, August 2021.

[MLGR23] Sanketh Menda, Julia Len, Paul Grubbs, and Thomas Ristenpart. Context discovery and commitment attacks - how to break CCM, EAX, SIV, and more. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part IV*, volume 14007 of *LNCS*, pages 379–407. Springer, Cham, April 2023.

[MMV+14] Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. Chaskey: An efficient MAC algorithm for 32-bit microcontrollers. In Antoine Joux and Amr M. Youssef, editors, *SAC 2014*, volume 8781 of *LNCS*, pages 306–323. Springer, Cham, August 2014.

[MOSW15] Daniel P. Martin, Elisabeth Oswald, Martijn Stam, and Marcin Wójcik. A leakage resilient MAC. In Jens Groth, editor, *15th IMA International Conference on Cryptography and Coding*, volume 9496 of *LNCS*, pages 295–310. Springer, Cham, December 2015.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 278–296. Springer, Berlin, Heidelberg, February 2004.

[NRS14] Chanathip Namprempre, Phillip Rogaway, and Thomas Shrimpton. Reconsidering generic composition. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 257–274. Springer, Berlin, Heidelberg, May 2014.

[ST13] Thomas Shrimpton and R. Seth Terashima. A modular framework for building variable-input-length tweakable ciphers. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 405–423. Springer, Berlin, Heidelberg, December 2013.

[STA+15] Yu Sasaki, Yosuke Todo, Kazumaro Aoki, Yusuke Naito, Takeshi Sugawara, Yumiko Murakami, Mitsuru Matsui, and Shoichi Hirose. Minalpher v1.1. Technical report, Submission to the CAESAR Competition, 2015. Available at https://competitions.cr.yp.to/round2/minalpherv11.pdf.

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

[USS+20] Florian Unterstein, Marc Schink, Thomas Schamberger, Lars Tebelmann, Manuel Ilg, and Johann Heyszl. Retrofitting leakage resilient authenticated encryption to microcontrollers. *IACR TCHES*, 2020(4):365–388, 2020.

[VCS22] Corentin Verhamme, Gaëtan Cassiers, and François-Xavier Standaert. Analyzing the leakage resistance of the NIST's lightweight crypto competition's finalists. In *CARDIS 2022*, 2022.

A Additional Background

The pseudocode for the N-schemes is given in Figure 14. Figure 15 provides the pseudocode of the sponge-based encryption scheme SLENC and MAC SLMAC in terms of their underlying components SLFUNC, SPRG, and SVHASH, whose pseudocodes are presented in Figure 16.

$\underline{\texttt{Enc}(K, N, A, M) \text{ in N1}}$
$(K_e, K_m) \leftarrow K$
$C \leftarrow \texttt{Enc}^S(K_e, N, M)$
$T \leftarrow \texttt{Tag}(K_m, N, A, M)$
return (C, T)

$\underline{\texttt{Enc}(K, N, A, M) \text{ in N2}}$
$(K_e, K_m) \leftarrow K$
$C_e \leftarrow \texttt{Enc}^S(K_e, N, M)$
$T \leftarrow \texttt{Tag}(K_m, N, A, C)$
return (C, T)

$\underline{\texttt{Enc}(K, N, A, M) \text{ in N3}}$
$(K_e, K_m) \leftarrow K$
$T \leftarrow \texttt{Tag}(K_m, N, A, M)$
$C \leftarrow \texttt{Enc}^S(K_e, N, M \parallel T)$
return C

$\underline{\texttt{Dec}(K, N, A, (C, T)) \text{ in N1}}$
$(K_e, K_m) \leftarrow K$
$M \leftarrow \texttt{Dec}^S(K_e, N, C)$
if $\texttt{Ver}(K_m, N, A, M, T) = 0$
  return $\perp$
return M

$\underline{\texttt{Dec}(K, N, A, (C, T)) \text{ in N2}}$
$(K_e, K_m) \leftarrow K$
if $\texttt{Ver}(K_m, N, A, C, T) = 0$
  return $\perp$
$M \leftarrow \texttt{Dec}^S(K_e, N, C)$
return M

$\underline{\texttt{Dec}(K, N, A, C) \text{ in N3}}$
$(K_e, K_m) \leftarrow K$
$M \parallel T \leftarrow \texttt{Dec}^S(K_e, N, C)$
if $\texttt{Ver}(K_m, N, A, M, T) = 0$
  return $\perp$
return M

Figure 14: Pseudocode of N1 (left), N2 (middle), and N3 (right) in terms of a symmetric encryption scheme $(\texttt{Enc}^S, \texttt{Dec}^S)$ and a MAC $(\texttt{Tag}, \texttt{Ver})$.

B Full Proofs

B.1 Proof of Theorem 3

Proof. Let $\mathcal{A}$ be a wCR adversary against $\textsc{Se}[\mathcal{F}, \mathcal{G}]$ with output $((K, N, M), (\overline{K}, \overline{N}, \overline{M}))$. We define E to be the event that $\mathcal{F}(K, N) = \mathcal{F}(\overline{K}, \overline{N})$.

Firstly, we construct a wbind adversary $\mathcal{B}$ against $\mathcal{F}$. It runs $((K, N, M), (\overline{K}, \overline{N}, \overline{M})) \leftarrow \mathcal{A}()$ and outputs $((K, N), (\overline{K}, \overline{N}))$. If $\mathcal{A}$ is successful, we have both $K \neq \overline{K}$ and $(N, M) = (\overline{N}, \overline{M})$. Next assume that additionally to $\mathcal{A}$ being successful, event E holds, that means $\mathcal{F}(K, N) = \mathcal{F}(\overline{K}, \overline{N})$. Then adversary $\mathcal{B}$ is successful in game wbind, as $K \neq \overline{K}$ and $N = \overline{N}$, but they map to the same element under $\mathcal{F}$. Hence we have shown

$$\Pr[\mathsf{E} \wedge \mathsf{wCR}(\mathcal{A}) \to 1] \leq \Pr[\mathsf{wbind}(\mathcal{B}) \to 1]\,.$$

Secondly, we construct a CR adversary $\mathcal{B}$ against $\mathcal{G}$. It runs $((K, N, M), (\overline{K}, \overline{N}, \overline{M})) \leftarrow \mathcal{A}()$ and outputs $(\mathcal{F}(K, N), \mathcal{F}(\overline{K}, \overline{N}))$. If $\mathcal{A}$ is successful, then $K \neq \overline{K}$ and $(N, M) = (\overline{N}, \overline{M})$, and $\textsc{Se}[\mathcal{F}, \mathcal{G}].\texttt{Enc}(K, N, M) = \textsc{Se}[\mathcal{F}, \mathcal{G}].\texttt{Enc}(\overline{K}, \overline{N}, \overline{M})$. By definition of $\textsc{Se}[\mathcal{F}, \mathcal{G}]$ the latter is equivalent to $\mathcal{G}(\mathcal{F}(K, N)) \oplus M = \mathcal{G}(\mathcal{F}(\overline{K}, \overline{N})) \oplus \overline{M}$. As $M = \overline{M}$, this implies that $\mathcal{G}(\mathcal{F}(K, N)) = \mathcal{G}(\mathcal{F}(\overline{K}, \overline{N}))$. Next assume that additionally to $\mathcal{A}$ being successful, event $\neg\mathsf{E}$ holds, that means $\mathcal{F}(K, N) \neq \mathcal{F}(\overline{K}, \overline{N})$. Then adversary $\mathcal{C}$ is successful, as he has found a collision for $\mathcal{G}$. Thus we obtain

$$\Pr[\neg\mathsf{E} \wedge \mathsf{wCR}(\mathcal{A}) \to 1] \leq \Pr[\mathsf{CR}(\mathcal{C}) \to 1]\,.$$

$\underline{\textsc{SlEnc-Enc}(K_e, N, M)}$

$z \leftarrow \textsc{SlFunc}(K_e, N)$
$C \leftarrow \textsc{SPrg}(z, |M|) \oplus M$
return C

$\underline{\textsc{SlEnc-Dec}(K_e, N, C)}$

$z \leftarrow \textsc{SlFunc}(K_e, N)$
$M \leftarrow \textsc{SPrg}(z, |C|) \oplus C$
return M

$\underline{\textsc{SlMac-Tag}(K_m, N, A, C)}$

$H \leftarrow \textsc{SvHash}(N, A, C)$
$T \leftarrow \textsc{SlFunc}(K_m, H)$
return $\lceil T \rceil_\tau$

$\underline{\textsc{SlMac-Ver}(K_m, N, A, C, T)}$

$H \leftarrow \textsc{SvHash}(N, A, C)$
$\overline{T} \leftarrow \textsc{SlFunc}(K_m, H)$
if $\overline{T} \neq T$
  return $\bot$
return $\top$

Figure 15: Pseudocode of the encryption scheme $\textsc{SlEnc} = (\textsc{SlEnc-Enc}, \textsc{SlEnc-Dec})$ and the MAC $\textsc{SlMac} = (\textsc{SlMac-Tag}, \textsc{SlMac-Ver})$ in terms of the sponge-based primitives SlFunc, SPrg, and SvHash.

In total, one can conclude

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{wCR}}_{\textsc{Se}[\mathcal{F},\mathcal{G}]}(\mathcal{A}) &= \Pr[\mathsf{wCR}(\mathcal{A}) \to 1] \\
&= \Pr[\mathsf{E} \wedge \mathsf{wCR}(\mathcal{A}) \to 1] + \Pr[\neg\mathsf{E} \wedge \mathsf{wCR}(\mathcal{A}) \to 1] \\
&\leq \Pr[\mathsf{wbind}(\mathcal{B}) \to 1] + \Pr[\mathsf{CR}(\mathcal{C}) \to 1] \\
&= \mathbf{Adv}^{\mathsf{wbind}}_{\mathcal{F}}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{CR}}_{\mathcal{G}}(\mathcal{C})\,,
\end{aligned}$$

which finishes the proof. □

B.2 Proof of Theorem 4

Proof. Let $\mathcal{A}$ be a CR adversary against $\textsc{Mac}[\mathcal{H}, \mathcal{F}']$ with output $((K, X), (\overline{K}, \overline{X}))$ and let E be the event that $K = \overline{K}$ and $\mathcal{H}(X) = \mathcal{H}(\overline{X})$.

Firstly, we construct a CR adversary $\mathcal{B}$ against $\mathcal{H}$. It runs $((K, X), (\overline{K}, \overline{X})) \leftarrow \mathcal{A}()$ and outputs $(X, \overline{X})$. If $\mathcal{A}$ is successful, we know in particular that $(K, X) \neq (\overline{K}, \overline{X})$. Next assume that additionally to $\mathcal{A}$ being successful, event E holds, that means $K = \overline{K}$ and $\mathcal{H}(X) = \mathcal{H}(\overline{X})$. Thus adversary $\mathcal{B}$ is successful as $X, \overline{X}$ are different but map to the same element under $\mathcal{H}$. Hence we have shown

$$\Pr[\mathsf{E} \wedge \mathsf{CR}(\mathcal{A}) \to 1] \leq \Pr[\mathsf{CR}(\mathcal{B}) \to 1]\,.$$

Secondly, we construct a bind adversary $\mathcal{C}$ against $\mathcal{F}'$. It runs $((K, X), (\overline{K}, \overline{X})) \leftarrow \mathcal{A}()$ and outputs $((K, \mathcal{H}(X)), (\overline{K}, \mathcal{H}(\overline{X})))$. If $\mathcal{A}$ is successful, then $(K, X) \neq (\overline{K}, \overline{X})$ and $\textsc{Mac}[\mathcal{H}, \mathcal{F}'].\mathtt{Tag}(K, X) = \textsc{Mac}[\mathcal{H}, \mathcal{F}'].\mathtt{Tag}(\overline{K}, \overline{X})$. By definition of $\textsc{Mac}[\mathcal{H}, \mathcal{F}']$ the latter is equivalent to $\mathcal{F}'(K, \mathcal{H}(X)) = \mathcal{F}'(\overline{K}, \mathcal{H}(\overline{X}))$. If additionally to $\mathcal{A}$ being successful, event $\neg\mathsf{E}$ holds, we know that either $K \neq \overline{K}$ or $\mathcal{H}(X) \neq \mathcal{H}(\overline{X})$. Therefore adversary $\mathcal{C}$ is successful, because $(K, \mathcal{H}(X)) \neq (\overline{K}, \mathcal{H}(\overline{X}))$ and $\mathcal{F}'(K, \mathcal{H}(X)) = \mathcal{F}'(\overline{K}, \mathcal{H}(\overline{X}))$. This yields

$$\Pr[\neg\mathsf{E} \wedge \mathsf{CR}(\mathcal{A}) \to 1] \leq \Pr[\mathsf{bind}(\mathcal{C}) \to 1]$$

$\textsc{SlFunc}(K, X)$	$\textsc{SPrg}(z, L)$	$\textsc{SvHash}(N, A, C)$
$X_1, \ldots, X_l \xleftarrow{r} X$	$l \leftarrow \lceil \frac{L}{r} \rceil$	$A_1, \ldots, A_\alpha \xleftarrow{r} \mathtt{pad}_{10^*}(A, r)$
$Y \leftarrow K \parallel \mathsf{IV}$	$Z \leftarrow \varepsilon$	$C_1, \ldots, C_\gamma \xleftarrow{r} \mathtt{pad}_{10^*}(C, r)$
$S \leftarrow \rho(Y)$	$S \leftarrow z$	$Y \leftarrow N \parallel \mathsf{IV}$
for $i = 1, \ldots, l$	$Z \leftarrow Z \parallel \lceil S \rceil_r$	$S \leftarrow \rho(Y)$
$\quad Y \leftarrow (\lceil S \rceil_r \oplus X_i) \parallel \lfloor S \rfloor_c$	**for** $i = 1, \ldots, l-1$	**for** $i = 1, \ldots, \alpha$
$\quad S \leftarrow \rho(Y)$	$\quad S \leftarrow \rho(S)$	$\quad Y \leftarrow (\lceil S \rceil_r \oplus A_i) \parallel \lfloor S \rfloor_c$
return S	$\quad Z \leftarrow Z \parallel \lceil S \rceil_r$	$\quad S \leftarrow \rho(Y)$
	return $\lceil Z \rceil_L$	$S \leftarrow \lceil S \rceil_r \parallel (\lfloor S \rfloor_c \oplus (1 \parallel 0^{c-1}))$
		for $i = 1, \ldots, \gamma$
		$\quad Y \leftarrow (\lceil S \rceil_r \oplus C_i) \parallel \lfloor S \rfloor_c$
		$\quad S \leftarrow \rho(Y)$
		$H \leftarrow \lceil S \rceil_w$
		return H

Figure 16: Pseudocode of the sponge-based function $\textsc{SlFunc}$, pseudorandom generator $\textsc{SPrg}$, and hash function $\textsc{SvHash}$, constructed from a random transformation $\rho\colon \{0,1\}^n \to \{0,1\}^n$, where $n = r + c$.

and in total we can conclude

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{CR}}_{\textsc{Mac}[\mathcal{H},\mathcal{F}']}(\mathcal{A}) &= \Pr[\mathsf{CR}(\mathcal{A}) \to 1] \\
&= \Pr[\mathsf{E} \wedge \mathsf{CR}(\mathcal{A}) \to 1] + \Pr[\neg\mathsf{E} \wedge \mathsf{CR}(\mathcal{A}) \to 1] \\
&\leq \Pr[\mathsf{CR}(\mathcal{B}) \to 1] + \Pr[\mathsf{bind}(\mathcal{C}) \to 1] \\
&= \mathbf{Adv}^{\mathsf{CR}}_{\mathcal{H}}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{bind}}_{\mathcal{F}'}(\mathcal{C})\,.
\end{aligned}$$

This finishes the proof. □

B.3 Proof of Theorem 5

Proof. Consider a CMT adversary $\mathcal{A}$ against $\textsc{N1}[\textsc{Se}[\mathcal{F},\mathcal{G}], \textsc{Mac}[\mathcal{H},\mathcal{F}']]$. By Theorem 2, there exist adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ such that

$$\mathbf{Adv}^{\mathsf{CMT}}_{\textsc{N1}[\textsc{Se}[\mathcal{F},\mathcal{G}],\textsc{Mac}[\mathcal{H},\mathcal{F}']]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{wCR}}_{\textsc{Se}[\mathcal{F},\mathcal{G}]}(\mathcal{A}_1) + \mathbf{Adv}^{\mathsf{CR}}_{\textsc{Mac}[\mathcal{H},\mathcal{F}']}(\mathcal{A}_2)\,.$$

Applying Theorem 3 for adversary $\mathcal{A}_1$, then yields adversaries $\mathcal{B}$ and $\mathcal{C}$ such that

$$\mathbf{Adv}^{\mathsf{wCR}}_{\textsc{Se}[\mathcal{F},\mathcal{G}]}(\mathcal{A}_1) \leq \mathbf{Adv}^{\mathsf{wbind}}_{\mathcal{F}}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{CR}}_{\mathcal{G}}(\mathcal{C})\,.$$

Analogously, Theorem 4 for adversary $\mathcal{A}_2$ gives adversaries $\mathcal{D}$ and $\mathcal{E}$ such that

$$\mathbf{Adv}^{\mathsf{CR}}_{\textsc{Mac}[\mathcal{H},\mathcal{F}']}(\mathcal{A}_2) \leq \mathbf{Adv}^{\mathsf{CR}}_{\mathcal{H}}(\mathcal{D}) + \mathbf{Adv}^{\mathsf{bind}}_{\mathcal{F}'}(\mathcal{E})\,.$$

Combining these results shows the claim. □

B.4 Proof of Theorem 6

Proof. Let $\mathcal{A}$ be a pbind adversary against $\textsc{SlFunc}$ with output $((K, N), (\overline{K}, \overline{N}))$. For the proof we will refer to the different states of the sponge as S_i and Y_i as shown in Figure 3 (upper part). We show that the probability for $\mathcal{A}$ to win is bounded above by

the probability that for different inputs, the respective outputs of ρ collide in the first k bits (we call this a k-collision) or in the last c bits (this is referred to as an inner collision). For this we define C to be the event that there are $Y \neq \overline{Y}$ such that $\lceil \rho(Y) \rceil_k = \lceil \rho(\overline{Y}) \rceil_k$ or $\lfloor \rho(Y) \rfloor_c = \lfloor \rho(\overline{Y}) \rfloor_c$ for all states Y and $\overline{Y}$ occurring during the evaluation of the tuples outputted by $\mathcal{A}$. Then we can split up the probability of $\mathcal{A}$ winning the game pbind as follows

$$\Pr[\mathsf{pbind}(\mathcal{A}) \to 1] = \Pr[\mathsf{C} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] + \Pr[\neg\mathsf{C} \wedge \mathsf{pbind}(\mathcal{A}) \to 1]\,. \tag{4}$$

As a next step we show that $\Pr[\neg\mathsf{C} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] = 0$. For this consider E to be the event that $N \neq \overline{N}$ and observe that

$$\begin{aligned}\Pr[\neg\mathsf{C} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] = \Pr[\mathsf{E} \wedge \neg\mathsf{C} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] \\ + \Pr[\neg\mathsf{E} \wedge \neg\mathsf{C} \wedge \mathsf{pbind}(\mathcal{A}) \to 1]\,.\end{aligned} \tag{5}$$

We proceed by showing that both summands in the above equation are zero.

First we show $\Pr[\mathsf{E} \wedge \neg\mathsf{C} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] = 0$. We assume that $\neg\mathsf{C}$ holds, hence we know in particular that for all $i \in \{0, \ldots, l\}$ with $Y_i \neq \overline{Y}_i$ it holds that $\lceil \rho(Y_i) \rceil_k \neq \lceil \rho(\overline{Y}_i) \rceil_k$ and $\lfloor \rho(Y_i) \rfloor_c \neq \lfloor \rho(\overline{Y}_i) \rfloor_c$. As $\mathcal{A}$ is successful, we know that $(K, N) \neq (\overline{K}, \overline{N})$ and $\lceil S_{l+1} \rceil_k = \textsc{SlFunc}(K, N) = \textsc{SlFunc}(\overline{K}, \overline{N}) = \lceil \overline{S}_{l+1} \rceil_k$. Furthermore E holds, which implies that $N \neq \overline{N}$. We distinguish the following two cases:

1. $K \neq \overline{K}$: In this case, we have $Y_0 = K \neq \overline{K} = \overline{Y}_0$, hence we obtain that $\lfloor \rho(Y_0) \rfloor_c \neq \lfloor \rho(\overline{Y}_0) \rfloor_c$ since C holds. This implies that $Y_1 \neq \overline{Y}_1$, because $\lfloor Y_1 \rfloor_c = \lfloor \rho(Y_0) \rfloor_c \neq \lfloor \rho(\overline{Y}_0) \rfloor_c = \lfloor \overline{Y}_1 \rfloor_c$. Repeating this argument for all $i \in \{1, \ldots, l-1\}$ yields $Y_l \neq \overline{Y}_l$.

2. $K = \overline{K}$: In this case, we use the fact that $N \neq \overline{N}$ implies the existence of a smallest $m \in \{1, \ldots, l\}$ with $N_m \neq \overline{N}_m$. Since the keys we start with are the same and $N_i = \overline{N}_i$ holds for all $i < m$, we can deduce that $S_m = \overline{S}_m$. Then $N_m \neq \overline{N}_m$ yields $Y_m \neq \overline{Y}_m$. Therefore we obtain $\lfloor \rho(Y_m) \rfloor_c \neq \lfloor \rho(\overline{Y}_m) \rfloor_c$ since C holds, which implies that $\lfloor Y_{m+1} \rfloor_c = \lfloor \rho(Y_m) \rfloor_c \neq \lfloor \rho(\overline{Y}_m) \rfloor_c = \lfloor \overline{Y}_{m+1} \rfloor_c$. Thus in particular it holds that $Y_{m+1} \neq \overline{Y}_{m+1}$ and repeating this argument for all $i \in \{m+1, \ldots, l-1\}$ shows $Y_l \neq \overline{Y}_l$.

For both cases, we have shown that $Y_l \neq \overline{Y}_l$. Hence, since C holds, we can deduce that $\lceil S_{l+1} \rceil_k = \lceil \rho(Y_l) \rceil_k \neq \lceil \rho(\overline{Y}_l) \rceil_k = \lceil \overline{S}_{l+1} \rceil_k$, which contradicts the assumption that $\mathcal{A}$ is successful. Therefore the probability $\Pr[\mathsf{E} \wedge \neg\mathsf{C} \wedge \mathsf{pbind}(\mathcal{A}) \to 1]$ must be zero.

Next we show that also $\Pr[\neg\mathsf{E} \wedge \neg\mathsf{C} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] = 0$. For this, note that event C is composed of the event that there is a k-collision, denoted by C_k, and the event that there is an inner collision, denoted by C_c. More precisely, we have $\mathsf{C} = \mathsf{C}_k \vee \mathsf{C}_c$, which implies that $\neg\mathsf{C} = \neg\mathsf{C}_k \wedge \neg\mathsf{C}_c$ holds. Using this, we obtain that

$$\begin{aligned}\Pr[\neg\mathsf{E} \wedge \neg\mathsf{C} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] &= \Pr[\neg\mathsf{E} \wedge \neg\mathsf{C}_k \wedge \neg\mathsf{C}_c \wedge \mathsf{pbind}(\mathcal{A}) \to 1] \\ &\leq \Pr[\neg\mathsf{E} \wedge \neg\mathsf{C}_k \wedge \mathsf{pbind}(\mathcal{A}) \to 1]\end{aligned} \tag{6}$$

and hence it suffices to show $\Pr[\neg\mathsf{E} \wedge \neg\mathsf{C}_k \wedge \mathsf{pbind}(\mathcal{A}) \to 1] = 0$, which will be done in the following.

Since $\mathcal{A}$ wins game bind, we can deduce that $(K, N) \neq (\overline{K}, \overline{N})$ and $\lceil S_{l+1} \rceil_k = \textsc{SlFunc}(K, N) = \textsc{SlFunc}(\overline{K}, \overline{N}) = \lceil \overline{S}_{l+1} \rceil_k$. By assumption event $\neg\mathsf{E}$ holds, that means $N = \overline{N}$, which in turn implies that $K \neq \overline{K}$. By construction of $\textsc{SlFunc}$, we have $Y_0 = K \neq \overline{K} = \overline{Y}_0$ and hence $\lceil \rho(Y_0) \rceil_k \neq \lceil \rho(\overline{Y}_0) \rceil_k$ as $\neg\mathsf{C}_k$ holds. Then we know in particular that $S_1 = \rho(Y_0) \neq \rho(\overline{Y}_0) = \overline{S}_1$. By assumption, we have $N = \overline{N}$ and thus obtain

$$Y_1 = S_1 \oplus (N_1 \parallel 0^c) \neq \overline{S}_1 \oplus (\overline{N}_1 \parallel 0^c) = \overline{Y}_1\,,$$

where $N_1 = \lceil N \rceil_r$ and $\overline{N}_1 = \lceil \overline{N} \rceil_r$. This puts us in the same situation we started with and by repeating the above argument for all $i \in \{1, \ldots, l-1\}$, we get $Y_l \neq \overline{Y}_l$ and hence $\lceil S_{l+1} \rceil_k = \lceil \rho(Y_l) \rceil_k \neq \lceil \rho(\overline{Y}_l) \rceil_k = \lceil \overline{S}_{l+1} \rceil_k$. This contradicts the assumption that $\mathcal{A}$ is a successful adversary hence the probability that $\mathcal{A}$ wins the game pbind and simultaneously $\neg\mathsf{E}$ and $\neg\mathsf{C}_k$ hold is zero. Hence we have shown

$$\Pr[\neg\mathsf{E} \wedge \neg\mathsf{C}_k \wedge \mathsf{pbind}(\mathcal{A}) \to 1] = 0\,, \tag{7}$$

which proves by Eq. (6), that also $\Pr[\neg\mathsf{E} \wedge \neg\mathsf{C} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] = 0$. Note that for this part of the proof, we split up C into C_k and C_c to emphasize that we need just the absence of k-collisions (i.e., $\neg\mathsf{C}_k$) to show $\Pr[\neg\mathsf{E} \wedge \neg\mathsf{C} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] = 0$. This will become relevant later in the proof when the bound for the wbind advantage is proven.

As both summands in Eq. (5) were shown to be zero, Eq. (4) simplifies to

$$\Pr[\mathsf{pbind}(\mathcal{A}) \to 1] = \Pr[\mathsf{C} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] \leq \Pr[\mathsf{C}]\,.$$

For an adversary $\mathcal{A}$ with q queries to the random transformation ρ, the probability to find a k-collision is

$$\sum_{j=1}^{q-1} \frac{j2^{n-k}}{2^n} = \sum_{j=1}^{q-1} \frac{j}{2^k} = \frac{q^2-q}{2^{k+1}}$$

and the probability to find an inner collision is

$$\sum_{j=1}^{q-1} \frac{j2^{n-c}}{2^n} = \sum_{j=1}^{q-1} \frac{j}{2^c} = \frac{q^2-q}{2^{c+1}}\,.$$

As C is the event that there is either a k-collision or an inner collision, we obtain

$$\mathbf{Adv}^{\mathsf{wbind}}_{\textsc{SLFunc}}(\mathcal{A}) = \Pr[\mathsf{pbind}(\mathcal{A}) \to 1] \leq \Pr[\mathsf{C}] = \frac{q^2-q}{2^{k+1}} + \frac{q^2-q}{2^{c+1}}\,,$$

which proves the first part of the theorem.

From this we can easily derive the bound for the pbind advantage, as for $k = n$ it holds that game pbind equals game bind and thus

$$\mathbf{Adv}^{\mathsf{bind}}_{\textsc{SLFunc}}(\mathcal{A}) = \mathbf{Adv}^{\mathsf{pbind}}_{\textsc{SLFunc}}(\mathcal{A}) \leq \frac{q^2-q}{2^{c+1}} + \frac{q^2-q}{2^{n+1}}\,.$$

It is left to prove the bound for the advantage of a wbind adversary $\mathcal{A}$. Note that for such an $\mathcal{A}$ we have $k = n$ and the event E can never occur. The latter is the case, as $K \neq \overline{K}$, $N = \overline{N}$, and $S_{l+1} = \overline{S}_{l+1}$ has to hold for the output of $\mathcal{A}$, otherwise $\mathcal{A}$ loses game pbind. Therefore it holds that

$$\begin{aligned}\Pr[\mathsf{pbind}(\mathcal{A}) \to 1] &= \Pr[\mathsf{E} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] + \Pr[\neg\mathsf{E} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] \\ &= \Pr[\neg\mathsf{E} \wedge \mathsf{pbind}(\mathcal{A}) \to 1]\,.\end{aligned}$$

As before, we let C_n be the event that there is a n-collision, i.e., that there a are two different inputs such that the respective outputs of ρ collide. Then we obtain

$$\begin{aligned}\Pr[\neg\mathsf{E} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] &= \Pr[\neg\mathsf{E} \wedge \neg\mathsf{C}_n \wedge \mathsf{pbind}(\mathcal{A}) \to 1] + \Pr[\neg\mathsf{E} \wedge \mathsf{C}_n \wedge \mathsf{pbind}(\mathcal{A}) \to 1] \\ &\leq \Pr[\neg\mathsf{E} \wedge \neg\mathsf{C}_n \wedge \mathsf{pbind}(\mathcal{A}) \to 1] + \Pr[\mathsf{C}_n] \\ &= \Pr[\mathsf{C}_n]\,,\end{aligned}$$

where the last equality holds by Eq. (7) for $k = n$. Putting together the above results, and using the bound that was computed for the probability of n-collisions, yields

$$\mathbf{Adv}^{\mathsf{wbind}}_{\textsc{SLFunc}}(\mathcal{A}) = \Pr[\mathsf{pbind}(\mathcal{A}) \to 1] = \Pr[\neg\mathsf{E} \wedge \mathsf{pbind}(\mathcal{A}) \to 1] \leq \Pr[\mathsf{C}_n] = \frac{q^2-q}{2^{n+1}}\,.$$

This finishes the proof of the theorem. □

B.5 Proof of Theorem 7

Proof. For sake of simplicity, we assume that m is a multiple of the rate r and we set $l = \frac{m}{r}$. Finding a collision for SPRG equals finding distinct states $z_1 \neq z_2$ that agree on their outer state (first r bits) for l invocations of ρ. It holds that

$$\begin{aligned}\mathbf{Adv}^{\mathsf{CR}}_{\mathrm{SPRG}}(\mathcal{A}) = \Pr\big[&\lceil z_1 \rceil_r = \lceil z_2 \rceil_r \wedge \lceil \rho^1(z_1) \rceil_r = \lceil \rho^1(z_2) \rceil_r \\ &\wedge \ldots \wedge \lceil \rho^l(z_1) \rceil_r = \lceil \rho^l(z_2) \rceil_r \mid z_1, z_2 \leftarrow \mathcal{A}^\rho() \big] \,.\end{aligned}$$

We define events $\mathsf{E}_0, \mathsf{E}_1, \ldots, \mathsf{E}_l$, where E_i equals $\lceil \rho^i(z_1) \rceil_r = \lceil \rho^i(z_2) \rceil_r$, for $z_1, z_2 \leftarrow \mathcal{A}^\rho()$. Thus

$$\begin{aligned}\mathbf{Adv}^{\mathsf{CR}}_{\mathrm{SPRG}}(\mathcal{A}) &= \Pr[\mathsf{E}_0 \wedge \mathsf{E}_1 \wedge \ldots \wedge \mathsf{E}_l] \\ &\leq \Pr[\mathsf{E}_1 \wedge \ldots \wedge \mathsf{E}_l] \\ &= \Pr[\mathsf{E}_1] \cdot \Pr[\mathsf{E}_2 \mid \mathsf{E}_1] \cdot \ldots \cdot \Pr[\mathsf{E}_l \mid \mathsf{E}_1 \wedge \ldots \wedge \mathsf{E}_{l-1}] \\ &= \Pr[\mathsf{E}_1] \cdot \prod_{i=2}^{l} \Pr[\mathsf{E}_i \mid \mathsf{E}_1 \wedge \ldots \wedge \mathsf{E}_{i-1}] \,.\end{aligned}$$

Bounding event E_1 is a simple counting argument. Since $\mathcal{A}$ makes q queries to ρ, let X_i be the event that the i-th query to ρ triggers E_1. It holds that

$$\Pr[\mathsf{E}_1] \leq \sum_{i=1}^{q} \Pr[\mathsf{X}_i] = \sum_{i=1}^{q} \frac{(i-1)2^c}{2^n} = \sum_{i=1}^{q} \frac{i-1}{2^r} = \frac{q^2 - q}{2^{r+1}} \,.$$

Note that only the probability for E_1 depends on q as $\mathcal{A}$ can only influence the outcome of the first application of ρ with his queries. For the remaining events $(\mathsf{E}_2, \ldots, \mathsf{E}_l)$, the fact that ρ is a random function yields that the events are independent from one another which gives for all $i \in \{2, \ldots, l\}$

$$\Pr[\mathsf{E}_i \mid \mathsf{E}_1 \wedge \ldots \wedge \mathsf{E}_{i-1}] = \Pr[\mathsf{E}_i] \leq \frac{1}{2^r} \,.$$

Collecting the above finally provides

$$\mathbf{Adv}^{\mathsf{CR}}_{\mathrm{SPRG}}(\mathcal{A}) \leq \Pr[\mathsf{E}_1] \cdot \prod_{i=2}^{l} \Pr[\mathsf{E}_i \mid \mathsf{E}_1 \wedge \ldots \wedge \mathsf{E}_{i-1}] \leq \frac{q^2 - q}{2^{r+1}} \cdot \prod_{i=2}^{l} \frac{1}{2^r} = \frac{q^2 - q}{2^{m+1}} \,.$$

This finishes the proof. □

B.6 Hierarchy of Binding Notions

The following theorem describes the relation between the different binding notions.

Theorem 11. *Let* $\mathrm{F} \colon \mathcal{K} \times \{0,1\}^x \to \{0,1\}^y$ *be a function. Then for any adversary* $\mathcal{A}$ *there exists an adversary* $\mathcal{B}$ *such that*

$$\mathbf{Adv}^{\mathsf{bind}}_{\mathrm{F}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{pbind}}_{\mathrm{F}}(\mathcal{B})$$

and for any adversary $\mathcal{C}$ *there exists an adversary* $\mathcal{D}$ *such that*

$$\mathbf{Adv}^{\mathsf{wbind}}_{\mathrm{F}}(\mathcal{C}) \leq \mathbf{Adv}^{\mathsf{bind}}_{\mathrm{F}}(\mathcal{D}) \,.$$

Proof. For the first part, we consider a bind adversary $\mathcal{A}$ against F and construct a pbind adversary $\mathcal{B}$ against F. It runs $((K, X), (\overline{K}, \overline{X})) \leftarrow \mathcal{A}$ and outputs $((K, X), (\overline{K}, \overline{X}))$. If $\mathcal{A}$

succeeds then $(K, X) \neq (\overline{K}, \overline{X})$ and $\mathrm{F}(K, X) = \mathrm{F}(\overline{K}, \overline{X})$. The latter implies in particular that $\lceil Y_1 \rceil_l = \lceil Y_2 \rceil_l$ for $Y_i \leftarrow \mathrm{F}(K_i, X_i)$. Thus $\mathcal{B}$ is also successful in game pbind.

For the second part, let $\mathcal{C}$ be a wbind adversary against F and construct a bind adversary $\mathcal{D}$ against F. It runs $((K, X), (\overline{K}, \overline{X})) \leftarrow \mathcal{A}$ and outputs $((K, X), (\overline{K}, \overline{X}))$. If $\mathcal{C}$ succeeds then $K \neq \overline{K}$ and $\mathrm{F}(K, X) = \mathrm{F}(\overline{K}, \overline{X})$. Note that the first condition implies in particular that $(K, X) \neq (\overline{K}, \overline{X})$ and hence $\mathcal{D}$ is successful as well. $\square$