# A Long Tweak Goes a Long Way:
# High Multi-user Security Authenticated
# Encryption from Tweakable Block Ciphers

Benoît Cogliati[1], Jérémy Jean[2], Thomas Peyrin[3], and Yannick Seurin[2]

[1] CISPA Helmholtz Center for Information Security
benoit.cogliati@cispa.saarland

[2] ANSSI, Paris, France
{jeremy.jean,yannick.seurin}@ssi.gouv.fr

[3] Nanyang Technological University, Singapore
thomas.peyrin@ntu.edu.sg

**Abstract.** We analyze the multi-user (mu) security of a family of nonce-based authentication encryption (nAE) schemes based on a tweakable block cipher (TBC). The starting point of our work is an analysis of the mu security of the SCT-2 mode which underlies the nAE scheme Deoxys-II, winner of the CAESAR competition for the *defense-in-depth* category. We extend this analysis in two directions, as we detail now.
First, we investigate the mu security of several TBC-based variants of the counter encryption mode (including CTRT, the encryption mode used within SCT-2) that differ by the way a nonce, a random value, and a counter are combined as tweak and plaintext inputs to the TBC to produce the keystream blocks that will mask the plaintext blocks. Then, we consider the authentication part of SCT-2 and study the mu security of the nonce-based MAC Nonce-as-Tweak (NaT) built from a TBC and an almost universal (AU) hash function. We also observe that the standard construction of an AU hash function from a (T)BC can be proven secure under the assumption that the underlying TBC is *unpredictable* rather than pseudorandom, allowing much better conjectures on the concrete AU advantage. This allows us to derive the mu security of the family of nAE modes obtained by combining these encryption/MAC building blocks through the NSIV composition method.
Some of these modes require an underlying TBC with a larger tweak length than what is usually available for existing ones. We then show the practicality of these modes by instantiating them with two new TBC constructions, Deoxys-TBC-512 and Deoxys-TBC-640, which can be seen as natural extensions of the Deoxys-TBC family to larger tweak lengths. Designing such TBCs with unusually large tweaks is prone to pitfalls: Indeed, we show that a large-tweak proposal for SKINNY published at EUROCRYPT 2020 presents an inherent construction flaw. We therefore provide a sound design strategy to construct large-tweak TBCs within the Superposition Tweakey (STK) framework, leading to new Deoxys-TBC and SKINNY variants. We provide software benchmarks indicating that while ensuring a very high security level, the performances of our proposals remain very competitive.

# 1 Introduction

## 1.1 Background

MULTI-USER SECURITY FOR AE. Authenticated Encryption (AE), providing confidentiality and integrity in a single primitive, is arguably among the most widely used component in applied cryptography. Its usage is widespread to protect all modern communication protocols, such as TLS, IPSec, SSH, etc. Traditionally, designers considered mainly single-user security, where an attacker is restrained to target only a single key. However, one can witness a growing concern towards large-scale adversaries, like state actors, which can potentially try to attack a large number of users at the same time, for example for mass-surveillance purpose. We observe more efforts conducted recently on multi-user security research and increased attention to these threats in standardization bodies discussions.

Multi-user (mu) security (sometimes called multi-key security) was first formally introduced for PRFs (as a technical tool) by Bellare, Canetti, and Krawczyk [BCK96] and later for public-key encryption (as a full-fledged security goal) by Bellare, Boldyreva, and Micali [BBM00]. In the mu setting, the attacker can distribute its resources to attack multiple users (all having independent keys) and is considered successful if it compromises at least one of them. In symmetric cryptography, multi-user attacks against block ciphers have been proposed by Biham [Bih02] and Fouque, Joux, and Mavromati [FJM14], and provable security results have been obtained for block ciphers [ML15, Tes15, HT16, HT17, GW18] and MACs [ADMA15, BBT16, SWGW21]. The study of the mu setting for AE has been first explored in [BT16] for the "randomized nonce" mechanism proposed for the Galois Counter-Mode with AES (AES-GCM) in TLS 1.3, an analysis later improved with tighter bounds in [LMP17, HTT18]. Results have also been obtained for the keyed duplex construction [DMA17] and AES-GCM-SIV [BHT18].

Single-user (su) security implies multi-user security, however the best generic reduction implies a security loss of $u$, where $u$ is the number of users that the adversary can simultaneously attack. Sometimes this loss is unavoidable, meaning there exist a matching attack, for example in the case of key-recovery attacks against block ciphers [Bih02]. However, for some schemes it can be shown that security does not degrade substantially when going from the su to the mu setting [ML15, Tes15, BBT16, HT16, HT17, LMP17]. In other words, in such a case, the adversary does not gain much from having the opportunity to allocate its attack resources (such as the total number of queries to oracles available in the security game) across multiple users.

TWEAKABLE BLOCK CIPHERS. Tweakable block ciphers (TBC) are block ciphers with an extra public input, the tweak, that can be used to randomize the family of permutations defined. The first published TBC was the AES competition candidate Hasty Pudding Cipher [Sch98], but the first formal treatment of TBCs was due to Liskov et al. [LRW11]. We can generally observe two main strategies to build a TBC: either from an existing BC or directly with a dedicated design [JNP14]. Since their emergence, TBCs have proven to be very flexible and

attractive primitives, leading to efficient AE operating modes [PS16] with stronger security guarantees than classical BC-based ones [RBBK01, KR11]. Indeed, a recurrent issue with BC-based AE operating modes is that security is usually only guaranteed up to the birthday bound [Fer02], i.e., up to $2^{n/2}$ queries for a $n$-bit block cipher. While this might be sufficient for most practical cases using a 128-bit cipher such as AES, future applications might suffer from such limitation. This problem is even more present when multiple users are considered, which can further reduce the security guarantees. TBC-based AE modes, in contrary, would usually provide beyond-birthday bound (BBB) security, while retaining very good performances. It is therefore natural to consider the usage of TBCs to try to tackle efficiently the problem of multi-user security as well.

## 1.2 Our Contributions

Our contributions are two-fold. On one hand, we analyze the multi-user security of a family of nonce-based authenticated encryption (nAE) modes of operation for TBCs that follow the SCT-2 blueprint. On the other hand, we design new TBCs for instantiating some of these modes which require a large tweak length.

MULTI-USER ANALYSIS OF nAE MODES FOR TBCs. We consider a family of nAE modes that follow the NSIV composition paradigm [PS16] which combines a nonce-based PRF/MAC and a nonce and IV-based encryption scheme to obtain a nonce-misuse resistant nAE scheme. This family includes in particular the SCT-2 mode which underlies the nAE scheme Deoxys-II, selected as first choice in the portfolio of the CAESAR competition for the *defense-in-depth* category, and whose only existing security analysis is in the single-user setting [JNPS21, ABPV21]. For the encryption part, we consider TBC-based variants of the counter encryption mode (including the CTRT [PS16] mode used in SCT-2) that combine a nonce, a random IV (to be generated pseudorandomly by the PRF applied to the inputs of the nAE mode once embedded in the NSIV construction), and a counter to define the tweak and plaintext inputs to the TBC to produce the keystream blocks. Our analysis can be seen as an extension of the recent work of Andreeva *et al.* [ABPV21], who carried out a similar investigation of TBC-based counter-like encryption modes, albeit limited to the single-user setting. Our proof is in the ideal (tweakable) cipher model, which is customary for multi-user analysis [BT16, BHT18, HTT18] as it allows to capture the offline computations performed by the adversary. For a TBC with key length $k$ and block length $n$, our security bounds show that some variants ensure security up to roughly $2^k$ ideal cipher queries and $2^n$ encryption queries in the nonce-respecting setting, with graceful security degradation as nonces are repeated, independently from the number of users. In other words, security does not suffer from collisions between user keys, something that can only be achieved for randomized cryptographic schemes.

Then, we consider the authentication part, for which we focus on the Nonce-as-Tweak (NaT) scheme [CLS17], a simple nonce-based PRF/MAC constructed from an almost universal (AU) hash function $H$ and a TBC $E$ and that follows

3

the classical "Hash-then-PRF" paradigm. Again, our findings (in the ideal cipher model for $E$) indicate that NaT security does not degrade in the multi-user setting and that it provides security up to roughly $2^k$ ideal cipher queries and $2^n$ tag/verification queries in the nonce-respecting setting, with graceful security degradation as nonces are repeated.

UNIVERSAL HASHING FROM UNPREDICTABLE TBCs. Equipped with our results regarding NaT with a generic AU hash function, we consider the question of how to instantiate it from a TBC.[4] The standard construction consists in concatenating the input blocks with a counter, applying the TBC (or more generally any keyed function), and xoring the outputs, something we refer informally to as the *Xor-Hash* construction, and whose origin can be traced back to Bellare *et al.* [BGR95]. Slight variants of this construction are the basis of multiple "Hash-then-PRF" MACs such as Protected Counter Sum [Ber99], PMAC [BR02], or LightMAC [LPTY16]. Usually, these constructions are proven secure under the assumption that the underlying primitive is pseudorandom. Indeed, it is not hard to prove that Xor-Hash is $\delta$-AU (i.e., for any pair of distinct inputs $X$ and $X'$, $H_K(X) = H_{K'}(K)$ with probability at most $\delta$) for $\delta = 2^{-n} + \varepsilon_{\mathrm{prp}}$ (see for example [BS20], Section 7.2.3). However, we face the problem that the NaT security bound contains a term $\mu q \delta$, where $\mu$ is the maximal number of repetitions of nonces for any user and $q$ is the total number of tag/verification queries made by the adversary. Yet for a TBC (or any keyed deterministic function) with a $k$-bit key, there exist non-uniform attacks that can distinguish it from random with advantage $2^{-k/2}$ in constant time and queries [DTT10, BL13]. As a result, provable security for NaT caps at $q \simeq 2^{k/2}$ even in the nonce-respecting setting, short of our objectives to deliver $k$-bit security.

We overcome this hurdle with the simple observation (which, to the best of our knowledge, was never made before) that Xor-Hash can be proved almost universal assuming that the underlying TBC is only unpredictable, a much weaker assumption than pseudorandomness. In particular, no attacks better than key recovery or random guessing seems to be known against unpredictability, even in the non-uniform case. See [DS09, Section 7] for a discussion of unpredictability versus pseudorandomness.

LARGE-TWEAK TBC PROPOSALS. While our new nAE modes are clean and simple, they require a larger tweak length than what is usually available from existing TBCs such as Deoxys-TBC [JNP14, JNPS16, JNPS21] or SKINNY [BJK+16]. Our second contribution is therefore to propose new TBC designs achieving larger tweak lengths. This is not a trivial task: we first show that SKINNYe-64/256, a recent large-tweak variant of SKINNY proposed at EUROCRYPT 2020 [NSS20] (64-bit block with 256-bit tweakey) presents some structural design flaw. Indeed, the STK construction extension proposed by the authors does not follow the actual STK design paradigm and allows many unwanted cancellations in the sub-tweakeys throughout the round when the attacker inserts well-chosen differences

---

[4] We could use statistical AU hash functions based for example on polynomial hashing, but we aim at a purely TBC-based design.

in the tweak input. As the number of difference cancellations is directly used in the corresponding MILP model, our finding implies that the proven bounds on the number of active Sboxes for this primitive are actually wrong.

We therefore provide a sound design strategy to construct large-tweak TBCs using the STK paradigm [JNP14], leading to new Deoxys-TBC variants. More specifically, we propose Deoxys-TBC-512 and Deoxys-TBC-640, which can be seen as natural extensions of the Deoxys-TBC family to larger tweakey lengths (512 and 640 bits, respectively). We also propose a patch for the SKINNYe-64/256 variant. Finally, we produced software benchmarks for our modes when instantiated with Deoxys-TBC-512 and Deoxys-TBC-640. They indicate that while ensuring a extremely high security level, even in the multi-user setting, the performance remains competitive. We believe these new large-tweak TBCs might find further applications such as TBC-based hashing.

COMPARISON WITH AES-GCM-SIV. To finish, we compare the security bound of our new modes with the one proven for AES-GCM-SIV in [BHT18]. Their bound is dominated by $\sigma B/2^n + d(\sigma + q_{\mathrm{ic}})/2^k$, where $\sigma$ is the total number of blocks in encryption and decryption queries, $B$ is an upper bound on the number of blocks encrypted by any user, $d$ is an upper bound on the number of users that re-use a particular nonce value, and $q_{\mathrm{ic}}$ is the number of ideal cipher queries. In comparison, the bound for our new AE mode with the best variant of the counter encryption mode (namely GCTR-Z) is dominated by

$$nq_{\mathrm{ic}}/2^k + \mu\sigma/2^k + \mu q/2^n + \min\{\mu\ell_{\max}^{\mathrm{enc}}q_{\mathrm{ver}}/2^k, Bq_{\mathrm{ver}}/2^k\},$$

where $q$ is the total number of encryption and decryption queries, $\mu$ is the maximal number of repetitions of any nonce by any user, and $\ell_{\max}^{\mathrm{enc}}$ is the maximal number of blocks in any encryption query. This calls for some comments. First, note that even in the nonce-respecting setting, $d$ can be as large as the number of encryption queries $q_{\mathrm{enc}}$, e.g. if all users rely on a counter with the same initial value to generate nonces. Hence, AES-GCM-SIV can only achieve beyond-birthday security if some small upper bound on $d$ can be enforced (e.g. when all users generate nonces randomly), or by increasing the key length of the underlying block cipher. Second, since one always has $\mu \leq B$, assuming $k \geq n$, the terms $\mu\sigma/2^k$, $\mu q/2^n$, and $\min\{\mu\ell_{\max}^{\mathrm{enc}}q_{\mathrm{ver}}/2^k, Bq_{\mathrm{ver}}/2^k\}$ are *always* smaller than $\sigma B/2^n$, and in fact *much smaller* when $\mu \ll B$ and $q_{\mathrm{ver}} \ll \sigma$, a common situation when nonces are only "mildly" misused and a large number of blocks are encrypted/decrypted per user and in total.

## 2  Preliminaries

### 2.1  General Notation and Definitions

Given a finite non-empty set $\mathcal{X}$, we let $X \leftarrow_\$ \mathcal{X}$ denote the draw of an element $X$ from $\mathcal{X}$ uniformly at random. Given a positive integer $n$, we let $\{0,1\}^n$ denote the set of all bit strings of length $n$, $\{0,1\}^{\leq n}$ denote the set of all bit strings

of length at most $n$, and $\{0,1\}^*$ denote the set of all bit strings. The empty string is denoted $\epsilon$ and the all-zero string of length $n \geq 1$ is denoted $0^n$. The length of a bit string $X$ is denoted $|X|$. The concatenation of two bit strings $X$ and $Y$ is denoted $X\|Y$. Given a bit string $X$ and an integer $n > 0$, we define $|X|_n = \lceil |X|/n \rceil$ and for $X \neq \epsilon$ we write the parsing operation of $X$ into $n$-bit blocks as $X_0\|\cdots\|X_{\ell-1} \xleftarrow{n} X$, where $\ell = |X|_n$, $|X_i| = n$ for $i = 0, \ldots, \ell - 2$ and $|X_{\ell-1}| \leq |n|$. Given an integer $n > 0$, we let $\mathsf{ozp}_n$ (or $\mathsf{ozp}$ when the parameter $n$ is implicit) denote the padding function defined for $X \neq \epsilon$ as

$$\mathsf{ozp}(X) := \begin{cases} X & \text{if } |X| = n \\ X\|1\|0^{n-|X|-1} & \text{if } |X| < n. \end{cases}$$

Given a bit string $X$ of length $i$ or larger, the $i$ leftmost bits of $X$ are denoted $\lceil X \rceil_i$ and the $i$ rightmost bits of $X$ are denoted $\lfloor X \rfloor_i$. We let $X \lll a$ denote the bit string $X$ rotated by $a$ positions to the left. Given two integers $b > 0$ and $i \geq 0$ such that $i < 2^b$, we let $\langle i \rangle_b$ denote the $b$-bit binary representation of $i$. We simply write $\langle i \rangle$ when the length $b$ is clear form the context or unspecified. Given integers $a \leq b$, we let $[\![a, b]\!]$ denote the set $\{a, \ldots, b\}$. We say that a function $F\colon \mathcal{X} \to \mathcal{Y}$ is *regular* if all $Y \in \mathcal{Y}$ have the same number of preimages by $F$. We say that a function $H\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is $\gamma$-uniform if for every $X \in \mathcal{X}$ and every $Y \in \mathcal{Y}$, $\Pr[K \leftarrow_\$ \mathcal{K}\colon H(K, X) = Y] \leq \gamma$.

ALMOST UNIVERSAL HASHING. Let $\mathcal{K}$, $\mathcal{X}$ and $\mathcal{Y}$ be two non-empty sets with $\mathcal{K}$ and $\mathcal{Y}$ finite. A keyed hash function with key space $\mathcal{K}$, domain $\mathcal{X}$, and range $\mathcal{Y}$ is a function $H\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. We write $H_K(X)$ for $H(K, X)$. We defined two notions of almost universal (AU) hashing, namely computational AU (cAU) hashing and statistical AU (sAU) hashing.

**Definition 1 (AU hash function).** *Let $\mathcal{K}$, $\mathcal{X}$ and $\mathcal{Y}$ be two non-empty sets with $\mathcal{K}$ and $\mathcal{Y}$ finite and let $\mathsf{len}\colon \mathcal{X} \to \mathbb{N}$ be some function measuring the "length" of an element of $\mathcal{X}$ (e.g., the number of $n$-bit blocks when $\mathcal{X}$ consists of bit strings). Let $H\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a keyed hash function and $\mathcal{A}$ be an adversary. The advantage of $\mathcal{A}$ against the AU security of $H$ is defined as*

$$\mathsf{Adv}_H^{\mathrm{au}}(\mathcal{A}) = \Pr[(X, X') \leftarrow \mathcal{A}(), K \leftarrow_\$ \mathcal{K}\colon X \neq X' \wedge H_K(X) = H_K(X')].$$

*Let $\delta\colon \mathbb{N} \to [0, 1]$ be some function. We say that $H$ is a $(\zeta, \delta)$-cAU hash function (w.r.t $\mathsf{len}$) if for any adversary $\mathcal{A}$ running in time at most $\zeta$ and returning $X$ and $X'$ of length at most $\ell$ (i.e., $\max\{\mathsf{len}(X), \mathsf{len}(X')\} \leq \ell$), $\mathsf{Adv}_H^{\mathrm{au}}(\mathcal{A}) \leq \delta(\ell)$. We say that $H$ is a $\delta$-sAU hash function if for any any adversary $\mathcal{A}$ (even computationally unbounded) returning $X$ and $X'$ of length at most $\ell$, $\mathsf{Adv}_H^{\mathrm{au}}(\mathcal{A}) \leq \delta(\ell)$. Equivalently, $H$ is $\delta$-sAU if and only if for every $(X, X') \in \mathcal{X}^2$ with $X \neq X'$ and $\max\{\mathsf{len}(X), \mathsf{len}(X')\} \leq \ell$, $\Pr[K \leftarrow_\$ \mathcal{K}\colon H_K(X) = H_K(X')] \leq \delta(\ell)$.*

The last equivalence is easily proven: the *if* direction is trivial; for the *only if* direction, assume towards a contradiction that there exists $(X, X') \in \mathcal{X}^2$ with $X \neq X'$ and $\max\{\mathsf{len}(X), \mathsf{len}(X')\} \leq \ell$ such that $\Pr[K \leftarrow_\$ \mathcal{K}\colon H_K(X) = H_K(X')] >$
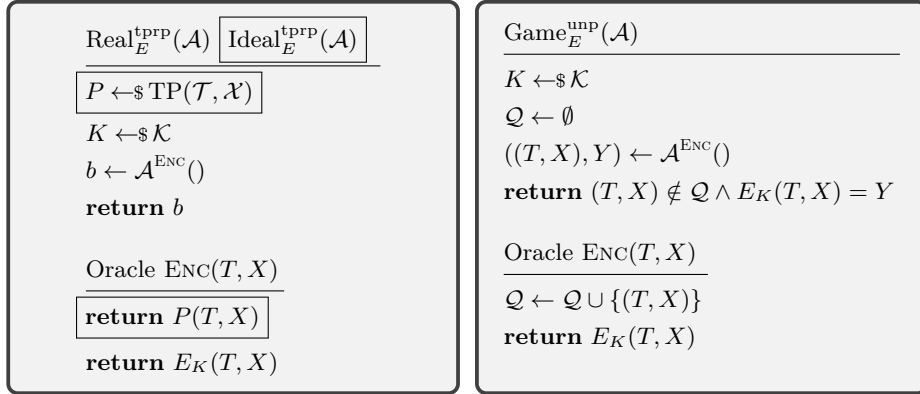
**Fig. 1.** The TPRP and UNP security games for a TBC $E \in \mathrm{TBC}(\mathcal{K}, \mathcal{T}, \mathcal{X})$.

$\delta(\ell)$ and consider the adversary $\mathcal{A}$ which computes the collision probability for every pair of messages of length at most $\ell$ and returns $(X, X')$; then $\mathcal{A}$ has AU advantage strictly larger than $\delta(\ell)$, contradicting the assumption that $H$ is $\delta$-sAU.

### 2.2 Tweakable Block Ciphers

A tweakable block cipher (TBC) with key space $\mathcal{K}$, tweak space $\mathcal{T}$, and domain $\mathcal{X}$ is a mapping $E \colon \mathcal{K} \times \mathcal{T} \times \mathcal{X} \to \mathcal{X}$ such that for any key $K \in \mathcal{K}$ and any tweak $T \in \mathcal{T}$, the mapping $E(K, T, \cdot)$ is a permutation of $\mathcal{X}$. The set of all such TBCs will be denoted $\mathrm{TBC}(\mathcal{K}, \mathcal{T}, \mathcal{X})$. Slightly abusing the notation, given positive integers $k$, $t$, and $n$, we also let $\mathrm{TBC}(k, t, n)$ denote $\mathrm{TBC}(\{0,1\}^k, \{0,1\}^t, \{0,1\}^n)$ and we often write $E_K(T, X)$ or $E_K^T(X)$ in place of $E(K, T, X)$. A tweakable permutation with tweak space $\mathcal{T}$ and domain $\mathcal{X}$ is a mapping $P \colon \mathcal{T} \times \mathcal{X} \to \mathcal{X}$ such that for any tweak $T \in \mathcal{T}$, the mapping $P(T, \cdot)$ is a permutation of $\mathcal{X}$. The set of all tweakable permutations with tweak space $\mathcal{T}$ and domain $\mathcal{X}$ will be written $\mathrm{TP}(\mathcal{T}, \mathcal{X})$. We consider two security notions for TBCs, namely pseudorandomness and unpredictability.

**Definition 2 (TPRP and UNP security).** *Let $E \in \mathrm{TBC}(\mathcal{K}, \mathcal{T}, \mathcal{X})$ and $\mathcal{A}$ be an adversary. Consider games* $\mathrm{Real}_E^{\mathrm{tprp}}(\mathcal{A})$ *and* $\mathrm{Ideal}_E^{\mathrm{tprp}}(\mathcal{A})$ *defined in Figure 1 (left). The advantage of $\mathcal{A}$ in breaking the TPRP security of $E$ is defined as*

$$\mathsf{Adv}_E^{\mathrm{tprp}}(\mathcal{A}) = \left| \Pr\left[1 \leftarrow \mathrm{Real}_E^{\mathrm{tprp}}(\mathcal{A})\right] - \Pr\left[1 \leftarrow \mathrm{Ideal}_E^{\mathrm{tprp}}(\mathcal{A})\right] \right|.$$

*Consider game* $\mathrm{Game}_E^{\mathrm{unp}}(\mathcal{A})$ *defined in Figure 1 (right). The advantage of $\mathcal{A}$ in breaking the UNP security of $E$ is defined as*

$$\mathsf{Adv}_E^{\mathrm{unp}}(\mathcal{A}) = \Pr[\mathbf{true} \leftarrow \mathrm{Game}_E^{\mathrm{unp}}(\mathcal{A})].$$

THE IDEAL TWEAKABLE CIPHER MODEL. Our proofs will use the ideal (tweakable) cipher model, where the TBC underlying a specific construction is drawn uniformly at random from the set of all TBCs with the adequate key space, tweak space, and message space, and given as a black box that can be queried in the forward and backward direction by the adversary. More formally, given a cryptographic scheme $\Pi$ based on a TBC $E \in \text{TBC}(k, t, n)$, the game defining the security of $\Pi$ is modified as follows: at the beginning of the game, a random TBC $E_{\text{ic}}$ is drawn uniformly at random from $\text{TBC}(k, t, n)$ and the adversary is given access to an encryption oracle IC and a decryption oracle $\text{IC}^{-1}$ such that a query $\text{IC}(K, T, X)$ returns $E_{\text{ic}}(K, T, X)$ and a query $\text{IC}^{-1}(K, T, Y)$ returns $E_{\text{ic}}^{-1}(K, T, Y)$.

### 2.3 Multi-user Security Notions

Below we specify the multi-user (mu) security notions for encryption, authentication, and authenticated encryption used in this paper.

NONCE AND IV-BASED ENCRYPTION SCHEME. The notion of combined nonce and IV-based encryption (nivE) scheme was introduced in [PS16]. Syntactically, an nivE scheme is a tuple $\Pi = (\mathcal{K}, \mathcal{N}, \mathcal{R}, \mathcal{M}, \mathcal{C}, \mathsf{Enc}, \mathsf{Dec})$ where $\mathcal{K}$ is the key space, $\mathcal{N}$ is the nonce space, $\mathcal{R}$ the random value[5] space, $\mathcal{M}$ the message space, and $\mathcal{C}$ is the ciphertext space, all being non-empty sets of bit strings with $\mathcal{K}$, $\mathcal{N}$, and $\mathcal{R}$ finite, and $\mathsf{Enc}$ and $\mathsf{Dec}$ are algorithms such that:

- the encryption algorithm $\mathsf{Enc}$ takes as input a tuple $(K, N, R, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{R} \times \mathcal{M}$ and outputs a ciphertext $C \in \mathcal{C}$;
- the decryption algorithm takes as input a tuple $(K, N, R, C) \in \mathcal{K} \times \mathcal{N} \times \mathcal{R} \times \mathcal{C}$ and outputs a plaintext $M \in \mathcal{M}$.

We require that for all tuples $(K, N, R, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{R} \times \mathcal{M}$, one has $\mathsf{Dec}(K, N, R, \mathsf{Enc}(K, N, R, M)) = M$. We also require that if $\mathcal{M}$ contains a bit string of length $m$, then it contains all bit strings of length $m$, and that $|\mathsf{Enc}(K, N, R, M)| = |M|$ for all $(K, N, R, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{R} \times \mathcal{M}$. We write $\mathsf{Enc}_K(N, R, M)$ for $\mathsf{Enc}(K, N, R, M)$ and similarly for $\mathsf{Dec}$. The multi-user security of an nivE scheme is defined as follows.

**Definition 3 (mu-nivE security).** *Let* $\Pi = (\mathcal{K}, \mathcal{N}, \mathcal{R}, \mathcal{M}, \mathcal{C}, \mathsf{Enc}, \mathsf{Dec})$ *be an nivE scheme and let* $\mathcal{A}$ *be an adversary. Consider games* $\text{Real}_{\Pi}^{\text{mu-nive}}(\mathcal{A})$ *and* $\text{Ideal}_{\Pi}^{\text{mu-nive}}(\mathcal{A})$ *defined in Figure 2. The advantage of* $\mathcal{A}$ *in breaking the mu-nivE security of* $\Pi$ *is defined as*

$$\mathsf{Adv}_{\Pi}^{\text{mu-nive}}(\mathcal{A}) = \left| \Pr\left[ 1 \leftarrow \text{Real}_{\Pi}^{\text{mu-nive}}(\mathcal{A}) \right] - \Pr\left[ 1 \leftarrow \text{Ideal}_{\Pi}^{\text{mu-nive}}(\mathcal{A}) \right] \right|.$$

---

[5] Although we keep the security notion name "nivE" for historical reasons, we deliberately avoid calling this input an *initial value (IV)* as it is not used to initialize the counter, and prefer the more neutral term "random value".

| $\mathrm{Real}_{\Pi}^{\text{mu-nive}}(\mathcal{A})$  $\boxed{\mathrm{Ideal}_{\Pi}^{\text{mu-nive}}(\mathcal{A})}$ | Oracle NEW() | Oracle ENC$(i, N, M)$ |
|---|---|---|
| $u := 0$ | $u := u + 1$ | **if** $i \notin [\![1, u]\!]$ **then** |
| $b \leftarrow \mathcal{A}^{\text{NEW,ENC}}()$ | $K_u \leftarrow\!\!\$\, \mathcal{K}$ | $\quad$ **return** $\perp$ |
| **return** $b$ | **return** $\epsilon$ | $R \leftarrow\!\!\$\, \mathcal{R}$ |
| | | $C \leftarrow \mathsf{Enc}_{K_i}(N, R, M)$ |
| | | $\boxed{(R, C) \leftarrow\!\!\$\, \mathcal{R} \times \{0,1\}^{|M|}}$ |
| | | **return** $(R, C)$ |

**Fig. 2.** The mu-nivE security games for a nivE scheme $\Pi = (\mathcal{K}, \mathcal{N}, \mathcal{R}, \mathcal{M}, \mathcal{C}, \mathsf{Enc}, \mathsf{Dec})$.

NONCE-BASED PSEUDORANDOM MACS. A nonce-based pseudorandom MAC (nPRMAC for short) is a tuple $\Pi = (\mathcal{K}, \mathcal{N}, \mathcal{U}, \mathcal{V}, \mathsf{Tag})$, where $\mathcal{K}$ is the key space, $\mathcal{N}$ is the nonce space, $\mathcal{U}$ is the domain, and $\mathcal{V}$ is the tag space, all being non-empty sets with $\mathcal{K}$, $\mathcal{N}$, and $\mathcal{V}$ finite, and $\mathsf{Tag}$ is a deterministic algorithm which takes as input a tuple $(K, N, U) \in \mathcal{K} \times \mathcal{N} \times \mathcal{U}$ and returns a tag $V \in \mathcal{V}$. We write $\mathsf{Tag}_K(N, X)$ for $\mathsf{Tag}(K, N, X)$. The following security notion introduced in [JNPS21] combines the (nonce-based) PRF and MAC security notions.

**Definition 4 (mu-nPRMAC security).** *Let* $\Pi = (\mathcal{K}, \mathcal{N}, \mathcal{U}, \mathcal{V}, \mathsf{Tag})$ *be a nonce-based pseudorandom MAC and let* $\mathcal{A}$ *be an adversary. Consider games* $\mathrm{Real}_{\Pi}^{\text{mu-nprmac}}(\mathcal{A})$ *and* $\mathrm{Ideal}_{\Pi}^{\text{mu-nprmac}}(\mathcal{A})$ *defined in Figure 3. The advantage of* $\mathcal{A}$ *in breaking the mu-nPRMAC security of* $\Pi$ *is defined as*

$$\mathsf{Adv}_{\Pi}^{\text{mu-nprmac}}(\mathcal{A}) = |\Pr[1 \leftarrow \mathrm{Real}_{\Pi}^{\text{mu-nprmac}}(\mathcal{A})] - \Pr[1 \leftarrow \mathrm{Ideal}_{\Pi}^{\text{mu-nprmac}}(\mathcal{A})]|.$$

NONCE-BASED AUTHENTICATED ENCRYPTION. A nonce-based authenticated encryption scheme with associated data (nAE scheme for short) is a tuple $\Pi = (\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{C}, \mathcal{V}, \mathsf{Enc}, \mathsf{Dec})$ where $\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{C}$, and $\mathcal{V}$ are non-empty sets of bit strings with $\mathcal{K}$, $\mathcal{N}$, and $\mathcal{V}$ finite and $\mathsf{Enc}$ and $\mathsf{Dec}$ are deterministic algorithms such that:

- the encryption algorithm $\mathsf{Enc}$ takes as input a key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$, associated data $A \in \mathcal{A}$, and a message $M \in \mathcal{M}$ and outputs a ciphertext $C \in \mathcal{C}$ and a tag $V \in \mathcal{V}$;
- the decryption algorithm $\mathsf{Dec}$ takes as input a key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$, associated data $A \in \mathcal{A}$, a ciphertext $C \in \mathcal{C}$ and a tag $V \in \mathcal{V}$, and outputs either a message $M \in \mathcal{M}$ or a special symbol $\perp$ that indicates that decryption failed.

We require that for all tuples $(K, N, A, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}$, one has $\mathsf{Dec}(K, N, A, \mathsf{Enc}(K, N, A, M)) = M$. We also require that if $\mathcal{M}$ contains a bit string of length $m$, then it contains all bit strings of length $m$, and that for all

**Fig. 3.** The mu-nPRMAC security games for a nonce-based pseudorandom MAC $\Pi = (\mathcal{K}, \mathcal{N}, \mathcal{U}, \mathcal{V}, \mathsf{Tag})$.

$(K, N, A, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}$, $|C| = |M|$ where $(C, V) = \mathsf{Enc}(K, N, A, M)$. We write $\mathsf{Enc}_K(N, A, M)$ and $\mathsf{Dec}_K(N, A, C, V)$ in place of $\mathsf{Enc}(K, N, A, M)$ and $\mathsf{Dec}(K, N, A, C, V)$. The multi-user security of an nAE scheme is defined as follows.
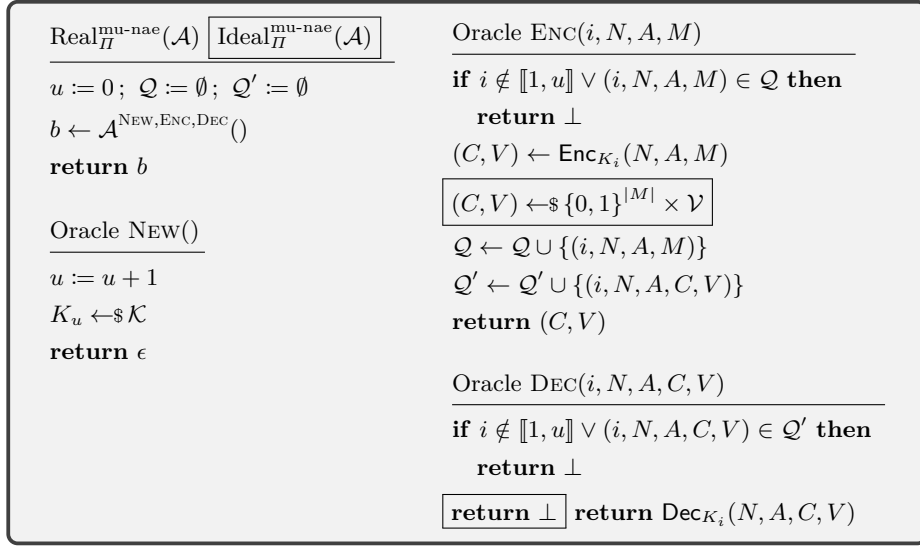
**Definition 5 (mu-nAE security).** *Let* $\Pi = (\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{C}, \mathcal{V}, \mathsf{Enc}, \mathsf{Dec})$ *be an nAE scheme and* $\mathcal{A}$ *be an adversary. Consider games* $\mathrm{Real}_\Pi^{\mathrm{mu\text{-}nae}}(\mathcal{A})$ *and* $\mathrm{Ideal}_\Pi^{\mathrm{mu\text{-}nae}}(\mathcal{A})$ *defined in Figure 4. The advantage of* $\mathcal{A}$ *in breaking the mu-nAE security of* $\Pi$ *is defined as*

$$\mathsf{Adv}_\Pi^{\mathrm{mu\text{-}nae}}(\mathcal{A}) = |\Pr[1 \leftarrow \mathrm{Real}_\Pi^{\mathrm{mu\text{-}nae}}(\mathcal{A})] - \Pr[1 \leftarrow \mathrm{Ideal}_\Pi^{\mathrm{mu\text{-}nae}}(\mathcal{A})]| \, .$$

### 2.4 The H-coefficients Technique

Let us fix a deterministic adversary $\mathcal{A}$ trying to break a cryptographic scheme $\Pi$ whose security is defined via a distinguishing experiment specified by two games $\mathrm{Real}_\Pi(\mathcal{A})$ and $\mathrm{Ideal}_\Pi(\mathcal{A})$. We summarize the attack in a queries transcript $\tau$ encoding all queries made by the adversary to the oracles available in the games together with their answer. Let $\Lambda_{\mathrm{re}}$ and $\Lambda_{\mathrm{id}}$ be two random variables sampled according to the probability distribution of $\tau$ in the real and in the ideal world respectively. We let $\Theta$ denote the set of all *attainable* queries transcripts, i.e. transcripts $\tau$ such that $\Pr[\Lambda_{\mathrm{id}} = \tau] > 0$. One has the following result, the proof of which can be found for example in [CS14].

**Theorem 1 (H-coefficients technique [Pat09]).** *Fix a deterministic adversary* $\mathcal{A}$*. Let* $\Theta_{\mathrm{bad}}$ *and* $\Theta_{\mathrm{good}}$ *be two disjoint subsets of* $\Theta$ *such that* $\Theta =$

$\mathrm{Real}_{\Pi}^{\mathrm{mu\text{-}nae}}(\mathcal{A})$ $\boxed{\mathrm{Ideal}_{\Pi}^{\mathrm{mu\text{-}nae}}(\mathcal{A})}$

$u := 0\,;\ \mathcal{Q} := \emptyset\,;\ \mathcal{Q}' := \emptyset$

$b \leftarrow \mathcal{A}^{\mathrm{New},\mathrm{Enc},\mathrm{Dec}}()$

**return** $b$

Oracle $\mathrm{New}()$

$u := u + 1$

$K_u \leftarrow\!\!\$\ \mathcal{K}$

**return** $\epsilon$

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

**if** $i \notin [\![1, u]\!] \vee (i, N, A, M) \in \mathcal{Q}$ **then**

   **return** $\perp$

$(C, V) \leftarrow \mathsf{Enc}_{K_i}(N, A, M)$

$\boxed{(C, V) \leftarrow\!\!\$\ \{0,1\}^{|M|} \times \mathcal{V}}$

$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(i, N, A, M)\}$

$\mathcal{Q}' \leftarrow \mathcal{Q}' \cup \{(i, N, A, C, V)\}$

**return** $(C, V)$

Oracle $\mathrm{Dec}(i, N, A, C, V)$

**if** $i \notin [\![1, u]\!] \vee (i, N, A, C, V) \in \mathcal{Q}'$ **then**

   **return** $\perp$

$\boxed{\textbf{return } \perp}$ **return** $\mathsf{Dec}_{K_i}(N, A, C, V)$

**Fig. 4.** The mu-nAE security games for a nAE scheme $\Pi = (\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{C}, \mathcal{V}, \mathsf{Enc}, \mathsf{Dec})$.

$\Theta_{\mathrm{bad}} \cup \Theta_{\mathrm{good}}$. *Assume that there exists $\beta \geq 0$ such that, for every $\tau \in \Theta_{\mathrm{good}}$,*

$$\frac{\Pr[\Lambda_{\mathrm{re}} = \tau]}{\Pr[\Lambda_{\mathrm{id}} = \tau]} \geq 1 - \beta.$$

*Then one has*

$$\mathsf{Adv}_{\Pi}(\mathcal{A}) := |\Pr[1 \leftarrow \mathrm{Real}_{\Pi}(\mathcal{A})] - \Pr[1 \leftarrow \mathrm{Ideal}_{\Pi}(\mathcal{A})]| \leq \Pr[\Lambda_{\mathrm{id}} \in \Theta_{\mathrm{bad}}] + \beta.$$

## 3  Multi-user Security of GCTR Encryption

In this section, we focus on the encryption part of the AE scheme and consider several variants of the counter encryption mode. Our goal is to fit three strings (a nonce $N$, a random value $R$, and a counter $j$) into the tweak and the plaintext inputs of a tweakable block cipher. Following Andreeva *et al.* [ABPV21], we define a family of nivE schemes called *generic CTR* (GCTR) based on a TBC $E$. For the remainder of this section, we fix positive integers $k$, $t$, and $n$ and a TBC $E \in \mathrm{TBC}(k, t, n)$. The schemes are parameterized by positive integers $\nu$, $r$, and $c$ that denote respectively the nonce length, the random value length, and the counter length, all expressed in bits, and two functions $f_T$ and $f_X$ taking as input a tuple $(N, R, j) \in \{0,1\}^{\nu} \times \{0,1\}^r \times [\![0, 2^c - 1]\!]$ and returning respectively an element of the tweak space and of the plaintext space of $E$. Given $E$, $f_T$, and $f_X$, the nivE scheme $\mathsf{GCTR}[E, f_T, f_X]$ has key space $\{0,1\}^k$, nonce space $\{0,1\}^{\nu}$, random value space $\{0,1\}^r$, message and ciphertext spaces $\{0,1\}^{\leq 2^c n}$,

$$\boxed{\begin{array}{ll}
\mathsf{GCTR}[E, f_T, f_X].\mathsf{Enc}_K(N, R, M) & \mathsf{GCTR}[E, f_T, f_X].\mathsf{Dec}_K(N, R, C)
\end{array}}$$

| $\mathsf{GCTR}[E, f_T, f_X].\mathsf{Enc}_K(N, R, M)$ | $\mathsf{GCTR}[E, f_T, f_X].\mathsf{Dec}_K(N, R, C)$ |
|---|---|
| **if** $M = \epsilon$ **then return** $\epsilon$ | **if** $C = \epsilon$ **then return** $\epsilon$ |
| $M_0\|\cdots\|M_{\ell-1} \xleftarrow{n} M$ | $C_0\|\cdots\|C_{\ell-1} \xleftarrow{n} C$ |
| **for** $j := 0\ldots\ell-1$ **do** | **for** $j := 0\ldots\ell-1$ **do** |
| $\quad T_j := f_T(N, R, j)$ | $\quad T_j := f_T(N, R, j)$ |
| $\quad X_j := f_X(N, R, j)$ | $\quad X_j := f_X(N, R, j)$ |
| **for** $j := 0\ldots\ell-2$ **do** | **for** $j := 0\ldots\ell-2$ **do** |
| $\quad C_j := M_j \oplus E_K^{T_j}(X_j)$ | $\quad M_j := C_j \oplus E_K^{T_j}(X_j)$ |
| $/\!\!/$ last block might be incomplete | $/\!\!/$ last block might be incomplete |
| $C_{\ell-1} := M_{\ell-1} \oplus \left[ E_K^{T_{\ell-1}}(X_{\ell-1}) \right]_{\|M_{\ell-1}\|}$ | $M_{\ell-1} := C_{\ell-1} \oplus \left[ E_K^{T_{\ell-1}}(X_{\ell-1}) \right]_{\|C_{\ell-1}\|}$ |
| **return** $C_0\|\cdots\|C_{\ell-1}$ | **return** $M_0\|\cdots\|M_{\ell-1}$ |

**Fig. 5.** The encryption and decryption algorithms of the $\mathsf{GCTR}[E, f_T, f_X]$ nivE scheme based on a TBC $E \in \mathrm{TBC}(k, t, n)$.

and algorithms Enc and Dec as defined in Figure 5. A specific instantiation of the GCTR family is obtained by picking up two functions $f_T$ and $f_X$.

Andreeva *et al.* [ABPV21] performed a systematic analysis of GCTR in the single user setting, identifying 22 secure variants named GCTR-1 to GCTR-22. Here, in addition to CTRT (GCTR-3), we focus on two simple and natural options:

- two strings are concatenated and used as the tweak, while the third one is used as the plaintext block (GCTR-1, -5, and -7 in [ABPV21]);
- the three strings are concatenated and used as the tweak while the plaintext is a constant: in this case, the TBC is simply used as a PRF applied to the three strings (not analyzed in [ABPV21]).

This yields five variants in total which are listed in Table 1. Except for CTRT, we refer to a specific variant using a suffix reflecting the block input $f_X$ as indicated in the table (e.g. GCTR-N refers to the variant such that $f_T(N, R, j) = R\|\langle j\rangle_c$ and $f_X(N, R, j) = N$). We establish the multi-user security of GCTR in the ideal cipher model for $E$ in the following theorem. The proof is deferred to Appendix B.

**Theorem 2 (mu security of GCTR).** *Let $k$, $t$, and $n$ be positive integers and $E \in \mathrm{TBC}(k, t, n)$ be a tweakable block cipher modeled as an ideal tweakable cipher $(\mathrm{IC}, \mathrm{IC}^{-1})$. Let $\nu$, $r$, and $c$ be positive integers and let $(f_T, f_X)$ be one of the pairs of functions from Table 1. Let $q_{\mathrm{ic}}$, $q_{\mathrm{enc}}$, $\ell_{\max}$, $\sigma$ and $\mu$ be positive integers such that $q_{\mathrm{enc}} \leq \min\{2^k, 2^r\}$, $\ell_{\max} \leq 2^c$, and $\sigma \leq \min\{2^n, 2^r\}$. Then, for any adversary $\mathcal{A}$ making at most $q_{\mathrm{ic}}$ queries in total to $\mathrm{IC}$ or $\mathrm{IC}^{-1}$ and $q_{\mathrm{enc}}$ queries to $\mathrm{ENC}$ of maximal length (in number of $n$-bit blocks) at most $\ell_{\max}$ and of total length (in number of $n$-bit blocks) at most $\sigma$, and such that any (user, nonce)*

**Table 1.** Variants of the GCTR nivE schemes family considered in this paper. For CTRT, $+$ is addition mod $2^t$.

| name in this paper | name in [ABPV21] | $f_T(N, R, j)$ | $f_X(N, R, j)$ |
|:---:|:---:|:---:|:---:|
| CTRT | 3 | $R + \langle j \rangle_c$ | $N$ |
| GCTR-N | 1 | $R \| \langle j \rangle_c$ | $N$ |
| GCTR-R | 7 | $N \| \langle j \rangle_c$ | $R$ |
| GCTR-C | 5 | $N \| R$ | $\langle j \rangle_c$ |
| GCTR-Z | — | $N \| R \| \langle j \rangle_c$ | $\langle 0 \rangle_n$ |

*pair $(i, N)$ appears at most $\mu$ times in* ENC *queries, one has*

$$\mathsf{Adv}^{\text{mu-nive}}_{\mathsf{GCTR}[E, f_T, f_X]}(\mathcal{A}) \leq \frac{1}{2^r} + \frac{1}{2^n} + \frac{2(r+n)q_{\text{ic}}}{2^k} + g(q_{\text{ic}}, q_{\text{enc}}, \ell_{\max}, \sigma, \mu),$$

*where*

$$g(q_{\text{ic}}, q_{\text{enc}}, \ell_{\max}, \sigma, \mu) := \begin{cases} \frac{(4\mu+1)\sigma}{2^{r+1}} & \text{for } \mathsf{CTRT}, \\ \frac{\mu q_{\text{enc}}}{2^{r+1}} + \frac{\sigma}{2^{n+1}} & \text{for } \mathsf{GCTR\text{-}N}, \\ \frac{\mu q_{\text{enc}}}{2^{r+1}} + \frac{\mu\sigma}{2^n} & \text{for } \mathsf{GCTR\text{-}R}, \\ \frac{(2\mu-1)q_{\text{enc}}}{2^{r+1}} + \frac{\sigma\ell_{\max}}{2^{n+1}} + \frac{\sigma^2}{2^{k+n+1}} & \text{for } \mathsf{GCTR\text{-}C}, \\ \frac{\mu q_{\text{enc}}}{2^{r+1}} & \text{for } \mathsf{GCTR\text{-}Z}. \end{cases}$$

DISCUSSION. The common part of the bound is negligible as long as $q_{\text{ic}}$ is small compared to $2^{k-\log_2 n}$ (assuming $r \simeq n$). Omitting constant factors, one can note that the bound for all variants except CTRT have a $\mu q_{\text{enc}}/2^r$ term but differ in how $\sigma$ shows up: the security bound for GCTR-Z is always better than the one for GCTR-N, which is always better than the one for GCTR-R and GCTR-C, whose relation depend on $\mu$ and $\ell_{\max}$. Note also that the bound for GCTR-Z does not depend on $\sigma$. The bound for CTRT has a term $\mu\sigma/2^r$ and hence is always worse than the bound of GCTR-Z and GCTR-N (for $r = n$).

## 4 Multi-user Security of NaT Authentication

In this section, we focus on the authentication part of the AE scheme. In Section 4.1, we consider the nonce-based pseudorandom MAC called *Nonce-as-Tweak* (NaT) [CLS17] based on a TBC $E$ and a generic AU hash function $H$ and study its mu-nPRMAC security. In Section 4.2, we study how to instantiate $H$ from an unpredictable TBC.

### 4.1 NaT Based on a Generic AU Hash Function

Let $k$, $t$, and $n$ be positive integers, $\mathcal{K}_{\text{in}}$ and $\mathcal{U}$ be non-empty sets with $\mathcal{K}_{\text{in}}$ finite, and $\mathsf{len}\colon \mathcal{U} \to \mathbb{N}$ be some length function. Let $E \in \mathrm{TBC}(k, t, n)$ be a tweakable

block cipher and $H \colon \mathcal{K}_{\mathrm{in}} \times \mathcal{U} \to \{0,1\}^n$ be a keyed hash function. Let also $\nu$ be an integer with $\nu \leq t$. We define the nonce-based pseudorandom MAC $\mathsf{NaT}[H, E]$ with key space $\mathcal{K}_{\mathrm{in}} \times \{0,1\}^k$, nonce space $\{0,1\}^\nu$, domain $\mathcal{U}$, and tag space $\{0,1\}^n$ as

$$\mathsf{NaT}[H, E].\mathsf{Tag}_{K_{\mathrm{in}}, K_{\mathrm{out}}}(N, U) \coloneqq E_{K_{\mathrm{out}}}^{0^{t-\nu} \| N} \left( H_{K_{\mathrm{in}}}(U) \right).$$

We establish the multi-user security of $\mathsf{NaT}$ in the ideal cipher model for $E$ and assuming that $H$ is a sAU hash function in the following theorem using the H-coefficients technique. The proof is deferred to Appendix C. We conjecture that a similar result can be proven assuming $H$ is a cAU hash function. However, as far as we can tell, the proof of this conjecture cannot use the H-coefficients technique if $H$ is only assumed computationally secure as there does not seem to be a way to replace it with a statistically secure counterpart as a first step in the reasoning. Hence, a game-based proof would be required instead, allowing to construct a reduction to the cAU security of $H$. We leave this for future work as we are able to prove that the TBC-based instantiation of $H[E_{\mathrm{in}}]$ defined in Section 4.2 is sAU under a plausible unpredictability assumption on the underlying TBC $E_{\mathrm{in}}$. We will also justify the specific form we assume for $\delta(\ell)$ and $\gamma(\ell)$.

**Theorem 3 (mu security of $\mathsf{NaT}$).** *Let $k$, $t$, and $n$ be positive integers, $\mathcal{K}_{\mathrm{in}}$ and $\mathcal{U}$ be non-empty sets with $\mathcal{K}_{\mathrm{in}}$ finite, $\mathsf{len} \colon \mathcal{U} \to \mathbb{N}$ be some length function (below, the length of an element $U \in \mathcal{U}$ refers to $\mathsf{len}(U)$), $E \in \mathrm{TBC}(k, t, n)$ be a tweakable block cipher modeled as an ideal tweakable cipher $(\mathrm{IC}, \mathrm{IC}^{-1})$, and $H \colon \mathcal{K}_{\mathrm{in}} \times \mathcal{U} \to \{0,1\}^n$ be a keyed hash function. Let $\nu$ be an integer such that $\nu \leq t$. Assume that $H$ is $\delta$-sAU and $\gamma$-uniform (w.r.t $\mathsf{len}$) for $\delta(\ell) = \gamma(\ell) = \alpha \ell / 2^k + \beta / 2^n$ with $\beta \geq 1$ (and hence $\delta \geq 2^{-n}$ and $\gamma \geq 2^{-n}$). Let $q_{\mathrm{ic}}$, $q_{\mathrm{tag}}$, $\sigma_{\mathrm{tag}}$, $q_{\mathrm{ver}}$, $\sigma_{\mathrm{ver}}$, $\ell_{\max}^{\mathrm{tag}}$, and $\mu$ be positive integers such that $q_{\mathrm{tag}} \leq 2^n$, $2q_{\mathrm{tag}} + q_{\mathrm{ic}} \leq 2^k$, and $q_{\mathrm{ic}} + \mu \leq 2^n / 2$. Then, for any (computationally unbounded) adversary $\mathcal{A}$ against the mu-nPRMAC security of $\mathsf{NaT}[H, E]$ making at most $q_{\mathrm{ic}}$ queries in total to $\mathrm{IC}$ or $\mathrm{IC}^{-1}$, $q_{\mathrm{tag}}$ queries to $\mathrm{TAG}$ of maximal length $\ell_{\max}^{\mathrm{tag}}$ and of total length at most $\sigma_{\mathrm{tag}}$, $q_{\mathrm{ver}}$ queries to $\mathrm{VER}$ of total length at most $\sigma_{\mathrm{ver}}$, and such that any (user, nonce) pair $(i, N)$ appears at most $\mu$ times in its $\mathrm{TAG}$ queries, one has*

$$\mathsf{Adv}_{\mathsf{NaT}[H,E]}^{\mathrm{mu\text{-}nprmac}}(\mathcal{A}) \leq \frac{2n q_{\mathrm{ic}}}{2^k} + \frac{\alpha \mu \sigma_{\mathrm{tag}}}{2^k} + \frac{\alpha(\mu+1)\sigma_{\mathrm{ver}}}{2^k} + \frac{\alpha \mu \ell_{\max}^{\mathrm{tag}} q_{\mathrm{ver}}}{2^k}$$
$$+ \frac{1}{2^n} + \frac{\beta \mu q_{\mathrm{tag}}}{2^n} + \frac{\beta(\mu+3)q_{\mathrm{ver}}}{2^n}.$$

*Moreover, if the total length of all $\mathrm{TAG}$ queries for any user is at most $B$, then the term $\alpha \mu \ell_{\max}^{\mathrm{tag}} q_{\mathrm{ver}} / 2^k$ can be replaced by $\alpha B q_{\mathrm{ver}} / 2^k$.*

DISCUSSION. We will justify in Section 4.2 that $\alpha$ and $\beta$ can be conjectured to be small absolute constant, hence we let $\alpha = \beta = 1$ for the sake of discussion. Then as long as $\mu$ remains small, all terms are negligible as long as $\sigma_{\mathrm{tag}}$ and $\sigma_{\mathrm{ver}}$ are small compared to $2^k$, $q_{\mathrm{tag}}$ is small compared to $2^n$, $q_{\mathrm{ver}} \ell_{\max}^{\mathrm{tag}}$ is small compared to $2^n$, and $q_{\mathrm{ic}}$ is small compared to $2^{k - \log_2 n}$. Note that in many situations, $\ell_{\max}^{\mathrm{tag}}$ can be reasonably upper bounded to provide beyond-birthday security with respect

to $q_{\text{ver}}$ (e.g., for $n = 128$ and $\ell_{\text{max}}^{\text{tag}} = 2^{32}$, security is ensured up to $2^{96}$ verification queries). When $\mu$ is allowed to grow as large as $q_{\text{tag}}$, we hit the birthday bound and security vanishes at $2^{n/2}$ tag/verification queries.

## 4.2 TBC-based Almost Universal Hashing

WARM-UP. Let $n$ and $c$ be positive integers and $\mathcal{K}$ and $\mathcal{X}$ be finite non-empty sets. Let $F : \mathcal{K} \times \mathcal{X} \times \{0,1\}^c \to \{0,1\}^n$ be a keyed function with key space $\mathcal{K}$, domain $\mathcal{X} \times \{0,1\}^c$, and range $\{0,1\}^n$. Consider the keyed hash function $F^{\oplus}$ with key space $\mathcal{K}$, domain $\mathcal{X}^{\leq 2^c}$ (the set of all sequences of elements of $\mathcal{X}$ of length at most $2^c$ equipped with the length function $\mathsf{len}$ returning the length of a sequence), and range $\{0,1\}^n$ defined for $K \in \mathcal{K}$ and $X = (X_0, \ldots, X_{a-1}) \in \mathcal{X}^{\leq 2^c}$ (with $a = \mathsf{len}(X) = 0$ if $X$ is empty) as

$$F_K^{\oplus}(X_1, \ldots, X_a) := \begin{cases} 0^n & \text{if } a = 0, \\ \bigoplus_{j=0}^{a-1} F_K(X_j, \langle j \rangle_c) & \text{if } a \geq 1. \end{cases}$$

Construction $F^{\oplus}$ is a simplified variant of the keyed hash function used for example in PMAC [BR02, Rog04]. The standard proof that $F^{\oplus}$ is AU relies on the assumption that $F$ is a PRF (see for example [BS20], Section 7.2.3). Here we show that $F^{\oplus}$ is a cAU hash function assuming $F$ is unpredictable,[6] as captured by the following theorem.

**Theorem 4.** *Let $F : \mathcal{K} \times \mathcal{X} \times \{0,1\}^c \to \{0,1\}^n$ be a keyed function with key space $\mathcal{K}$, domain $\mathcal{X} \times \{0,1\}^c$, and range $\{0,1\}^n$. Then, for any adversary $\mathcal{A}$ against the AU security of $F^{\oplus}$ running in time at most $\zeta$ and returning messages of length at most $\ell$, there exists an adversary $\mathcal{B}$ against the UNP security of $F$ making at most $2\ell$ oracle queries and running in time at most $\zeta + \alpha\ell$ for some small constant $\alpha$ that only depends on the computation model (but not on $F$), such that*

$$\mathsf{Adv}_{F^{\oplus}}^{\text{au}}(\mathcal{A}) \leq \mathsf{Adv}_F^{\text{unp}}(\mathcal{B}).$$

*Proof.* Let $\mathcal{A}$ be an adversary against the AU security of $F^{\oplus}$. We construct adversary $\mathcal{B}$ against the UNP security of $F$ as follows. (Recall that $\mathcal{B}$ has oracle access to $F_K$ for some random key $K$ and must return a pair $(x, F_K(x))$ for some $x$ that it did not query to its oracle). It runs $\mathcal{A}()$ which returns two messages $X = (X_0, \ldots, X_{a-1})$ and $X' = (X'_0, \ldots, X'_{a'-1})$ in $\mathcal{X}^{\leq \ell}$. Assuming $\mathcal{A}$ is successful, then $X \neq X'$ and $F_K^{\oplus}(X) = F_K^{\oplus}(X')$, where $K$ is the key drawn at random by the UNP challenger. Assume *wlog* that $a \leq a'$, and let $J$ denote the set of integers $j \in [\![0, a-1]\!]$ such that $X_j \neq X'_j$. Then $F_K^{\oplus}(X) = F_K^{\oplus}(X')$ is equivalent to

$$\left( \bigoplus_{j \in J} F_K(X_j, \langle j \rangle_c) \oplus F_K(X'_j, \langle j \rangle_c) \right) \oplus \bigoplus_{j=a}^{a'-1} F_K(X'_j, \langle j \rangle_c) = 0^n, \qquad (1)$$

---

[6] To avoid confusion, we warn that in his security proof of Protected Counter Sum, Bernstein uses "unpredictable" to mean pseudorandom, see beginning of [Ber99, Section 2].

where the second summation is empty if $a = a'$. Note that all $F$ inputs appearing in (1) are distinct. Then $\mathcal{B}$ proceeds as follows. If $a < a'$, then it queries all inputs appearing in (1) except $(X'_{a'-1}, \langle a' - 1 \rangle_c)$ and returns

$$\left( \bigoplus_{j \in J} F_K(X_j, \langle j \rangle_c) \oplus F_K(X'_j, \langle j \rangle_c) \right) \oplus \bigoplus_{j=a}^{a'-2} F_K(X'_j, \langle j \rangle_c)$$

as its guess for $F_K(X'_{a'-1}, \langle a' - 1 \rangle_c)$. (Note that this expression equals $0^n$ if $J = \emptyset$ and $a' = a + 1$.) If $a = a'$, then $J \neq \emptyset$ as otherwise this would imply $X = X'$. Let $j_0 = \max(J)$. Then $\mathcal{B}$ queries all inputs appearing in (1) except $(X'_{j_0}, \langle j_0 \rangle_c)$ and returns

$$\left( \bigoplus_{j \in J \setminus \{j_0\}} F_K(X_j, \langle j \rangle_c) \oplus F_K(X'_j, \langle j \rangle_c) \right) \oplus F_K(X_{j_0}, \langle j_0 \rangle_c)$$

as its guess for $F_K(X'_{j_0}, \langle j_0 \rangle_c)$.

Clearly, $\mathcal{B}$ succeeds when $\mathcal{A}$ succeeds, makes at most $2\ell$ oracle queries, and runs in time at most $\zeta + \alpha\ell$ for some small constant $\alpha$. $\mathcal{B}$'s additional computations besides running $\mathcal{A}$ only consist in equality checking and xoring on $n$-bit blocks, hence $\alpha$ only depends on the computation model but not on $F$. $\qquad\square$

Theorem 4 establishes that $F^\oplus$ is a computational AU in a constructive way: given an AU adversary $\mathcal{A}$ against $F^\oplus$, the proof describes an explicit UNP adversary $\mathcal{B}$ against $F$ using $\mathcal{A}$ as a black box. On the other hand, one can easily adapt the proof to show that $F^\oplus$ is a *statistical* AU in a non-constructive way, as shown in the following theorem. A similar constructive/non-constructive dichotomy for cAU/sAU security was proved by Bellare for the cascade construction applied to a PRF [Bel06, Lemma 3.1].

**Theorem 5.** *Let $F : \mathcal{K} \times \mathcal{X} \times \{0,1\}^c \to \{0,1\}^n$ be a keyed function with key space $\mathcal{K}$, domain $\mathcal{X} \times \{0,1\}^c$, and range $\{0,1\}^n$. Let $\delta(\ell) := \max_{\mathcal{A}} \{\mathsf{Adv}_F^{\mathrm{unp}}(\mathcal{A})\}$, where the maximum is over all adversaries making at most $2\ell$ oracle queries and running in time at most $\alpha\ell$ for some small constant $\alpha$ that only depends on the computation model (but not on $F$). Then $F^\oplus$ is $\delta$-sAU.*

*Proof.* Assume towards a contradiction that $F^\oplus$ is not $\delta$-sAU, i.e., there exists $\ell$ and $(X, X') \in (\mathcal{X}^{\leq \ell})^2$ such that $X \neq X'$ and

$$\delta' := \Pr\left[K \leftarrow_\$ \mathcal{K} : F_K^\oplus(X) = F_K^\oplus(X')\right] > \delta(\ell).$$

Define $\mathcal{B}'$ as the UNP adversary having $X$ and $X'$ hardwired in its code[7] and behaving exactly as $\mathcal{B}$ from the proof of Theorem 4 (except it does not have to run any AU adversary $\mathcal{A}$ to obtain $X$ and $X'$). Then $\mathcal{B}'$ makes at most $2\ell$ oracle queries, runs in time $\alpha\ell$ for some small constant $\alpha$, and wins the UNP game with probability $\delta' > \delta(\ell)$, contradicting the definition of $\delta$. $\qquad\square$

---

[7] Note that one cannot construct adversary $\mathcal{B}'$ above by running an sAU-adversary $\mathcal{A}$ against $F^\oplus$ and waiting for it to output $(X, X')$ since it would be impossible to upper bound the running time of $\mathcal{B}'$ (hence the non-constructiveness).

THE CONSTRUCTION. We now turn to the actual TBC-based hash function used to instantiate NaT in our nAE modes. Let $k$, $t$, and $n$ be positive integers and $E \in \text{TBC}(k, t, n)$ be a tweakable block cipher. Let $c$ be a positive integer such that $c \leq t - 3$ and let $m \coloneqq t - c - 3$ and $L \coloneqq 2^c \cdot (m + n)$.[8] We define the keyed hash function $H[E]$ with key space $\{0,1\}^k$, domain $\{0,1\}^{\leq L} \times \{0,1\}^{\leq L}$, and range $\{0,1\}^n$ as

$$H[E]_K(A, M) \coloneqq \begin{cases} E_K^{\langle 4 \rangle_3 | 0^{t-3}}(0^n) & \text{if } (A, M) = (\epsilon, \epsilon), \\ H[E]_K^0(A) \oplus H[E]_K^2(M) & \text{otherwise,} \end{cases} \tag{2}$$

where for $i \in \{0, 2\}$ and $X \in \{0,1\}^{\leq L}$ parsed as $X_0 \| \cdots \| X_{\ell-1} \xleftarrow{m+n} X$,

$$H[E]_K^i(X) \coloneqq \begin{cases} 0^n & \text{if } X = \epsilon, \\ \bigoplus_{j=0}^{\ell-1} E_K^{\langle i \rangle_3 \| \langle j \rangle_c \| \lceil X_j \rceil_m}(\lfloor X_j \rfloor_n) & \text{if } X \neq \epsilon, |X_{\ell-1}| = n, \\ \bigoplus_{j=0}^{\ell-2} E_K^{\langle i \rangle_3 \| \langle j \rangle_c \| \lceil X_j \rceil_m}(\lfloor X_j \rfloor_n) \\ \qquad \oplus E_K^{\langle i+1 \rangle_3 \| \langle \ell-1 \rangle_c \| \lceil X_{\ell-1}^* \rceil_m}(\lfloor X_{\ell-1}^* \rfloor_n) & \text{if } X \neq \epsilon, |X_{\ell-1}| < n, \end{cases}$$

where $X_{\ell-1}^* = \text{ozp}_{m+n}(X_{\ell-1})$. Define the length function for $(A, M) \in \{0,1\}^{\leq L} \times \{0,1\}^{\leq L}$ as $\text{len}(A, M) \coloneqq |A|_{n+m} + |M|_{n+m}$.

We establish that $H$ is a sAU hash function in the following theorem. The proof follows the one of Theorem 4 and Theorem 5 and is deferred to Appendix D.

**Theorem 6 (sAU security of $H$).** *Let $k$, $t$, and $n$ be positive integers, $E \in \text{TBC}(k, t, n)$ be a tweakable block cipher, and $c$ and $m$ be integers such that $t = m + c + 3$. Let $\delta(\ell) \coloneqq \max_{\mathcal{A}}\{\text{Adv}_E^{\text{unp}}(\mathcal{A})\}$, where the maximum is over all adversaries making at most $2\ell$ oracle queries and running in time at most $\alpha\ell$ for some small constant $\alpha$ that only depends on the computation model (but not on $E$). Then $H[E]$ is $\delta$-sAU.*

CONJECTURED UNPREDICTABILITY OF TBCS. In view of Theorem 6, it is interesting to contrast unpredictability and pseudorandomness in order the see if anything has been gained compared to previous proofs of AU-security for $H[E]$ and variants based on (T)PRP assumptions. As mentioned in the introduction, there exist non-uniform attacks against against any pseudorandom function with key length $k$ achieving advantage $2^{-k/2}$ in constant time and queries [DTT10, BL13]. On the other hand, no such attack is known for unpredictability. For adversaries running in time at most $\alpha\ell$ and making at most $2\ell$ queries, a reasonable conjecture is that $\max_{\mathcal{A}}\{\text{Adv}_E^{\text{unp}}(\mathcal{A})\}$ is close to $\alpha\ell/2^k + 1/(2^n - 2\ell)$, where the first term captures the success probability of exhaustive key search and the second term the success probability of a random guess among the unqueried values. In our specific case, this can even be pushed to $\ell/2^k + 1/(2^n - 2)$ as it can be

---

[8] As usual, $c$ denotes the counter length, while $m$ denotes the number of message input bits that can fit in the tweak in addition to the counter and a 3-bit prefix.

noted that adversaries constructed in the proof of Theorem 6 make at most 2 queries per tweak. Combined with Theorem 6, we obtain the following conjecture justifying the assumption regarding $\delta$ in Theorem 3.

*Conjecture 1.* Let $E \in \mathrm{TBC}(k, t, n)$ be a "secure" tweakable block cipher. Then there exists small absolute constants $\alpha$ and $\beta$ such that $H[E]$ is $\delta$-sAU with $\delta(\ell) = \alpha\ell/2^k + \beta/2^n$.

UNIFORMITY OF $H$. We note that the proof of Theorem 6 can easily be adapted to prove that $H[E]$ restricted to inputs of length at most $\ell$ is $\gamma$-uniform for $\gamma = \max_{\mathcal{A}}\{\mathsf{Adv}_E^{\mathrm{unp}}(\mathcal{A})\}$ where the maximum is over all adversaries making at most $\ell$ oracle queries and running in time at most $\alpha\ell$ for some small constant $\alpha$. Note that we cannot simply let $H[E]_K(\epsilon, \epsilon) = 0^n$ as this would break the uniformity property required for the proof of Theorem 3. In particular, the adversary could provoke bad condition (C-3) with probability close to 1 by making $2^{k/2}$ queries $\mathrm{TAG}(i, N, (\epsilon, \epsilon))$ for distinct users $i$ and some fixed nonce $N$ and $2^{k/2}$ queries $\mathrm{IC}(K, 0^{t-\nu}\|N, 0^n)$.

## 5 Authenticated Encryption Modes

Given an nPRMAC and an nivE scheme, one can obtain an nAE scheme by combining them to through the NSIV composition method [PS16], a variant of SIV [RS06]. Formally, given an AD space $\mathcal{A}$ and a message space $\mathcal{M}$, let $\Pi_{\mathrm{mac}} = (\mathcal{K}_1, \mathcal{N}, \mathcal{U}, \mathcal{V}, \mathsf{Tag})$ be an nPRMAC with $\mathcal{U} = \mathcal{A} \times \mathcal{M}$ and let $\Pi_{\mathrm{enc}} = (\mathcal{K}_2, \mathcal{R}, \mathcal{M}, \mathcal{C}, \mathsf{Enc}, \mathsf{Dec})$ be an nivE scheme. Let $\mathsf{Conv} \colon \mathcal{V} \to \mathcal{R}$ be a regular function. We define the nAE scheme $\mathsf{NSIV}[\Pi_{\mathrm{mac}}, \Pi_{\mathrm{enc}}]$ with key space $\mathcal{K}_1 \times \mathcal{K}_2$, nonce space $\mathcal{N}$, AD space $\mathcal{A}$, message space $\mathcal{M}$, ciphertext space $\mathcal{C}$, and tag space $\mathcal{V}$ as specified in Figure 6. As shown in [PS16, JNPS21] for the single user setting, assuming $\Pi_{\mathrm{mac}}$ is nPRMAC-secure and $\Pi_{\mathrm{enc}}$ is nivE-secure, then $\mathsf{NSIV}[\Pi_{\mathrm{mac}}, \Pi_{\mathrm{enc}}]$ is nAE-secure.

Given three TBCs $E_{\mathrm{in}}$, $E_{\mathrm{out}}$, and $E_{\mathrm{enc}}$ with respective key spaces $\mathcal{K}_{\mathrm{in}}$, $\mathcal{K}_{\mathrm{out}}$, and $\mathcal{K}_{\mathrm{enc}}$, any encryption scheme from the GCTR family (Section 3) based on $E_{\mathrm{enc}}$ can be combined through the NSIV composition method with the NaT MAC scheme (Section 4) based on $E_{\mathrm{in}}$ for hashing and $E_{\mathrm{out}}$ for encrypting the hash in order to obtain a TBC-based nAE scheme with key space $\mathcal{K}_{\mathrm{in}} \times \mathcal{K}_{\mathrm{out}} \times \mathcal{K}_{\mathrm{enc}}$. However, using tweak domain separation, one can save on key material by letting $E_{\mathrm{out}} = E_{\mathrm{enc}}$ and using the same key both for tag finalization (i.e., encrypting the hash) in NaT and in the encryption mode, separating the two types of calls by using a different tweak prefix. Moreover, it is possible to use a single TBC $E = E_{\mathrm{in}} = E_{\mathrm{enc}}$, again by using tweak domain separation, as long as the hashing key and the encryption key are independent. This way, we obtain a family of nAE schemes that we call *generic NSIV* (GNSIV) based on a TBC $E$ (or two TBCs $E_{\mathrm{in}}$ and $E_{\mathrm{enc}}$) and parameterized by the pair of functions $(f_T, f_X)$ used to instantiate the encryption part as $\mathsf{GCTR}[E/E_{\mathrm{enc}}, f_T, f_X]$.

We now describe the GNSIV family more formally, focusing on the single-TBC case for simplicity and study its nAE security. Let $k$, $t$, and $n$ be positive integers

18

$$
\begin{array}{ll}
\underline{\mathsf{Enc}_{(K_1,K_2)}(N,A,M)} & \underline{\mathsf{Dec}_{(K_1,K_2)}(N,A,C,V)} \\[4pt]
V \coloneqq \mathsf{Tag}_{K_1}(N,(A,M)) & R \coloneqq \mathsf{Conv}(V) \\[2pt]
R \coloneqq \mathsf{Conv}(Y) & M \coloneqq \Pi_{\mathrm{enc}}.\mathsf{Dec}_{K_2}(N,R,C) \\[2pt]
C \coloneqq \Pi_{\mathrm{enc}}.\mathsf{Enc}_{K_2}(N,R,M) & V' \coloneqq \mathsf{Tag}_{K_1}(N,(A,M)) \\[2pt]
\textbf{return } (C,V) & \textbf{if } V = V' \textbf{ then return } M \textbf{ else return } \perp
\end{array}
$$

**Fig. 6.** The encryption and decryption algorithms for the NSIV construction, combining an nPRMAC $\Pi_{\mathrm{mac}}$ and an nivE scheme $\Pi_{\mathrm{enc}}$ into an nAE scheme $\mathsf{NSIV}[\Pi_{\mathrm{mac}}, \Pi_{\mathrm{enc}}]$.

and let $E \in \mathrm{TBC}(k,t,n)$. Let $\nu$ and $c$ be positive integers[9] and let $f_T$ and $f_X$ be two functions from $\{0,1\}^\nu \times \{0,1\}^n \times [\![0, 2^c-1]\!]$ to $\{0,1\}^{t_2-3}$ and $\{0,1\}^n$ respectively.[10] We use a 3-bit prefix for tweak domain separation, with prefixes 0 to 4 used for hashing (see definition of $H[E]$ in Section 4.2), prefix 5 used to encrypt the hash, and prefix 6 used for message encryption. Given $E$, $f_T$, and $f_X$, the nAE scheme $\mathsf{GNSIV}[E, f_T, f_X]$ has key space $\{0,1\}^k \times \{0,1\}^k$, nonce space $\{0,1\}^\nu$, AD, message, and ciphertext spaces $\{0,1\}^{\leq 2^c n}$, tag space $\{0,1\}^n$, and algorithms Enc and Dec as defined in Figure 7.

Unfortunately, the generic composition theorem from [PS16, JNPS21] no longer applies due to key reuse for both tag finalization and encryption and our use of the ideal cipher model. However, in Appendix E, we show that the security bound for $\mathsf{GNSIV}[E, f_T, f_X]$ (with a generic hash function in place of $H[E]$) is essentially the sum of the bounds of NaT and the specific GCTR variant.

## 6  TBC Instantiations

In this section, we propose new tweakable block ciphers with large tweak length that can be used in the modes described previously. Our analysis will focus on TBCs derived from the Superposition Tweakey (STK) framework [JNP14] that we recall in Section 6.1. In Section 6.2, we first discuss the SKINNYe-64/256 proposal from [NSS20], a variant of the original SKINNY family of TBCs [BJK+16] with a 256-bit long tweakey (4 times larger than the block length), and explain why this variant is flawed. Then, we propose in Section 6.3 new instantiations of tweakable block ciphers with wide tweak spaces based on Deoxys-TBC [JNPS16], and discuss similar extensions for SKINNY-64 and SKINNY-128 [BJK+16]. We give security arguments for these new constructions in Section 6.4.

---

[9] We use the same counter length both for hashing and encryption for simplicity, but we could use two different lengths if needed.

[10] The random value length $r$ is equal to $n$ here as $R$ is the output of $\mathsf{NaT}[H[E_1], E_2]$ which is $n$-bit.

$$\begin{array}{ll}
\mathsf{GNSIV}[E, f_T, f_X].\mathsf{Enc}_{K_1, K_2}(N, A, M) & \mathsf{GNSIV}[E, f_T, f_X].\mathsf{Dec}_{K_1, K_2}(N, A, C, V)
\end{array}$$

$W \leftarrow H[E]_{K_1}(A, M)$

$V \leftarrow E_{K_2}^{\langle 5 \rangle_3 \| 0^{t-3-\nu} \| N}(W)$

**if** $M = \epsilon$ **then return** $(\epsilon, V)$

$R \leftarrow V$

$M_0 \| \cdots \| M_{\ell-1} \xleftarrow{n} M$

**for** $j := 0 \ldots \ell - 1$ **do**

  $T_j := f_T(N, R, j)$

  $X_j := f_X(N, R, j)$

**for** $j := 0 \ldots \ell - 2$ **do**

  $C_j := M_j \oplus E_{K_2}^{\langle 6 \rangle_3 \| T_j}(X_j)$

$/\!/$ last block might be incomplete

$C_{\ell-1} := M_{\ell-1} \oplus \left\lceil E_{K_2}^{\langle 6 \rangle_3 \| T_{\ell-1}}(X_{\ell-1}) \right\rceil_{|M_{\ell-1}|}$

**return** $(C_0 \| \cdots \| C_{\ell-1}, V)$

---

**if** $C \neq \epsilon$ **then**

  $R \leftarrow V$

  $C_0 \| \cdots \| C_{\ell-1} \xleftarrow{n} C$

  **for** $j := 0 \ldots \ell - 1$ **do**

    $T_j := f_T(N, R, j)$

    $X_j := f_X(N, R, j)$

  **for** $j := 0 \ldots \ell - 2$ **do**

    $M_j := C_j \oplus E_{K_2}^{\langle 6 \rangle_3 \| T_j}(X_j)$

  $/\!/$ last block might be incomplete

  $M_{\ell-1} := C_{\ell-1} \oplus \left\lceil E_{K_2}^{\langle 6 \rangle_3 \| T_{\ell-1}}(X_{\ell-1}) \right\rceil_{|C_{\ell-1}|}$

  $M \leftarrow M_0 \| \cdots \| M_{\ell-1}$

**else** $M \leftarrow \epsilon$

$W \leftarrow H[E]_{K_1}(A, M)$

$V' \leftarrow E_{K_2}^{\langle 5 \rangle_3 \| 0^{t-3-\nu} \| N}(W)$

**if** $V = V'$ **then return** $M$ **else return** $\perp$

**Fig. 7.** The encryption and decryption algorithms of the $\mathsf{GNSIV}[E, f_T, f_X]$ nAE scheme based on a TBC $E$ with $H[E]$ as defined by (2) in Section 4.2.

## 6.1 The STK Framework

The TWEAKEY framework was put forward in 2014 by Jean *et al.* [JNP14]. It blends together the key and the tweak input of a TBC in a so-called *tweakey*, allowing to design "tweakey schedules" for key-alternating ciphers using tools from related-key attacks against (traditional) block ciphers. A more specialized version of TWEAKEY, called the Superposition Tweakey (STK) framework, consists in splitting the $p \times n$ bits of tweakey material into $p$ words of $n$ bits, and update each word linearly and independently. The class of tweakable block ciphers following this construction having a ratio tweakey size to block size equals to $p$ are said to belong to **TK**$p$. Concrete instantiations were given in [JNP14] for $p = 2$ (**TK2**) and $p = 3$ (**TK3**) in the forms of two submissions to the CAESAR competition: Deoxys-TBC and Joltik-TBC.

The STK construction moreover specifies the type of update: in each $n$-bit word, the 16 elementary $c$-bit cells are permuted with the same permutation $h$, and then, each cell from the $i$-th word is seen as an element of $\mathbb{F}_{2^c}$ and gets multiplied by a constant $\alpha_i$, $i \in [\![0, p-1]\!]$ (see Figure 8). Finally, each $n$-bit subtweakey to be integrated in the key-alternating structure is simply taken as the XOR of the $n$-bit words.

The important consequence of this design that will matter for the rest of the section is that each fixed position $i \in [\![0, 15]\!]$ in the $p$ tweakey words, seen as a $p$-element vector $\mathbf{x}^i = (x_0^i, \ldots, x_{p-1}^i)$, is extended by the tweakey schedule to
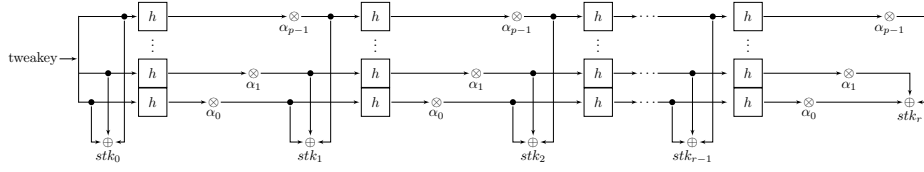
**Fig. 8.** The original STK construction from [JNP14].

an $(r + 1)$-element vector $\mathbf{y}^i = (y_0^i, \ldots, y_r^i)$, containing the $r + 1$ elements to be integrated in one position of the internal state by the $r + 1$ subtweakey additions over $r$ rounds.

With a small generalization, we can replace the multiplications by $\alpha_0, \ldots,$ $\alpha_{p-1} \in \mathbb{F}_{2^c}$ by linear functions $f_0, \ldots, f_{p-1}$ over $\mathbb{F}_2$ so that for $k \in [\![0, r]\!]$, one has $y_k^i = \bigoplus_{j=0}^{p-1} f_j^k(x_j^i)$. This simple construction allows to analyze the linear expansion $\mathbf{x}^i \to \mathbf{y}^i$ to find a lower bound on the element-wise Hamming weight of the expanded vector $\mathbf{y}^i$. One can then feed this bound to MILP solvers to help them assess resistance against linear or differential cryptanalysis, regardless of the number of **TK** words.

### 6.2  SKINNYe-64/256**: a Flawed TK4 Extension of** SKINNY

Several tweakable block ciphers have been designed following the STK framework, notably SKINNY [BJK+16]. SKINNY is now considered a well-studied primitive, yet only **TK1**, **TK2** and **TK3** variants have been proposed by the designers. To address the need for a **TK4** variant, Naito, Sasaki and Sugawara [NSS20] proposed a **TK4** variant of SKINNY-64 with lightweight applications in mind. They named their proposal SKINNYe-64/256.

Mimicking the **TK3** design of SKINNY-64/192, the authors naturally extended the tweakey size by $n = 64$ bits by choosing a new 4-bit function $f_3$ and increasing the number of rounds to 44. While in SKINNY-64/192 $f_0$ is the identity function and $f_1$, $f_2$ are two 4-bit LFSRs, in SKINNYe-64/256 the chosen transformation[11] $f_3$, although being a linear permutation over $\mathbb{F}_2$, is not an LFSR (as claimed in [NSS20])

However, we claim that this transformation fails to meet the assumptions required by the security analysis performed in [BJK+16] and reused in [NSS20]. Namely, to allow a computer-aided security evaluation of STK-based ciphers like SKINNY with respect to differential cryptanalysis, its tweakey schedule must verify specific linear properties. More precisely, abstracting the inner workings of the tweakey schedule away, one must guarantee that the incoming round subtweakeys $skt_i$ (see Figure 8) have a (lower) bounded number of active differences; or stated differently, that the number of differences cancellations made possible by the

---

[11] The transformation is $f_3 : (x_3 \,\|\, x_2 \,\|\, x_1 \,\|\, x_0) \longrightarrow (x_2 \,\|\, x_1 \,\|\, x_2 \oplus x_0 \,\|\, x_3 \oplus x_2 \oplus x_1)$.

21

**Table 2.** For starting difference $(\texttt{0x1}, \texttt{0x4}, \texttt{0x0}, \texttt{0x5})$, the tweakey schedule of SKINNYe-64/256 outputs very sparse binary subtweakey differences.

| Round | TK1 | TK2 | TK3 | TK4 | stk | Round | TK1 | TK2 | TK3 | TK4 | stk |
|-------|-----|-----|-----|-----|-----|-------|-----|-----|-----|-----|-----|
| 0 | 0x1 | 0x4 | 0x0 | 0x5 | 0x0 | 8 | 0x01 | 0x7 | 0x0 | 0x6 | 0x0 |
| 1 | 0x1 | 0x9 | 0x0 | 0x9 | 0x1 | 9 | 0x01 | 0xf | 0x0 | 0xe | 0x0 |
| 2 | 0x1 | 0x3 | 0x0 | 0x3 | 0x1 | 10 | 0x01 | 0xe | 0x0 | 0xf | 0x0 |
| 3 | 0x1 | 0x6 | 0x0 | 0x7 | 0x0 | 11 | 0x01 | 0xc | 0x0 | 0xd | 0x0 |
| 4 | 0x1 | 0xd | 0x0 | 0xc | 0x0 | 12 | 0x01 | 0x8 | 0x0 | 0x8 | 0x1 |
| 5 | 0x1 | 0xa | 0x0 | 0xa | 0x1 | 13 | 0x01 | 0x1 | 0x0 | 0x1 | 0x1 |
| 6 | 0x1 | 0x5 | 0x0 | 0x4 | 0x0 | 14 | 0x01 | 0x2 | 0x0 | 0x2 | 0x1 |
| 7 | 0x1 | 0xb | 0x0 | 0xb | 0x1 | 15 | 0x01 | 0x4 | 0x0 | 0x5 | 0x0 |

linearity of the tweakey schedule are (upper) bounded. In the original SKINNY paper, the MILP models correctly represent the cipher when the number of cancellations does not exceed $p - 1$ for each nibble position in the tweakey state for $\textbf{TK}p$, $p \in \{2, 3\}$, and when the number of rounds covered does not exceed 30.

By analyzing the effect of $f_3$ on that number of cancellations for SKINNYe-64/256 ($\textbf{TK4}$, $p = 4$), we discover that it can be far higher than $p - 1 = 3$ for a well-chosen tweakey difference. For instance, the difference $(\texttt{0x1}, \texttt{0x4}, \texttt{0x0}, \texttt{0x5})$ applied to the same position in the four tweakey words yields subtweakeys where only 7 are nonzero across 14 rounds (see Table 2). More surprisingly, the joint effect of the linear transformations is such that each of the 7 active subtweakeys contain a single active nible with difference equals to 1.

As a consequence, one can conclude that the security bounds computed in [NSS20, Table 2] from MILP optimization are incorrect: much sparser differential characteristics are likely to exist. We leave finding such characteristics and possible matching differential attacks out of the scope of this paper.

### 6.3 Extensions to TK4 and TK5

We now address the problem of designing a TBC with a large tweakey space within the STK framework. From a high-level point of view, what we want to design shares a lot of similarities with a linear code with bounded minimal distance. Indeed, an input vector of differences should expand to a vector containing as many nonzero differences as possible. Getting back to the original design, we show below that we can take advantage of results from coding theory to construct, under some assumptions, structured linear codes over $\mathbb{F}_{2^c}$ that can directly be used as tweakey schedules. Another option that we also consider below are transformations linear over $\mathbb{F}_2$.

LINEAR CODE OVER $\mathbb{F}_{2^c}$. In the original STK construction, the authors suggested to use linear functions $f_j : x \to \alpha_j\, x$ over $\mathbb{F}_{2^c}$, for constants $\alpha_i \in \mathbb{F}_{2^c}$. We revisit the approach and show that we can easily derive TBC with almost arbitrarily large tweakey space.

To build a TBC with $p$ words of tweakey, we can rely on a Reed-Solomon linear code constructed using a Vandermonde matrix

$$\mathbf{V}(\beta_0,\ldots,\beta_{p-1}) = \begin{bmatrix} 1 & \cdots & 1 & \cdots & 1 \\ \beta_0 & \cdots & \beta_k & \cdots & \beta_r \\ \vdots & & \vdots & & \vdots \\ \beta_0^{p-1} & \cdots & \beta_k^{p-1} & \cdots & \beta_r^{p-1} \end{bmatrix}$$

as generator matrix for distinct $\beta_0,\ldots,\beta_{p-1} \in \mathbb{F}_{2^c}$. Indeed, the code generated by the $p \times (r+1)$ generator matrix is known to be MDS. Hence, its minimal distance is $(r+1)-p+1$, which means that the number of cancellations in the $r+1$ outputs elements ($r$ rounds) is upper bounded by $p-1$.

Let $\alpha_j \in \mathbb{F}_{2^c}$ be the multiplication constants in the $j$-th tweakey word. To map the Vandermonde matrix structure to the STK construction, we can set $\alpha_j = \alpha^j$ for an element $\alpha \in \mathbb{F}_{2^c}$ with order at least $r$. Indeed, we then have the $p \times (r+1)$ Vandermonde matrix

$$\mathbf{V}(\alpha^0,\ldots,\alpha^{p-1}) = \begin{bmatrix} 1 \cdots & 1 & \cdots & 1 \\ 1 \cdots & \alpha^{k-1} & \cdots & \alpha^r \\ 1 \cdots & \alpha^{2(k-1)} & \cdots & \alpha^{2r} \\ \vdots & \vdots & & \vdots \\ 1 \cdots & \alpha^{(p-1)(k-1)} & \cdots & \alpha^{(p-1)r} \end{bmatrix} \tag{3}$$

describing the tweakey schedule, with all $\alpha^k$ being distinct, $k \in [\![0, r]\!]$, thanks to the condition on the order of $\alpha$.

This construction therefore allows to construct a $\mathbf{TK}p$ TBC with security arguments for $\max_{e \in \mathbb{F}_{2^c}}(\mathbf{ord}(e)) - 1$ rounds, which can be as high as $2^c - 2$ rounds by using any primitive element as $\alpha$.

As concrete instantiations, we propose Deoxys-TBC-512 ($\mathbf{TK4}$) with $\alpha = x \in \mathbb{F}_{2^8}$ defined by the AES polynomial $x^8 + x^4 + x^3 + x + 1$. Informally, this results in multiplications by 1, 2, 4, and 8 for the four tweakey words. Similarly, we naturally propose Deoxys-TBC-640 ($\mathbf{TK5}$) with multiplications by 1, 2, 4, 8 and 16 for the five words. We follow the rationale for the number of rounds of $10 + 2p$ used in Deoxys-TBC for $\mathbf{TK}p$, so we use 18 rounds for Deoxys-TBC-512 and 20 rounds for Deoxys-TBC-640. The round function is left unchanged.

We note that the order of the chosen $\alpha \in \mathbb{F}_{2^8}$ in both cases is not maximal and equals 51. We could have chosen $x + 1$ which is primitive, but first, we do not need security guarantees for more than 50 rounds (as Deoxys primitives stop at 20 rounds), and second, implementation-wise, it is more efficient to multiply by 2 than by 3.

About SKINNY, we do not formally propose new variants, but we note that the same reasoning would apply to extend the tweakey space.

*Remark 1.* When $c = 4$ (for instance SKINNY-64), we note that the maximal order of the field elements is $2^c - 1 = 15$. Consequently, if ones relies on the Vandermonde structure with field multiplications to build the tweakey schedule,

one cannot bound the number of cancellations for more that 14 rounds (i.e., 15 consecutive subtweakeys).

LINEAR OVER $\mathbb{F}_2$. To avoid the field multiplications described before, we can choose the functions $f_j$ as being linear over $\mathbb{F}_2$. In practice, the original Deoxys variants as well as all SKINNY TBCs use this strategy, with LFSRs with cheap update functions (a few XORs).

Theoretically, one could describe the multiplications in $\mathbb{F}_{2^c}$ as linear permutations over $\mathbb{F}_2$, but there are potentially more interesting $\mathbb{F}_2$-linear mappings (LFSRs being a subset of them) than multiplications in $\mathbb{F}_{2^c}$. However, one significant drawback is the absence of constructive method: one cannot rely anymore on a structured linear code to theoretically bound the number of cancellations. What has been done in Deoxys and SKINNY is an exhaustive search to derive this bound experimentally, but this can only work for low values of $p$.

To extend SKINNY-64 to get a **TK4** variant, we can for instance use the 4-bit linear permutation: $(x_3 \,\|\, x_2 \,\|\, x_1 \,\|\, x_0) \to (x_1 \,\|\, x_0 \,\|\, x_2 \oplus x_3 \,\|\, x_1 \oplus x_2)$. Using this transformation as $f_3$ instead of the one used in SKINNYe-64/256 would patch the mistake from [NSS20]. We indeed verified experimentally that the number of cancellations match the requirements of the STK approach applied to SKINNY.

### 6.4 Security Analysis

In this section, we adapt the elementary security analysis conducted in [JNPS16] to the new variants we proposed in the previous section based on field multiplication. As our proposed Deoxys-TBC variants do not change the overall design rationale, we expect all the security arguments provided in [JNPS16] to hold; we simply recompute the bounds for the number of differentially active Sboxes. As noted previously, an MILP-based analysis of the STK construction abstracts the actual tweakey schedule away and simply relies on the number of cancellations in the incoming subtweakeys for $r$ rounds. Depending on the type of the update, $r$ can vary: it can for instance be as large at 50 for Deoxys for field multiplications in $\mathbb{F}_{2^8}$, or lower when LFSRs are used.

By looking at the actual 8-bit LFSRs used in Deoxys, we remark that they have cycles of length 15 (even though most of the cycles are of length 30). Consequently, it is sufficient to study the element-wise Hamming weight of the expanded vector over one cycle, for $r - 1 = 14$ consecutive rounds so that there are $r = 15$ expanded elements (before they cycle).

However, in most of the security analyses of Deoxys we are aware of, e.g., [JNPS16, CHP$^+$17], the authors encode in the MILP model a lower bound of the number of cancellations for $r = 15$ rounds. Due to this short cycle, considering 15 rounds introduces an off-by-one error as one counts twice one element in the sequence of 16 elements.[12] Consequently, the number of cancellations of $p - 1$ for **TK**$p$ over 15 rounds claimed in [JNPS16] should be $p - 1$ for 14 rounds (or the

---

[12] We note that the situation is the same as described in the remark on the field multiplication in $\mathbb{F}_{2^4}$ due to the size of its multiplicative subgroup.

**Table 3.** Lower bounds for the number of differentially active Sboxes for the structure used in Deoxys-TBC. We report the two original designs from [JNPS16] as well as our new **TK4** and **TK5** variants, with two types of linear update in the tweakey schedule: either an LFSR or a field multiplication in $\mathbb{F}_{2^8}$ ("mul").

| | Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | ref. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **TK2** | LFSR | 0 | 0 | 1 | 5 | 9 | 12 | 16 | 19 | 23 | 26 | 29 | 32 | 35 | 38 | - | - | - | - | - | - | [JNPS16] |
| **TK3** | LFSR | 0 | 0 | 0 | 1 | 4 | 8 | 10 | 14 | 18 | 21 | 24 | 28 | 31 | 35 | 37 | 41† | - | - | - | - | [JNPS16] |
| **TK4** | LFSR | 0 | 0 | 0 | 0 | 1 | 3 | 6 | 10 | 12 | 14 | 19 | 23 | 26 | 30 | 33 | 37 | 38 | 39 | - | - | this paper |
| **TK4** | mul | 0 | 0 | 0 | 0 | 1 | 3 | 6 | 10 | 12 | 14 | 19 | 23 | 26 | 30 | 33 | 37 | 39 | 43 | - | - | this paper |
| **TK5** | LFSR | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 6 | 8 | 12 | 14 | 17 | 21 | 25 | 28 | 31 | 33 | 34 | 36 | 38 | this paper |
| **TK5** | mul | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 6 | 8 | 12 | 14 | 17 | 21 | 25 | 28 | 31 | 35 | 38 | 42 | 45 | this paper |

† [CHP+17] indicated 45.

looser bound $p$ for 15 rounds). The MILP model from [CHP+17] suffers from this off-by-one error, however, by chance, the Deoxys variants studied by the authors only uses up to 16 rounds (17 consecutive subtweakeys), which makes only the results on 16 rounds incorrect.

To accommodate this, and to be able to model more than 14 consecutive rounds, we slightly tweak the MILP model used for Deoxys based on LFSRs. In our model for **TK**$p$, for $r \leq 14$ rounds, if the Boolean variable $\mathtt{stk}_j[i]$ represents the activity of the $i$-th byte of the $j$-th subtweakey, $i, j \in [\![0, 14]\!] \times [\![0, r]\!]$, the linear inequality $\sum_j \mathtt{stk}_j[i] \geq \min(r+1, 15) - (p-1)$ is added to the MILP model for each window of 14 consecutive rounds.[13] For instance, to model 18 rounds, we introduce five inequalities, starting at Rounds 0, 1, 2, 3 and 4. This ensures that for any 15 consecutive subtweakeys, the linear property of the tweakey schedule is included in the MILP model.

In the case of the field multiplications in $\mathbb{F}_{2^8}$, the length of the cycle depends on the order of the element used to construct the Vandermonde matrix (3). In our case, for Deoxys-TBC-512 and Deoxys-TBC-640, we use an element with order 51, so as long as we model $r \leq 50$ rounds, we can keep a single inequality per position $i$, namely: $\sum_{j=0}^{r} \mathtt{stk}_j[i] \geq (r+1) - (p-1)$.

We report in Table 3 the results of the MILP optimizations for the different cases. Recall that Deoxys-TBC-512 corresponds to **TK4** of "mul" type, and Deoxys-TBC-640 corresponds to **TK5** of "mul" type. We include the other cases to emphasize the influence of the bounds on the modelization.

---

[13] The inequality given is simplified, as all $\mathtt{stk}_j[i]$ can be zero.

## 7 Implementations

In order to assess the performance of our new modes, we first make a general study on the expected computational cost when using Deoxys-TBC instances, focusing on GNSIV-N and GNSIV-Z. Then, we concentrate on the important case of Intel processors with `AESNI` instructions set. We compare our candidates with popular nonce-misuse resistant modes, such as AES-GCM-SIV [GLL17], ZAE [IMPS17] and SCT-2 [JNPS16, JNPS21]. For simplicity, we measure long messages, even though we note that GNSIV-N and GNSIV-Z, similarly to SCT-2, have a minimal overhead for small messages.

### 7.1 General Theoretical Performances

The encryption part of GNSIV-N and GNSIV-Z requires one TBC call per message block and this is independent of the Deoxys-TBC version used. Thus, we better use the smallest-tweak version of Deoxys-TBC (since it will use lesser AES rounds) as long as the tweak input is large enough to accommodate the mode inputs. Thus, considering an encryption key $K_2$, a nonce $N$, and a random value $R$ of 128 bits each and a counter of 125 bits (recall that we reserve 3 bits for tweak domain separation), the encryption part of GNSIV-N and GNSIV-Z can use Deoxys-TBC-384 and Deoxys-TBC-512 respectively. We note that we could consider increasing the encryption key length to 256 bits by using Deoxys-TBC-512 and Deoxys-TBC-640 instead.

Regarding the authentication part, using the same parameters, GNSIV-N and GNSIV-Z can rely on Deoxys-TBC-256 since only the counter is inserted in the tweak input of the TBC. However, using larger tweak instances of Deoxys-TBC could be useful as this extra input can directly take more associated data or message for a higher throughput. This is of course a trade-off as these Deoxys-TBC instances will use more AES rounds on the other hand.

As already observed in [IMPS17], in software with `AESNI` computing one Deoxys-TBC tweakey schedule line roughly represents a cost of 0.4 of an entire AES encryption, thus about 4 AES rounds. This is of course a very rough and relatively pessimistic estimation that is highly dependent on the actual platform on which the algorithm will be run, but it has the merit to take into account the extra cost coming from the tweakey schedule. Moreover, when measuring performances for long messages, if a tweak input in one TK position remains unchanged for many messages blocks (typically the key, the nonce, etc.) the cost for this TK is amortized since all TK parts can be computed independently in Deoxys-TBC. The same applies to a counter since its evolution is predictable and thus updating the TK values from a counter increment is very cheap.

We summarize the theoretical cost in terms of AES rounds for various GNSIV-N and GNSIV-Z instantiations in Table 4. We note that GNSIV-N instantiated with Deoxys-TBC-384 for both authentication and encryption part would correspond to the recent Deoxys-AE2 mode [JNPS21] (proposed without security proof). We further note that while our modes require two passes on the plaintext input for encryption, the decryption can be done online with a single pass. Finally, we

**Table 4.** Theoretical performance in number of AES rounds (for one 128-bit AD block and one 128-bit M block) for various GNSIV-N and GNSIV-Z modes instantiations with Deoxys-TBC versions. Notation $x$ ($y$) means that $x$ rounds of AES are required when not considering potential extra tweakey schedule cost, while $y$ takes into account the 1.4 factor that might occur depending on the platform. For AES-GCM-SIV, we consider that one POLYVAL evaluation is equivalent to 5 AES rounds [GLL17].

| Mode | Deoxys-TBC instance | | AES round cost | |
| | Hash | Enc | per AD | per M |
|---|---|---|---|---|
| GNSIV-N ($|K_2| = 128$) | Deoxys-TBC-256 | Deoxys-TBC-384 | 14 | 30 |
| | Deoxys-TBC-384 | | 8 (11.2) | 24 (27.2) |
| GNSIV-N ($|K_2| = 256$) | Deoxys-TBC-256 | Deoxys-TBC-512 | 14 | 32 |
| | Deoxys-TBC-512 | | 6 (11.8) | 24 (29.8) |
| GNSIV-Z ($|K_2| = 128$) | Deoxys-TBC-256 | Deoxys-TBC-512 | 14 | 32 |
| | Deoxys-TBC-512 | | 6 (11.8) | 24 (29.8) |
| GNSIV-Z ($|K_2| = 256$) | Deoxys-TBC-256 | Deoxys-TBC-640 | 14 | 34 |
| | Deoxys-TBC-640 | | 5 (13.7) | 25 (33.7) |
| SCT-2 [JNPS21] | Deoxys-TBC-256 | Deoxys-TBC-256 | 14 | 28 |
| ZAE [IMPS17] | Deoxys-TBC-384 | Deoxys-TBC-256 | 8 (11.2) | 22 (25.2) |
| AES-GCM-SIV-128 [GLL17] | - | - | 5 | 15 |

emphasize that our instances allow a counter with a size of almost 128 bits, in contrary to modes like AES-GCM-SIV that only handles inputs of size $2^{64}$ or less. Thus, for more accurate comparison, 64 bits of this counter could be traded for further authentication throughput improvement.

### 7.2 Software Performances with AESNI

As Deoxys-TBC is based on the AES round function and since our modes allow parallelization, they allow very efficient software implementations on the processors that support AES-accelerated instructions, and in particular AESNI.

We have implemented GNSIV-N and GNSIV-Z using Intel Intrinsics and we provide in Table 5 performance measurements on Intel Haswell processor. For completeness, we also measured performances of AES-GCM-SIV implementation[14] ("Instrinsics" ones), taking into account the key schedule.

Having to perform an update of one TK$p$ word, when $p \geq 2$, would be too costly using AESNI, because the shifts/XORs involved in the LFSR would require too many operations in comparison to nicely pipelined AESNI instructions.

Our modes perform quite well: they are generally slower of a factor around 1.6 than AES-GCM-SIV for very long messages, but closer when the message get smaller (actually on par for 64-byte messages). However, we recall that the Internet Mix is generally composed of mostly very small (smaller than

---

[14] Taken from https://github.com/Shay-Gueron/AES-GCM-SIV.

**Table 5.** Encryption benchmarks for GNSIV-N and GNSIV-Z (with Deoxys-TBC-384 in the authentication part), with comparison with AES-GCM-SIV-128, expressed in cycles per byte on `AESNI` enabled platforms (with Turbo Boost disabled) for increasing numbers of processed bytes. The key schedule is computed at each call and the loading of the bytes from the memory and storing them back to memory are included. The reported speed was taken as an average over multiple executions of the code with the same fixed message length. Naturally, smaller message or associated data sizes will lead to slower c/B performances, due to the various initialization overhead.

**Intel Haswell** (gcc v9.2)

| AD bytes | 0 | 0 | 0 | 64 | 576 | 1.5k | 64 | 576 | 1.5k | 0 | 65k | 65k |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M bytes | 64 | 576 | 1.5k | 0 | 0 | 0 | 64 | 576 | 1.5k | 65k | 0 | 65k |
| GNSIV-N ($|K_2| = 128$) | 10.10 | 3.17 | 2.55 | 8.49 | 1.95 | 1.36 | 6.13 | 2.21 | 1.82 | 2.23 | 1.03 | 1.64 |
| GNSIV-N ($|K_2| = 256$) | 10.12 | 3.25 | 2.68 | 8.49 | 1.95 | 1.36 | 6.14 | 2.24 | 1.89 | 2.38 | 1.03 | 1.71 |
| GNSIV-Z ($|K_2| = 128$) | 10.12 | 3.25 | 2.68 | 8.49 | 1.95 | 1.36 | 6.14 | 2.24 | 1.89 | 2.38 | 1.03 | 1.71 |
| GNSIV-Z ($|K_2| = 256$) | 10.15 | 3.33 | 2.82 | 8.49 | 1.95 | 1.36 | 6.15 | 2.28 | 1.95 | 2.53 | 1.03 | 1.78 |
| AES-GCM-SIV-128 | 10.00 | 2.62 | 1.77 | 8.87 | 1.90 | 1.14 | 6.29 | 1.68 | 1.23 | 1.34 | 0.71 | 1.02 |

100 bytes) packets, then some medium size (around 500 bytes) packets and finally a low proportion of maximum-size (maximum Ethernet packet payload size is 1536 bytes). Considering the added security guarantees when compared to AES-GCM-SIV, we believe our modes represent a very competitive trade-off.

In addition, we remark that our benchmarks were performed in the most possible advantageous situation for AES-GCM-SIV: a processor that has the `PCLMULQDQ` hardware accelerated instruction that allows efficient $GF(2^{128})$ multiplication on a different execution pipe than `AESNI`. Of course, our designs do not benefit from this advantage (for example having an hardware accelerated instruction to perform the tweakey schedule). On relatively cheap micro-controllers or, more importantly, in hardware, the situation will be very different. The $GF(2^{128})$ multiplication requires a large area (or alternatively many cycles). In our modes, no specific operation is used in addition to the TBC calls; they would therefore largely outperform AES-GCM-SIV even when instantiated with Deoxys-TBC. If we consider using a lightweight cipher as internal primitive, the contrast would be even clearer: our modes with SKINNY-128/512 variant would remain lightweight, while GCM-SIV with a lightweight cipher would be very large (see for example Fig. 8 of [KMY19], where AES-GCM is the worst mode for area).

# References

[ABPV21]   Elena Andreeva, Amit Singh Bhati, Bart Preneel, and Damian Vizár. 1, 2, 3, fork: Counter mode variants based on a generalized forkcipher. *IACR Trans. Symm. Cryptol.*, 2021(3):1–35, 2021.

[ADMA15]   Elena Andreeva, Joan Daemen, Bart Mennink, and Gilles Van Assche. Security of keyed sponge constructions using a modular proof approach. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 364–384. Springer, Heidelberg, March 2015.

[BBM00]    Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000.

[BBT16]    Mihir Bellare, Daniel J. Bernstein, and Stefano Tessaro. Hash-function based PRFs: AMAC and its multi-user security. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 566–595. Springer, Heidelberg, May 2016.

[BCK96]    Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th FOCS*, pages 514–523. IEEE Computer Society Press, October 1996.

[Bel06]    Mihir Bellare. New proofs for NMAC and HMAC: Security without collision-resistance. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 602–619. Springer, Heidelberg, August 2006.

[Ber99]    Daniel J. Bernstein. How to stretch random functions: The security of protected counter sums. *Journal of Cryptology*, 12(3):185–192, June 1999.

[BGR95]    Mihir Bellare, Roch Guérin, and Phillip Rogaway. XOR MACs: New methods for message authentication using finite pseudorandom functions. In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 15–28. Springer, Heidelberg, August 1995.

[BHT18]    Priyanka Bose, Viet Tung Hoang, and Stefano Tessaro. Revisiting AES-GCM-SIV: Multi-user security, faster key derivation, and better bounds. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 468–499. Springer, Heidelberg, April / May 2018.

[Bih02]    Eli Biham. How to decrypt or even substitute des-encrypted messages in $2^{28}$ steps. *Inf. Process. Lett.*, 84(3):117–124, 2002.

[BJK+16]   Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 123–153. Springer, Heidelberg, August 2016.

[BL13]     Daniel J. Bernstein and Tanja Lange. Non-uniform cracks in the concrete: The power of free precomputation. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 321–340. Springer, Heidelberg, December 2013.

[BR02]     John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 384–397. Springer, Heidelberg, April / May 2002.

[BS20]     Dan Boneh and Victor Shoup. *A Graduate Course in Applied Cryptography,
           v0.5.* 2020. Available at http://toc.cryptobook.us/book.pdf.

[BT16]     Mihir Bellare and Björn Tackmann. The multi-user security of authen-
           ticated 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, Heidelberg, August 2016.

[CHP⁺17]   Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. A
           security analysis of Deoxys and its internal tweakable block ciphers. *IACR
           Trans. Symm. Cryptol.*, 2017(3):73–107, 2017.

[CLS17]    Benoît Cogliati, Jooyoung Lee, and Yannick Seurin. New constructions
           of macs from (tweakable) block ciphers. *IACR Trans. Symm. Cryptol.*,
           2017(2):27–58, 2017.

[CS14]     Shan Chen and John P. Steinberger. Tight security bounds for key-
           alternating ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors,
           *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 327–350. Springer,
           Heidelberg, May 2014.

[DMA17]    Joan Daemen, Bart Mennink, and Gilles Van Assche. Full-state keyed
           duplex with built-in multi-user support. In Tsuyoshi Takagi and Thomas
           Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages
           606–637. Springer, Heidelberg, December 2017.

[DS09]     Yevgeniy Dodis and John P. Steinberger. Message authentication codes
           from unpredictable block ciphers. In Shai Halevi, editor, *CRYPTO 2009*,
           volume 5677 of *LNCS*, pages 267–285. Springer, Heidelberg, August 2009.

[DTT10]    Anindya De, Luca Trevisan, and Madhur Tulsiani. Time space tradeoffs
           for attacks against one-way functions and PRGs. In Tal Rabin, editor,
           *CRYPTO 2010*, volume 6223 of *LNCS*, pages 649–665. Springer, Heidelberg,
           August 2010.

[Fer02]    Niels Ferguson. Collision attacks on OCB. 2002.

[FJM14]    Pierre-Alain Fouque, Antoine Joux, and Chrysanthi Mavromati. Multi-user
           collisions: Applications to discrete logarithm, Even-Mansour and PRINCE.
           In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*,
           volume 8873 of *LNCS*, pages 420–438. Springer, Heidelberg, December
           2014.

[GLL17]    Shay Gueron, Adam Langley, and Yehuda Lindell. AES-GCM-SIV: Speci-
           fication and analysis. Cryptology ePrint Archive, Report 2017/168, 2017.
           https://eprint.iacr.org/2017/168.

[GW18]     Chun Guo and Lei Wang. Revisiting key-alternating Feistel ciphers for
           shorter keys and multi-user security. In Thomas Peyrin and Steven
           Galbraith, editors, *ASIACRYPT 2018, Part I*, volume 11272 of *LNCS*,
           pages 213–243. Springer, Heidelberg, December 2018.

[HT16]     Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-
           length extension: Exact bounds and multi-user security. In Matthew
           Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume
           9814 of *LNCS*, pages 3–32. Springer, Heidelberg, August 2016.

[HT17]     Viet Tung Hoang and Stefano Tessaro. The multi-user security of double
           encryption. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors,
           *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 381–411.
           Springer, Heidelberg, April / May 2017.

[HTT18]    Viet Tung Hoang, Stefano Tessaro, and Aishwarya Thiruvengadam. The
           multi-user security of GCM, revisited: Tight bounds for nonce randomiza-
           tion. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng

30

Wang, editors, *ACM CCS 2018*, pages 1429–1440. ACM Press, October 2018.

[IMPS17]  Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A fast tweakable block cipher mode for highly secure message authentication. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 34–65. Springer, Heidelberg, August 2017.

[JNP14]  Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and keys for block ciphers: The TWEAKEY framework. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 274–288. Springer, Heidelberg, December 2014.

[JNPS16]  Jérémy Jean, Ivica Nikolic, Thomas Peyrin, and Yannick Seurin. Deoxys v1.43. Submitted to CAESAR AE competition, 2016.

[JNPS21]  Jérémy Jean, Ivica Nikolic, Thomas Peyrin, and Yannick Seurin. The deoxys AEAD family. *Journal of Cryptology*, 34(3):31, July 2021.

[KMY19]  Elif Bilge Kavun, Hristina Mihajloska, and Tolga Yalcin. A survey on authenticated encryption – asic designer's perspective. Cryptology ePrint Archive, Report 2019/739, 2019. https://eprint.iacr.org/2019/739.

[KR11]  Ted Krovetz and Phillip Rogaway. The software performance of authenticated-encryption modes. In Antoine Joux, editor, *FSE 2011*, volume 6733 of *LNCS*, pages 306–327. Springer, Heidelberg, February 2011.

[LMP17]  Atul Luykx, Bart Mennink, and Kenneth G. Paterson. Analyzing multi-key security degradation. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 575–605. Springer, Heidelberg, December 2017.

[LPTY16]  Atul Luykx, Bart Preneel, Elmar Tischhauser, and Kan Yasuda. A MAC mode for lightweight block ciphers. In Thomas Peyrin, editor, *FSE 2016*, volume 9783 of *LNCS*, pages 43–59. Springer, Heidelberg, March 2016.

[LRW11]  Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. *Journal of Cryptology*, 24(3):588–613, July 2011.

[ML15]  Nicky Mouha and Atul Luykx. Multi-key security: The Even-Mansour construction revisited. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 209–223. Springer, Heidelberg, August 2015.

[NSS20]  Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Lightweight authenticated encryption mode suitable for threshold implementation. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 705–735. Springer, Heidelberg, May 2020.

[Pat09]  Jacques Patarin. The "coefficients H" technique (invited talk). In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *SAC 2008*, volume 5381 of *LNCS*, pages 328–345. Springer, Heidelberg, August 2009.

[PS16]  Thomas Peyrin and Yannick Seurin. Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 33–63. Springer, Heidelberg, August 2016.

[RBBK01]  Phillip Rogaway, Mihir Bellare, John Black, and Ted Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In Michael K. Reiter and Pierangela Samarati, editors, *ACM CCS 2001*, pages 196–205. ACM Press, November 2001.

[Rog04]      Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 16–31. Springer, Heidelberg, December 2004.

[RS06]       Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 373–390. Springer, Heidelberg, May / June 2006.

[Sch98]      R. Schroeppel. Hasty pudding cipher. http://www.cs.arizona.edu/rcs/hpc, 1998.

[SWGW21]     Yaobin Shen, Lei Wang, Dawu Gu, and Jian Weng. Revisiting the security of DbHtS MACs: Beyond-birthday-bound in the multi-user setting. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part III*, volume 12827 of *LNCS*, pages 309–336, Virtual Event, August 2021. Springer, Heidelberg.

[Tes15]      Stefano Tessaro. Optimally secure block ciphers from ideal primitives. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 437–462. Springer, Heidelberg, November / December 2015.

# Supplementary Material

## A    Balls-into-bins Lemmas

We will repeatedly appeal to the following classical result for the proof of Theorem 2 and Theorem 3.

**Lemma 1.** *Assume we throw $M$ balls uniformly at random and independently into $N$ bins with $8 \leq M \leq N$. Then, with probability larger than $1 - 1/N$, the maximal number of balls in any bin is $2 \log_2(M)$.*

*Proof.* Let $X_i$, $1 \leq i \leq N$, denote the load of the $i$-th bin and let $Y_{i,j}$, $1 \leq i \leq N$ and $1 \leq j \leq M$, be the indicator variable which is 1 if ball $j$ lands in bin $i$ and 0 otherwise. Then $X_i = \sum_{j=1}^{M} Y_{i,j}$ and $\Pr[Y_{i,j} = 1] = 1/N$. Let $\mu$ be the expected value of $X_i$. By linearity of the expectation, $\mu = M/N$. Since $X_i$ is the sum of independent Bernoulli trials, by the multiplicative Chernoff bound, we have that for any $\delta > 0$,

$$\Pr[X_i \geq (1 + \delta)\mu] \leq \left( \frac{e^\delta}{(1 + \delta)^{1+\delta}} \right)^\mu.$$

Letting $\alpha = 1 + \delta$, we obtain

$$\Pr[X_i \geq \alpha\mu] \leq \left( \frac{e^{\alpha - 1}}{\alpha^\delta} \right)^\mu < \left( \frac{e}{\alpha} \right)^{\alpha\mu}.$$

Taking $\alpha = 2N \log_2(M)/M$ yields (note that $\alpha > 1$)

$$
\begin{aligned}
\Pr[X_i \geq 2\log_2(M)] &\leq \left( \frac{eM}{2N \log_2(M)} \right)^{2 \log_2(M)} \\
&= \frac{M^2}{N^2} \left( \frac{e}{2 \log_2(M)} \right)^{2 \log_2(M)} \left( \frac{M}{N} \right)^{2 \log_2(M) - 2} \\
&\leq \frac{M^2}{N^2} \left( \frac{1}{2} \right)^{2 \log_2(M)} \left( \frac{M}{N} \right)^{2 \log_2(M) - 2} &(M \geq 8) \\
&= \frac{1}{N^2} \left( \frac{M}{N} \right)^{2 \log_2(M) - 2} \\
&\leq \frac{1}{N^2} &(M \leq N).
\end{aligned}
$$

By the union bound over all bins, the probability that there is a bin with $2 \log_2(M)$ balls or more is at most $1/N$. $\square$

We will also need the following variant.

**Lemma 2.** *Consider $N$ bins numbered from $0$ to $N-1$. Fix some integer $q$, $1 \leq q \leq M$, and a sequence of integers $(\ell_1, \ldots, \ell_q)$ with $1 \leq \ell_j \leq M$ and $\sum_{j=1}^{q} \ell_j = M$. Consider the following random process: for $j = 1, \ldots, q$, an initial bin $x_j$ is drawn uniformly at random and a ball is thrown in each bin $x_j$, $x_j + 1$, $\ldots$, $x_j + \ell_j - 1$, where bin numbers are taken mod $N$. (Hence, at the end of the process, $M$ balls have been thrown in total.) Then, with probability larger than $1 - 1/N$, the maximal number of balls in any bin is $2\log_2(M)$.*

*Proof.* The proof is similar to the proof of Lemma 1. Let $X_i$, $1 \leq i \leq N$, denote the load of the $i$-th bin and let $Y_{i,j}$, $1 \leq i \leq N$ and $1 \leq j \leq q$, be the indicator variable which is 1 if some ball of the $j$-th throw lands in bin $i$ and 0 otherwise. Then $X_i = \sum_{j=1}^{q} Y_{i,j}$ (as for each throw, at most one ball can land in any bin) and $\Pr[Y_{i,j} = 1] = \ell_j/N$. Let $\mu$ be the expected value of $X_i$. By linearity of the expectation, $\mu = \sum_{j=1}^{q} \ell_j/N = M/N$. Since $X_i$ is the sum of independent Poisson trials, the multiplicative Chernoff bound applies as in the proof of Lemma 1 with exactly the same parameters. $\square$

## B    Proof of Theorem 2

The proof of Theorem 2 relies on the H-coefficients method. Fix a deterministic adversary $\mathcal{A}$ and assume without loss of generality that $\mathcal{A}$ makes *exactly* $q_{\mathrm{ic}}$ queries in total to IC or $\mathrm{IC}^{-1}$ and exactly $q_{\mathrm{enc}}$ queries to ENC of total length (in number of $n$-bit blocks) exactly $\sigma$ and that it never makes a pointless query, where a pointless query is either:

- a repeated query to IC, $\mathrm{IC}^{-1}$, or ENC,
- a query $\mathrm{IC}(K, T, X)$ if there was a previous query $\mathrm{IC}^{-1}(K, T, Y)$ that returned $X$ or a query $\mathrm{IC}^{-1}(K, T, Y)$ if there was a previous query $\mathrm{IC}(K, T, X)$ that returned $Y$,
- a query $\mathrm{ENC}(i, N, M)$ such that $M = \epsilon$ or $i > u$, where $u$ is the current value of the counter keeping track of NEW queries.

The transcript consists of queries to the ideal cipher that we record as a list $\tau_{\mathrm{ic}}$ containing tuples $(K, T, X, Y)$ such that $\mathcal{A}$ made either a query $\mathrm{IC}(K, T, X)$ that returned $Y$ or a query $\mathrm{IC}^{-1}(K, T, Y)$ that returned $X$ and queries to ENC that we record as a list $\tau_{\mathrm{enc}}$ containing tuples $(i, N, M, (R, C))$ such that $\mathcal{A}$ made a query $\mathrm{ENC}(i, N, M)$ that returned $(R, C)$. If an ENC query is such that the message length is not a multiple of $n$, we assume that the last block is padded with enough zeros so that it has length $n$ before returning the oracle answer (this is *wlog* as this can only increase the adversary's advantage). With this convention, the length of messages and ciphertexts in encryption query/answer pairs is always a multiple of $n$, and we write such a pair $(i, N, M_0 \| \cdots \| M_{\ell-1}, (R, C_0 \| \cdots \| C_{\ell-1}))$ where $\ell = |M|_n$ and $|M_j| = |C_j| = n$ for every $j \in [\![0, \ell-1]\!]$. Queries to NEW are not explicitly recorded, but after the adversary has finished interacting with the oracles, we reveal the keys $\mathbf{K} = (K_1, \ldots, K_u)$ generated by NEW calls, where $u$ is the final value of the counter keeping track of NEW queries (note that unlike $q_{\mathrm{enc}}$,

$u$ is not fixed and depends on the transcript) and add $\mathbf{K}$ to the transcript. (Again, this is *wlog* as this can only increase the adversary's advantage; note that these keys are also generated in the ideal world, but never used afterwards.) We let $\Lambda_{\mathrm{re}}$, resp. $\Lambda_{\mathrm{id}}$ denote the probability distribution of the transcript when $\mathcal{A}$ interacts with game $\mathrm{Real}_{\mathsf{GCTR}[E,f_T,f_X]}^{\mathrm{mu\text{-}nive}}(\mathcal{A})$, resp. $\mathrm{Ideal}_{\mathsf{GCTR}[E,f_T,f_X]}^{\mathrm{mu\text{-}nive}}(\mathcal{A})$ (see Figure 2).

From the transcript of encryption queries $\tau_{\mathrm{enc}}$, we define the *split* transcript $\overline{\tau_{\mathrm{enc}}}$ as follows. For a query $Q = (i, N, M_0 \| \cdots \| M_{\ell-1}, (R, C_0 \| \cdots \| C_{\ell-1})) \in \tau_{\mathrm{enc}}$, we define the list

$$\mathcal{S}(Q) := \big((i, N, R, j, M_j, C_j)\big)_{j \in [\![0, \ell-1]\!]}.$$

Then $\overline{\tau_{\mathrm{enc}}} := \|_{Q \in \tau_{\mathrm{enc}}} \mathcal{S}(Q)$, the concatenation of all lists $\mathcal{S}(Q)$ for $Q \in \tau_{\mathrm{enc}}$. In all the following, we call a tuple $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ a *split encryption query*, or simply *split query* for short. Note that $\overline{\tau_{\mathrm{enc}}}$ contains exactly $\sigma$ split queries and that each split query $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ is associated with an evaluation of IC for which it allows to recover the corresponding key $K_i$, tweak $f_T(N, R, j)$, plaintext $f_X(N, R, j)$, and output $M \oplus C$.

We say that a transcript $\tau = (\tau_{\mathrm{ic}}, \tau_{\mathrm{enc}}, \mathbf{K})$ is bad if one of the following conditions is satisfied:

(C-1) There exist two distinct[15] split encryption queries $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ and $(i', N', R', j', M', C') \in \overline{\tau_{\mathrm{enc}}}$ such that

$$\begin{cases} K_i = K_{i'} \\ f_T(N, R, j) = f_T(N', R', j') \\ f_X(N, R, j) = f_X(N', R', j'). \end{cases}$$

(C-2) There exist two distinct split encryption queries $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ and $(i', N', R', j', M', C') \in \overline{\tau_{\mathrm{enc}}}$ such that

$$\begin{cases} K_i = K_{i'} \\ f_T(N, R, j) = f_T(N', R', j') \\ M \oplus C = M' \oplus C'. \end{cases}$$

(C-3) There exist $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ and $(K, T, X, Y) \in \tau_{\mathrm{ic}}$ such that

$$\begin{cases} K_i = K \\ f_T(N, R, j) = T \\ f_X(N, R, j) = X. \end{cases}$$

(C-4) There exist $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ and $(K, T, X, Y) \in \tau_{\mathrm{ic}}$ such that

$$\begin{cases} K_i = K \\ f_T(N, R, j) = T \\ M \oplus C = Y. \end{cases}$$

Otherwise, we say that $\tau$ is good and let $\Theta_{\mathrm{bad}}$, resp. $\Theta_{\mathrm{good}}$ denote the set of bad, resp. good transcripts.

---

[15] By distinct, we mean at different positions in the list $\overline{\tau_{\mathrm{enc}}}$, but the two tuples might be equal.

PROBABILITY OF BAD TRANSCRIPTS. First, we upper bound the probability of bad transcripts in the ideal world. This depends on the specific choice of functions $f_T$ and $f_X$. We consider each condition in turn, letting $\Theta_i$ denote the set of transcripts satisfying condition (C-$i$), $i \in [\![1, 4]\!]$.

CONDITION (C-1). We must distinguish CTRT and the four other modes. Before that, we define two subsets $\Theta_{1,1}$ and $\Theta_{1,2}$ of $\Theta_1$. Subset $\Theta_{1,1}$ consists of transcripts such that there exist distinct split queries $(i, N, R, j, M, C)$ and $(i, N', R', j', M', C')$ in $\overline{\tau_{\text{enc}}}$ such that $f_T(N, R, j) = f_T(N', R', j')$ and $f_X(N, R, j) = f_X(N', R', j')$. (Note the equality of the first component of the two split queries.) Subset $\Theta_{1,2}$ consists of transcripts such that there exist distinct split queries $(i, N, R, j, M, C) \in \overline{\tau_{\text{enc}}}$ and $(i', N', R', j', M', C') \in \overline{\tau_{\text{enc}}}$ such that $i \neq i'$, $K_i = K_{i'}$, $f_T(N, R, j) = f_T(N', R', j')$, and $f_X(N, R, j) = f_X(N', R', j')$.

<u>CTRT:</u> The condition is equivalent to $(K_i, R + \langle j \rangle_c, N) = (K_{i'}, R' + \langle j' \rangle_c, N')$. The two split queries cannot originate from the same encryption query as otherwise $R = R'$ and hence $\langle j \rangle_c = \langle j' \rangle_c$, which would imply that these two split queries are the same. First, we introduce some additional notation. For any (user, nonce) pair $(i, N)$ appearing in the encryption queries, let $q_{\text{enc}}(i, N)$ denote the number of encryption queries for this pair $(i, N)$, let $\ell_\iota(i, N), 1 \leq \iota \leq q_{\text{enc}}(i, N)$, denote the length of the $\iota$-th encryption query (for some arbitrary ordering) for pair $(i, N)$, and let $\sigma(i, N) = \sum_{\iota=1}^{q_{\text{enc}}(i,N)} \ell_\iota(i, N)$. Note that $q_{\text{enc}}(i, N) \leq \mu$ for any pair $(i, N)$ and $\sum_{(i,N)} \sigma(i, N) = \sigma$. We start with $\Theta_{1,1}$ (i.e., the two split queries are for the same user, so that the equality of the keys is automatically satisfied). Fix some pair $(i, N)$ and consider two distinct encryption queries for this (user, nonce) pair $(i, N, M, (R, C))$ and $(i, N, M', (R', C'))$ of length respectively $\ell$ and $\ell'$. Then, the probability over the random draw of $R$ and $R'$ that $R + \langle j \rangle_c = R' + \langle j' \rangle_c$ for some $j \in [\![0, \ell - 1]\!]$ and $j' \in [\![0, \ell' - 1]\!]$ is $(\ell + \ell' - 1)/2^r$. Summing over all pairs of queries and all pairs $(i, N)$, we have

$$
\begin{aligned}
\Pr[\Lambda_{\text{id}} \in \Theta_{1,1}] &\leq \sum_{(i,N)} \sum_{\iota=1}^{q_{\text{enc}}(i,N)-1} \sum_{\iota'=\iota+1}^{q_{\text{enc}}(i,N)} \frac{\ell_\iota(i, N) + \ell_{\iota'}(i, N) - 1}{2^r} \\
&\leq \sum_{(i,N)} \sum_{\iota=1}^{q_{\text{enc}}(i,N)-1} \frac{(q_{\text{enc}}(i, N) - 1)\ell_\iota(i, N) + \sigma(i, N)}{2^r} \\
&\leq \sum_{(i,N)} \frac{2(q_{\text{enc}}(i, N) - 1)\sigma(i, N)}{2^r} \\
&\leq \sum_{(i,N)} \frac{2(\mu - 1)\sigma(i, N)}{2^r} \\
&= \frac{2(\mu - 1)\sigma}{2^r}.
\end{aligned}
$$

Consider now $\Theta_{1,2}$. Let $\ell_\iota$, $1 \leq \iota \leq q_{\text{enc}}$, denote the length of the $\iota$-th encryption query (for some arbitrary ordering). Consider two distinct encryption

36

queries $(i, N, M, (R, C))$ and $(i', N', M', (R', C'))$ with $i \neq i'$ of length respectively $\ell$ and $\ell'$. Then $K_i = K_{i'}$ with probability $2^{-k}$ and, as before, the probability over the random draw of $R$ and $R'$ that $R + \langle j \rangle_c = R' + \langle j' \rangle_c$ for some $j \in [\![0, \ell - 1]\!]$ and $j' \in [\![0, \ell' - 1]\!]$ is $(\ell + \ell' - 1)/2^r$. Summing over all pairs of queries, we have

$$
\begin{aligned}
\Pr[\Lambda_{\mathrm{id}} \in \Theta_{1,2}] &\leq \sum_{\iota=1}^{q_{\mathrm{enc}}-1} \sum_{\iota'=\iota+1}^{q_{\mathrm{enc}}} \frac{\ell_\iota + \ell_{\iota'} - 1}{2^{k+r}} \\
&\leq \sum_{\iota=1}^{q_{\mathrm{enc}}-1} \frac{(q_{\mathrm{enc}} - 1)\ell_\iota + \sigma}{2^{k+r}} \\
&\leq \frac{2(q_{\mathrm{enc}} - 1)\sigma}{2^{k+r}}.
\end{aligned}
$$

All in all,

$$
\Pr[\Lambda_{\mathrm{id}} \in \Theta_1] \leq \frac{2(\mu - 1)\sigma}{2^r} + \frac{2q_{\mathrm{enc}}\sigma}{2^{k+r}} \leq \frac{2\mu\sigma}{2^r},
$$

where we used $q_{\mathrm{enc}}/2^k \leq 1$ for the last inequality.

GCTR-N, R, C, Z: One can easily check that for all four variants, condition (C-1) is equivalent to

$$
(K_i, N, R, \langle j \rangle_c) = (K_{i'}, N', R', \langle j' \rangle_c).
$$

Two split queries $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ and $(i', N', R', j', M', C') \in \overline{\tau_{\mathrm{enc}}}$ satisfying (C-1) cannot originate from the same encryption query as otherwise $\langle j \rangle_c = \langle j' \rangle_c$ would imply that these two split queries are the same. Hence, condition (C-1) is satisfied *iff* there exist two distinct encryption queries $(i, N, M, (R, C)) \in \tau_{\mathrm{enc}}$ and $(i', N', M', (R', C')) \in \tau_{\mathrm{enc}}$ such that $(K_i, N, R) = (K_{i'}, N', R')$. We start with $\Theta_{1,1}$. Since $R$ and $R'$ are independently distributed for two distinct encryption queries, $R = R'$ is satisfied with probability $2^{-r}$. Moreover, $N = N'$ implies that we only need to apply the union bound on the at most $(\mu - 1)q_{\mathrm{enc}}/2$ unordered pairs of encryption queries sharing the same nonce. (In details: fix an encryption query; then there are at most $\mu - 1$ other encryption queries with the same nonce; summing over the $q_{\mathrm{enc}}$ queries and dividing by two as each pair is counted twice yields the result.) Hence, $\Pr[\Lambda_{\mathrm{id}} \in \Theta_{1,1}] \leq (\mu - 1)q_{\mathrm{enc}}/2^{r+1}$. Consider now $\Theta_{1,2}$. Then $K_i = K_{i'}$ with probability $2^{-k}$ and $R = R'$ with probability $2^{-r}$ (on the other hand, the adversary can set $N = N'$ at will). Summing over the $q_{\mathrm{enc}}(q_{\mathrm{enc}} - 1)/2$ unordered pairs of encryption queries, we obtain $\Pr[\Lambda_{\mathrm{id}} \in \Theta_{1,2}] \leq q_{\mathrm{enc}}^2/2^{k+r+1}$. All in all,

$$
\Pr[\Lambda_{\mathrm{id}} \in \Theta_1] \leq \frac{(\mu - 1)q_{\mathrm{enc}}}{2^{r+1}} + \frac{q_{\mathrm{enc}}^2}{2^{k+r+1}} \leq \frac{\mu q_{\mathrm{enc}}}{2^{r+1}},
$$

where we used $q_{\mathrm{enc}}/2^k \leq 1$ for the last inequality.

CONDITION (C-2). Here we must distinguish the five variants. Before that, we define two subsets $\Theta_{2,1}$ and $\Theta_{2,2}$ of $\Theta_2$. Subset $\Theta_{2,1}$ consists of transcripts such that there exist distinct split queries $(i, N, R, j, M, C)$ and $(i, N', R', j', M', C')$ in $\overline{\tau_{\text{enc}}}$ such that $f_T(N, R, j) = f_T(N', R', j')$ and $M \oplus C = M' \oplus C'$. Subset $\Theta_{2,2}$ consists of transcripts such that there exist distinct split queries $(i, N, R, j, M, C) \in \overline{\tau_{\text{enc}}}$ and $(i', N', R', j', M', C') \in \overline{\tau_{\text{enc}}}$ such that $i \neq i'$, $K_i = K_{i'}$, $f_T(N, R, j) = f_T(N', R', j')$, and $M \oplus C = M' \oplus C'$. In the following, we fix two split queries $(i, N, R, j, M, C) \in \overline{\tau_{\text{enc}}}$ and $(i', N', R', j', M', C') \in \overline{\tau_{\text{enc}}}$ (with either $i = i'$ for $\Theta_{2,1}$ or $i \neq i'$ for $\Theta_{2,2}$), upper bound the probability that conditions are satisfied, and then apply the union bound.

<u>CTRT:</u> Then $f_T(N, R, j) = f_T(N', R', j')$ translates to $R + \langle j \rangle_c = R' + \langle j' \rangle_c$. We upper bound directly the probability that $\Lambda_{\text{id}} \in \Theta_2$ without distinguishing whether $\Lambda_{\text{id}} \in \Theta_{2,1}$ or $\Lambda_{\text{id}} \in \Theta_{2,2}$. The two split queries cannot originate from the same encryption query as otherwise $R = R'$ and hence $\langle j \rangle_c = \langle j' \rangle_c$, which would imply that these two split queries are the same. Hence $R$ and $R'$ are uniformly random and independent and $R + \langle j \rangle_c = R' + \langle j' \rangle_c$ holds with probability $2^{-r}$. Moreover, $C$ and $C'$ are uniformly random and independent, hence $M \oplus C = M' \oplus C'$ holds with probability $2^{-n}$. Summing over the $\sigma(\sigma - 1)/2$ unordered pairs of split encryption queries, one has

$$\Pr[\Lambda_{\text{id}} \in \Theta_2] \leq \frac{\sigma^2}{2^{r+n+1}}.$$

<u>GCTR-N:</u> Then $f_T(N, R, j) = f_T(N', R', j')$ translates to $(R, \langle j \rangle_c) = (R', \langle j' \rangle_c)$. We upper bound directly the probability that $\Lambda_{\text{id}} \in \Theta_2$ without distinguishing whether $\Lambda_{\text{id}} \in \Theta_{2,1}$ or $\Lambda_{\text{id}} \in \Theta_{2,2}$. The two split queries cannot originate from the same encryption query as otherwise $\langle j \rangle_c = \langle j' \rangle_c$ would imply that these two split queries are the same. Hence $R$ and $R'$ are uniformly random and independent and $R = R'$ holds with probability $2^{-r}$. Moreover, $C$ and $C'$ are uniformly random and independent, hence $M \oplus C = M' \oplus C'$ holds with probability $2^{-n}$. There are $\sigma$ possible choices for the first split query $(i, N, R, j, M, C)$; then there are at most $q_{\text{enc}}$ other split queries $(i', N', R', j', M', C')$ with $\langle j' \rangle_c = \langle j \rangle_c$; dividing by two as each pair is counted twice, one has
$$\Pr[\Lambda_{\text{id}} \in \Theta_2] \leq \frac{\sigma q_{\text{enc}}}{2^{r+n+1}}.$$

<u>GCTR-R:</u> Then $f_T(N, R, j) = f_T(N', R', j')$ translates to $(N, \langle j \rangle_c) = (N', \langle j' \rangle_c)$. The two split queries cannot originate from the same encryption query as otherwise $\langle j \rangle_c = \langle j' \rangle_c$ would imply that these two split queries are the same. We start with $\Theta_{2,1}$. One has $M \oplus C = M' \oplus C'$ with probability $2^{-n}$. There are $\sigma$ possible choices for the first split query $(i, N, R, j, M, C)$; then there are at most $\mu - 1$ other split queries $(i, N', R', j', M', C')$ with $(N', \langle j' \rangle_c) = (N, \langle j \rangle_c)$; dividing by two as each pair is counted twice, one has $\Pr[\Lambda_{\text{id}} \in \Theta_{2,1}] \leq (\mu - 1)\sigma/2^n$. Consider now $\Theta_{2,2}$. Then $K_i = K_{i'}$ with probability $2^{-k}$ and $M \oplus C = M' \oplus C'$ with probability $2^{-n}$. There are $\sigma$ possible choices for the first split query $(i, N, R, j, M, C)$; then there are at

most $q_{\mathrm{enc}}$ other split queries $(i', N', R', j', M', C')$ with $\langle j'\rangle_c = \langle j\rangle_c$; dividing by two as each pair is counted twice, one has $\Pr[\Lambda_{\mathrm{id}} \in \Theta_{2,2}] \leq \sigma q_{\mathrm{enc}}/2^{k+n+1}$. All in all,

$$\Pr[\Lambda_{\mathrm{id}} \in \Theta_2] \leq \frac{(\mu-1)\sigma}{2^n} + \frac{\sigma q_{\mathrm{enc}}}{2^{k+n+1}}.$$

<u>GCTR-C</u>: Then $f_T(N, R, j) = f_T(N', R', j')$ translates to $(N, R) = (N', R')$. Note that here the two split queries may originate from the same encryption query. We start with $\Theta_{2,1}$. One has $M \oplus C = M' \oplus C'$ with probability $2^{-n}$. Let $F$ denote the event that there exist two distinct encryption queries for the same pair $(i, N)$ such that the resulting $R$ values collide. Then $\Pr[F] \leq (\mu-1)q_{\mathrm{enc}}/2^{r+1}$. Conditioned on $\neg F$, we can count the number of pairs of split queries on which we must sum as follows: there are $\sigma$ possible choices for the first split query $(i, N, R, j, M, C)$; then there are at most $\ell_{\max} - 1$ other split queries $(i, N', R', j', M', C')$ with $(N', R') = (N, R)$; dividing by two as each pair is counted twice, one has $\Pr[\Lambda_{\mathrm{id}} \in \Theta_{2,1}] \leq (\mu-1)q_{\mathrm{enc}}/2^{r+1} + \sigma\ell_{\max}/2^{n+1}$. Consider now $\Theta_{2,2}$. Then $K_i = K_{i'}$ with probability $2^{-k}$ and $M \oplus C = M' \oplus C'$ with probability $2^{-n}$. Summing over the $\sigma(\sigma - 1)$ unordered pairs of distinct split queries, one has $\Pr[\Lambda_{\mathrm{id}} \in \Theta_{2,2}] \leq \sigma^2/2^{k+n+1}$. All in all,

$$\Pr[\Lambda_{\mathrm{id}} \in \Theta_2] \leq \frac{(\mu-1)q_{\mathrm{enc}}}{2^{r+1}} + \frac{\sigma\ell_{\max}}{2^{n+1}} + \frac{\sigma^2}{2^{k+n+1}}.$$

<u>GCTR-Z</u>: Here we can observe that if $\tau \notin \Theta_1$, then necessarily $\tau \notin \Theta_2$. Indeed, since condition $f_X(N, R, j) = f_X(N', R', j')$ always vacuously holds, $\tau \notin \Theta_1$ means that for every distinct split encryption queries $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ and $(i', N', R', j', M', C') \in \overline{\tau_{\mathrm{enc}}}$, either $K_i \neq K_{i'}$ or $(N, R, \langle j\rangle_c) \neq (N', R', \langle j'\rangle_c)$, which implies that $\tau \notin \Theta_2$.

$$\Pr[\Lambda_{\mathrm{id}} \in \Theta_2 \mid \Lambda_{\mathrm{id}} \notin \Theta_1] = 0.$$

CONDITION (C-3). We must distinguish CTRT and the four other modes.

<u>CTRT</u>: We can view each encryption query $(i, N, M, (R, C)) \in \tau_{\mathrm{enc}}$ as throwing $\ell$ balls $R + \langle j\rangle_c$ into $2^r$ bins (where $\ell$ is the length of $M$ in $n$-bit blocks), the first bin being chosen uniformly at random and the balls are thrown in consecutive bins. In total, we throw $\sigma \leq 2^r$ balls. Hence, by Lemma 2, each bin contains at most $2\log_2(\sigma) \leq 2r$ balls, except with probability at most $2^{-r}$. Assume this is the case and fix an ideal cipher query $(K, T, X, Y) \in \tau_{\mathrm{ic}}$. Then there are at most $2r$ split queries $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ such that $R + \langle j\rangle_c = T$ and the probability that $K_i = K$ for one of those split queries is at most $2r/2^k$. Summing over all $q_{\mathrm{ic}}$ ideal cipher queries, one has

$$\Pr[\Lambda_{\mathrm{id}} \in \Theta_3] \leq \frac{1}{2^r} + \frac{2r q_{\mathrm{ic}}}{2^k}.$$

<u>GCTR-N, R, C, Z</u>: We can view each encryption query $(i, N, M, (R, C)) \in \tau_{\mathrm{enc}}$ as throwing a ball $R$ uniformly at random into $2^r$ bins. In total, we throw

$q_{\mathrm{enc}} \leq 2^r$ balls and by [Lemma 1](#) each bin contains at most $2\log_2(q_{\mathrm{enc}}) \leq 2r$ balls, except with probability at most $2^{-r}$. Assume this is the case and fix a query $(K, T, X, Y) \in \tau_{\mathrm{ic}}$. Then, for all four modes, there are at most $2r$ split queries $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ such that $f_T(N, R, j) = T$ and $f_X(N, R, j) = X$ (since these two equations uniquely fix $R$ and $\langle j \rangle_c$) and the probability that $K_i = K$ for one of those split queries is at most $2r/2^k$. Summing over all $q_{\mathrm{ic}}$ ideal cipher queries, one has

$$\Pr[\Lambda_{\mathrm{id}} \in \Theta_3] \leq \frac{1}{2^r} + \frac{2r q_{\mathrm{ic}}}{2^k}.$$

CONDITION (C-4). The reasoning is the same for the five modes. We can view each split query $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ as throwing a ball $M \oplus C$ uniformly at random into $2^n$ bins. In total, we throw $\sigma$ balls and by [Lemma 1](#) each bin contains at most $2\log_2(\sigma) \leq 2n$ balls, except with probability at most $2^{-n}$. Assume this is the case and fix a query $(K, T, X, Y) \in \tau_{\mathrm{ic}}$. Then, there are at most $2n$ split queries $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ such that $M \oplus C = Y$ and the probability that $K_i = K$ for one of those split queries is at most $2n/2^k$. Summing over all $q_{\mathrm{ic}}$ ideal cipher queries, one has

$$\Pr[\Lambda_{\mathrm{id}} \in \Theta_4] \leq \frac{1}{2^n} + \frac{2n q_{\mathrm{ic}}}{2^k}.$$

We can obtain an upper bound on $\Pr[\Lambda_{\mathrm{id}} \in \Theta_{\mathrm{bad}}]$ by adding probabilities $\Pr[\Lambda_{\mathrm{id}} \in \Theta_i]$ above for $i \in [\![1, 4]\!]$.

GOOD TRANSCRIPTS PROBABILITY RATIO. We now prove that for every good transcript $\tau$, $\Pr[\Lambda_{\mathrm{re}} = \tau] \geq \Pr[\Lambda_{\mathrm{id}} = \tau]$ (meaning we can use [Theorem 1](#) with $\beta = 0$). Fix a good transcript $\tau = (\tau_{\mathrm{ic}}, \tau_{\mathrm{enc}}, \mathbf{K})$ and let $u$ be the length of $\mathbf{K}$. Let also $\ell_i$, $i \in [\![1, q_{\mathrm{enc}}]\!]$, denote the length (in $n$-bit blocks) of the message (and hence the ciphertext) in the $i$-th encryption query. Note that $\sum_{i \in [\![1, q_{\mathrm{enc}}]\!]} \ell_i = \sigma$. For any $(K, T) \in \{0, 1\}^k \times \{0, 1\}^t$, let $\mathcal{X}_{\mathrm{ic}}(K, T)$ denote the set of inputs $X \in \{0, 1\}^n$ such that there exists $Y$ with $(K, T, X, Y) \in \tau_{\mathrm{ic}}$ and $\mathcal{Y}_{\mathrm{ic}}(K, T)$ denote the set of outputs $Y \in \{0, 1\}^n$ such that there exists $X$ with $(K, T, X, Y) \in \tau_{\mathrm{ic}}$. Note that since $\tau$ is good, for any $(K, T, X) \in \{0, 1\}^k \times \{0, 1\}^t \times \{0, 1\}^n$, there is at most one split query $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ such that $K = K_i$, $T = f_T(N, R, j)$, and $X = f_X(N, R, j)$ as otherwise condition (C-1) would be satisfied. With this in mind, let $\mathcal{X}_{\mathrm{enc}}(K, T)$ denote the set of $X \in \{0, 1\}^n$ such that IC was evaluated on $(K, T, X)$ in some encryption query, i.e.,

$$\mathcal{X}_{\mathrm{enc}}(K, T) := \{X \in \{0, 1\}^n \colon \exists (i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}} \colon$$
$$K = K_i \wedge T = f_T(N, R, j) \wedge X = f_X(N, R, j)\}.$$

Note that by the previous observation, one has $\sum_{(K, T)} |\mathcal{X}_{\mathrm{enc}}(K, T)| = \sigma$. Similarly, for any $(K, T, Y) \in \{0, 1\}^k \times \{0, 1\}^t \times \{0, 1\}^n$, there is at most one split query $(i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}}$ such that $K = K_i$, $T = f_T(N, R, j)$, and $Y = M \oplus C$ (as

otherwise condition (C-2) would be satisfied). Let $\mathcal{Y}_{\mathrm{enc}}$ be the set defined as

$$\mathcal{Y}_{\mathrm{enc}}(K,T) \coloneqq \{Y \in \{0,1\}^n \colon \exists (i, N, R, j, M, C) \in \overline{\tau_{\mathrm{enc}}} \colon$$
$$K = K_i \wedge T = f_T(N, R, j) \wedge Y = M \oplus C\}.$$

Because $\tau$ is good, for any $(K,T)$, $\mathcal{X}_{\mathrm{ic}}(K,T) \cap \mathcal{X}_{\mathrm{enc}}(K,T) = \emptyset$ (as otherwise condition (C-3) would be satisfied) and $\mathcal{Y}_{\mathrm{ic}}(K,T) \cap \mathcal{Y}_{\mathrm{enc}}(K,T) = \emptyset$ (as otherwise condition (C-4) would be satisfied). Hence, for each pair $(K,T)$, $\tau_{\mathrm{ic}}$ and $\tau_{\mathrm{enc}}$ together impose a set of $|\mathcal{X}_{\mathrm{ic}}(K,T)| + |\mathcal{X}_{\mathrm{enc}}(K,T)|$ equations on the random permutation $E_{\mathrm{ic}}(K, T, \cdot)$ internally sampled by the ideal cipher of the form $E_{\mathrm{ic}}(K, T, X) = Y$ where all $X$'s are distinct and all $Y$'s are distinct. From this it follows that

$$\Pr[\Lambda_{\mathrm{re}} = \tau] = \frac{1}{2^{uk}} \cdot \frac{1}{2^{rq_{\mathrm{enc}}}} \cdot \left( \prod_{\substack{K \in \{0,1\}^k \\ T \in \{0,1\}^t}} \prod_{i=0}^{|\mathcal{X}_{\mathrm{ic}}(K,T)| + |\mathcal{X}_{\mathrm{enc}}(K,T)| - 1} \frac{1}{2^n - i} \right),$$

where the first term accounts for the random choice of keys, the second term for the random choice of $R$ values, and the third term for the probability that $E_{\mathrm{ic}}$ satisfies the constraints imposed by the transcripts $\tau_{\mathrm{ic}}$ and $\tau_{\mathrm{enc}}$. On the other hand, since in the ideal world the $\sigma$ ciphertext blocks are uniformly random and independent, one has

$$\Pr[\Lambda_{\mathrm{id}} = \tau] = \frac{1}{2^{uk}} \cdot \frac{1}{2^{rq_{\mathrm{enc}}}} \cdot \frac{1}{2^{\sigma n}} \cdot \left( \prod_{\substack{K \in \{0,1\}^k \\ T \in \{0,1\}^t}} \prod_{i=0}^{|\mathcal{X}_{\mathrm{ic}}(K,T)| - 1} \frac{1}{2^n - i} \right).$$

Hence,

$$\frac{\Pr[\Lambda_{\mathrm{re}} = \tau]}{\Pr[\Lambda_{\mathrm{id}} = \tau]} = 2^{\sigma n} \cdot \left( \prod_{\substack{K \in \{0,1\}^k \\ T \in \{0,1\}^t}} \prod_{i=0}^{|\mathcal{X}_{\mathrm{enc}}(K,T)| - 1} \frac{1}{2^n - |\mathcal{X}_{\mathrm{ic}}(K,T)| - i} \right)$$

$$= \prod_{\substack{K \in \{0,1\}^k \\ T \in \{0,1\}^t}} \prod_{i=0}^{|\mathcal{X}_{\mathrm{enc}}(K,T)| - 1} \frac{2^n}{2^n - |\mathcal{X}_{\mathrm{ic}}(K,T)| - i} \geq 1,$$

where we used that $\sum_{(K,T)} |\mathcal{X}_{\mathrm{enc}}(K,T)| = \sigma$.

CONCLUDING. The theorem follows from Theorem 1 with $\beta = 0$ by collecting all cases from the bad transcripts analysis and simplifying the bounds using $q_{\mathrm{enc}}/2^r \leq 1$, $q_{\mathrm{enc}}/2^k \leq 1$, and $\sigma/2^n \leq 1$ as follows:

CTRT:

$$\mathsf{Adv}^{\text{mu-nive}}_{\mathsf{CTRT}}(\mathcal{A}) \leq \frac{2\mu\sigma}{2^r} + \underbrace{\frac{\sigma^2}{2^{r+n+1}}}_{\leq \sigma/2^{r+1}} + \frac{1}{2^r} + \frac{2rq_{\text{ic}}}{2^k} + \frac{1}{2^n} + \frac{2nq_{\text{ic}}}{2^k}$$

$$\leq \frac{1}{2^r} + \frac{1}{2^n} + \frac{2(r+n)q_{\text{ic}}}{2^k} + \frac{(4\mu+1)\sigma}{2^{r+1}}$$

GCTR-N:

$$\mathsf{Adv}^{\text{mu-nive}}_{\mathsf{GCTR-N}}(\mathcal{A}) \leq \frac{\mu q_{\text{enc}}}{2^{r+1}} + \underbrace{\frac{\sigma q_{\text{enc}}}{2^{r+n+1}}}_{\leq \sigma/2^{n+1}} + \frac{1}{2^r} + \frac{2rq_{\text{ic}}}{2^k} + \frac{1}{2^n} + \frac{2nq_{\text{ic}}}{2^k}$$

$$\leq \frac{1}{2^r} + \frac{1}{2^n} + \frac{2(r+n)q_{\text{ic}}}{2^k} + \frac{\mu q_{\text{enc}}}{2^{r+1}} + \frac{\sigma}{2^{n+1}}$$

GCTR-R:

$$\mathsf{Adv}^{\text{mu-nive}}_{\mathsf{GCTR-R}}(\mathcal{A}) \leq \frac{\mu q_{\text{enc}}}{2^{r+1}} + \frac{(\mu-1)\sigma}{2^n} + \underbrace{\frac{\sigma q_{\text{enc}}}{2^{k+n+1}}}_{\leq \sigma/2^{n+1}} + \frac{1}{2^r} + \frac{2rq_{\text{ic}}}{2^k} + \frac{1}{2^n} + \frac{2nq_{\text{ic}}}{2^k}$$

$$\leq \frac{1}{2^r} + \frac{1}{2^n} + \frac{2(r+n)q_{\text{ic}}}{2^k} + \frac{\mu q_{\text{enc}}}{2^{r+1}} + \frac{\mu\sigma}{2^n}$$

GCTR-C:

$$\mathsf{Adv}^{\text{mu-nive}}_{\mathsf{GCTR-C}}(\mathcal{A}) \leq \frac{\mu q_{\text{enc}}}{2^{r+1}} + \frac{(\mu-1)q_{\text{enc}}}{2^{r+1}} + \frac{\sigma\ell_{\max}}{2^{n+1}} + \frac{\sigma^2}{2^{k+n+1}}$$

$$+ \frac{1}{2^r} + \frac{2rq_{\text{ic}}}{2^k} + \frac{1}{2^n} + \frac{2nq_{\text{ic}}}{2^k}$$

$$\leq \frac{1}{2^r} + \frac{1}{2^n} + \frac{2(r+n)q_{\text{ic}}}{2^k} + \frac{(2\mu-1)q_{\text{enc}}}{2^{r+1}} + \frac{\sigma\ell_{\max}}{2^{n+1}} + \frac{\sigma^2}{2^{k+n+1}}$$

GCTR-Z:

$$\mathsf{Adv}^{\text{mu-nive}}_{\mathsf{GCTR-Z}}(\mathcal{A}) \leq \frac{\mu q_{\text{enc}}}{2^{r+1}} + \frac{1}{2^r} + \frac{2rq_{\text{ic}}}{2^k} + \frac{1}{2^n} + \frac{2nq_{\text{ic}}}{2^k}$$

$$\leq \frac{1}{2^r} + \frac{1}{2^n} + \frac{2(r+n)q_{\text{ic}}}{2^k} + \frac{\mu q_{\text{enc}}}{2^{r+1}}.$$

# C   Proof of Theorem 3

The proof uses the H-coefficients technique. Fix a deterministic adversary $\mathcal{A}$ and assume without loss of generality that $\mathcal{A}$ makes *exactly* $q_{\text{ic}}$ queries in total to IC or $\text{IC}^{-1}$, exactly $q_{\text{tag}}$ queries to TAG of total length at most $\sigma_{\text{tag}}$, exactly $q_{\text{ver}}$ queries to VER of total length at most $\sigma_{\text{ver}}$, and that it never makes a pointless query, where a pointless query is either:

- a repeated query to IC, $\mathrm{IC}^{-1}$, TAG, or VER,
- a query $\mathrm{IC}(K,T,X)$ if there was a previous query $\mathrm{IC}^{-1}(K,T,Y)$ that returned $X$ or a query $\mathrm{IC}^{-1}(K,T,Y)$ if there was a previous query $\mathrm{IC}(K,T,X)$ that returned $Y$,
- a query $\mathrm{TAG}(i,N,U)$ or $\mathrm{VER}(i,N,U,V)$ such that $i > u$, where $u$ is the current value of the counter keeping track of NEW queries,
- a query $\mathrm{VER}(i,N,U,V)$ if there was a previous query $\mathrm{TAG}(i,N,U)$ that returned $V$.

The transcript consists of three types of queries:

- queries to the ideal cipher that we record as a list $\tau_{\mathrm{ic}}$ containing tuples $(K,T,X,Y)$ such that $\mathcal{A}$ made either a query $\mathrm{IC}(K,T,X)$ that returned $Y$ or a query $\mathrm{IC}^{-1}(K,T,Y)$ that returned $X$;
- queries to TAG that we record as a list $\tau_{\mathrm{tag}}$ containing tuples $(i,N,U,V)$ such that $\mathcal{A}$ made a query $\mathrm{TAG}(i,N,U)$ that returned $V$;
- queries to VER that we record as a list $\tau_{\mathrm{ver}}$ containing all tuples $(i,N,U,V)$ that were queried to VER (we do not keep track of the answers since we are interested in attainable transcripts, i.e., transcripts that can be obtained in the ideal world in which all queries to VER return $\perp$).

We do not keep track explicitly of NEW queries, but when $\mathcal{A}$ has finished interacting with the oracles, we reveal all keys $\mathbf{K} = ((K_{\mathrm{in}}^1, K_{\mathrm{out}}^1), \ldots, (K_{\mathrm{in}}^u, K_{\mathrm{out}}^u))$ generated by calls to NEW, where $u$ is the final value of the counter keeping track of NEW queries (note that $u$ is transcript-dependent) and add them to the transcript (this is *wlog* as this can only increase the adversary's advantage; note that these keys are also generated in the ideal world but never used afterwards). We let $\Lambda_{\mathrm{re}}$, resp. $\Lambda_{\mathrm{id}}$ denote the probability distribution of the transcript when $\mathcal{A}$ interacts with game $\mathrm{Real}_{\mathsf{NaT}[H,E]}^{\mathrm{mu\text{-}nprmac}}(\mathcal{A})$, resp. $\mathrm{Ideal}_{\mathsf{NaT}[H,E]}^{\mathrm{mu\text{-}nprmac}}(\mathcal{A})$ (see Figure 3).

We say that a transcript $\tau = (\tau_{\mathrm{ic}}, \tau_{\mathrm{tag}}, \tau_{\mathrm{ver}}, \mathbf{K})$ is bad if one of the following conditions is satisfied:

(C-1) There exist two distinct queries $(i,N,U,V) \in \tau_{\mathrm{tag}}$ and $(i',N',U',V') \in \tau_{\mathrm{tag}}$ such that
$$\begin{cases} K_{\mathrm{out}}^i = K_{\mathrm{out}}^{i'} \\ N = N' \\ H_{K_{\mathrm{in}}^i}(U) = H_{K_{\mathrm{in}}^{i'}}(U'). \end{cases}$$

(C-2) There exist two distinct queries $(i,N,U,V) \in \tau_{\mathrm{tag}}$ and $(i',N',U',V') \in \tau_{\mathrm{tag}}$ such that
$$\begin{cases} K_{\mathrm{out}}^i = K_{\mathrm{out}}^{i'} \\ N = N' \\ V = V'. \end{cases}$$

(C-3) There exist queries $(i,N,U,V) \in \tau_{\mathrm{tag}}$ and $(K,T,X,Y) \in \tau_{\mathrm{ic}}$ such that
$$\begin{cases} K_{\mathrm{out}}^i = K \\ 0^{t-\nu}\|N = T \\ H_{K_{\mathrm{in}}^i}(U) = X. \end{cases}$$

43

(C-4) There exist queries $(i, N, U, V) \in \tau_{\text{tag}}$ and $(K, T, X, Y) \in \tau_{\text{ic}}$ such that

$$\begin{cases} K_{\text{out}}^i = K \\ 0^{t-\nu} \| N = T \\ V = Y. \end{cases}$$

(C-5) There exist queries $(i, N, U, V) \in \tau_{\text{tag}}$ and $(i', N', U', V') \in \tau_{\text{ver}}$ such that

$$\begin{cases} K_{\text{out}}^i = K_{\text{out}}^{i'} \\ N = N' \\ H_{K_{\text{in}}^i}(U) = H_{K_{\text{in}}^{i'}}(U') \\ V = V'. \end{cases}$$

(C-6) There exist queries $(i, N, U, V) \in \tau_{\text{ver}}$ and $(K, T, X, Y) \in \tau_{\text{ic}}$ such that

$$\begin{cases} K_{\text{out}}^i = K \\ 0^{t-\nu} \| N = T \\ H_{K_{\text{in}}^i}(U) = X \\ V = Y. \end{cases}$$

Otherwise, we say that $\tau$ is good and let $\Theta_{\text{bad}}$, resp. $\Theta_{\text{good}}$ denote the set of bad, resp. good transcripts.

PROBABILITY OF BAD TRANSCRIPTS. First, we upper bound the probability of bad transcripts in the ideal world. We consider each condition in turn, letting $\Theta_i$ denote the set of transcripts satisfying condition (C-$i$), $i \in [\![1, 6]\!]$. For conditions (C-1), (C-3), (C-5), and (C-6), we assume *wlog* that the queries transcript $(\tau_{\text{ic}}, \tau_{\text{tag}}, \tau_{\text{ver}})$ is fixed as we can upper bound the probability using only the randomness of keys **K**. We assume that it involves $u$ users, and for every $i \in [\![1, u]\!]$ and $N \in \{0, 1\}^{\nu}$ we let $q_{i,N}$ denote the number of TAG queries involving user $i$ and nonce $N$ (with $q_{i,N} = 0$ if there were no such queries). When $q_{i,N} \geq 1$, we also let $\ell_{i,N,j}$, $1 \leq j \leq q_{i,N}$, denote the length of the $j$-th TAG query for user $i$ with nonce $N$ and we assume that queries are reordered such that $\ell_{i,N,1} \leq \ell_{i,N,2} \leq \cdots \leq \ell_{i,N,q_{i,N}}$. Note that by our assumptions, we have

$$q_{i,N} \leq \mu \text{ for every } (i, N), \tag{4}$$

$$\sum_{i=1}^{u} \sum_{N \in \{0,1\}^{\nu}} q_{i,N} = q_{\text{tag}}, \tag{5}$$

$$\text{and } \sum_{i=1}^{u} \sum_{N \in \{0,1\}^{\nu}} \sum_{j=1}^{q_{i,N}} \ell_{i,N,j} \leq \sigma_{\text{tag}}. \tag{6}$$

Similarly, for every $i \in [\![1, u]\!]$ and $N \in \{0, 1\}^{\nu}$ we let $q'_{i,N}$ denote the number of VER queries involving user $i$ and nonce $N$ (with $q'_{i,N} = 0$ if there were no

44

such queries) and when $q'_{i,N} \geq 1$, we let $\ell'_{i,N,j}$, $1 \leq j \leq q'_{i,N}$, denote the length of the $j$-th VER query for user $i$ with nonce $N$ and we assume that queries are reordered such that $\ell'_{i,N,1} \leq \ell'_{i,N,2} \leq \cdots \leq \ell'_{i,N,q'_{i,N}}$. Then

$$\sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} q'_{i,N} = q_{\text{ver}} \tag{7}$$

$$\text{and } \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} \sum_{j=1}^{q'_{i,N}} \ell'_{i,N,j} \leq \sigma_{\text{ver}}. \tag{8}$$

In the following, for $(U, U') \in \mathcal{U}^2$, we let

$$\delta_{U,U'} := \max\{\delta(\text{len}(U)), \delta(\text{len}(U'))\} = \alpha \max\{\text{len}(U), \text{len}(U')\}/2^k + \beta/2^n$$
$$\gamma_{U,U'} := \min\{\gamma(\text{len}(U)), \gamma(\text{len}(U'))\} = \alpha \min\{\text{len}(U), \text{len}(U')\}/2^k + \beta/2^n.$$

CONDITION (C-1). We define two subsets $\Theta_{1,1}$ and $\Theta_{1,2}$ of $\Theta_1$. Subset $\Theta_{1,1}$ consists of transcripts $\tau$ such that there exist distinct queries $(i, N, U, V)$ and $(i, N', U', V')$ in $\tau_{\text{tag}}$ such that $N = N'$ and $H_{K_{\text{in}}^i}(U) = H_{K_{\text{in}}^i}(U')$. Subset $\Theta_{1,2}$ consists of transcripts $\tau$ such that there exists $(i, N, U, V)$ and $(i', N', U', V')$ in $\tau_{\text{tag}}$ such that $i \neq i'$, $K_{\text{out}}^i = K_{\text{out}}^{i'}$, $N = N'$, and $H_{K_{\text{in}}^i}(U) = H_{K_{\text{in}}^{i'}}(U')$.

We start with $\Theta_{1,1}$. For any two distinct $(i, N, U, V)$ and $(i, N', U', V')$ in $\tau_{\text{tag}}$ such that $N = N'$, one has $U \neq U'$ by the assumption that $\mathcal{A}$ never repeats queries. Hence, by the assumption that $H$ is $\delta$-sAU, $H_{K_{\text{in}}^i}(U) = H_{K_{\text{in}}^i}(U')$ with probability at most $\delta_{U,U'}$. By the union bound over users $i \in [\![1, u]\!]$, nonces $N \in \{0,1\}^\nu$, and pairs of queries involving $(i, N)$ and using our assumption that queries are ordered according to their lengths, one has

$$\Pr[\Lambda_{\text{id}} \in \Theta_{1,1}] \leq \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} \sum_{j=1}^{q_{i,N}} \sum_{j'=1}^{j-1} \left( \frac{\alpha \ell_{i,N,j}}{2^k} + \frac{\beta}{2^n} \right)$$

$$= \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} \sum_{j=1}^{q_{i,N}} (j-1) \left( \frac{\alpha \ell_{i,N,j}}{2^k} + \frac{\beta}{2^n} \right)$$

$$\leq \frac{\alpha}{2^k} \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} \sum_{j=1}^{q_{i,N}} (q_{i,N} - 1)\ell_{i,N,j}$$

$$+ \frac{\beta}{2^n} \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} \frac{q_{i,N}(q_{i,N} - 1)}{2}$$

$$\leq \frac{\alpha}{2^k} \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} \sum_{j=1}^{q_{i,N}} (\mu - 1)\ell_{i,N,j} + \frac{\beta}{2^n} \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} \frac{q_{i,N}(\mu - 1)}{2}$$

$$\leq \frac{\alpha(\mu - 1)\sigma_{\text{tag}}}{2^k} + \frac{\beta(\mu - 1)q_{\text{tag}}}{2^{n+1}},$$

45

where we used (4) for the penultimate inequality and (5) and (6) for the last inequality.

Consider now $\Theta_{1,2}$. For any two distinct $(i, N, U, V)$ and $(i', N', U', V')$ in $\tau_{\text{tag}}$ such that $i \neq i'$, $K_{\text{out}}^i = K_{\text{out}}^{i'}$ with probability $2^{-k}$ and $H_{K_{\text{in}}^i}(U) = H_{K_{\text{in}}^{i'}}(U')$ with probability at most $\gamma_{U,U'}$ by $\gamma$-uniformity of $H$. (Indeed, assume *wlog* that $\text{len}(U) \leq \text{len}(U')$; then for any $K_{\text{in}}^{i'}$, $H_{K_{\text{in}}^i}(U) = H_{K_{\text{in}}^{i'}}(U')$ with probability at most $\gamma(\text{len}(U))$ over the draw of $K_{\text{in}}^i$.) Consider the $j$-th query for user $i$ and nonce $N$. Then, by the union bound, the probability that this query satisfies the condition with any other Tag query is at most $q_{\text{tag}}(\alpha\ell_{i,N,j}/2^{2k} + \beta/2^{k+n})$. By the union bound over all queries, we have

$$\Pr[\Lambda_{\text{id}} \in \Theta_{1,2}] \leq \sum_{i=1}^{u} \sum_{N \in \{0,1\}^{\nu}} \sum_{j=1}^{q_{i,N}} q_{\text{tag}} \left( \frac{\alpha\ell_{i,N,j}}{2^{2k}} + \frac{\beta}{2^{k+n}} \right)$$
$$\leq \frac{\alpha q_{\text{tag}} \sigma_{\text{tag}}}{2^{2k}} + \frac{\beta q_{\text{tag}}^2}{2^{k+n}}.$$

All in all,

$$\Pr[\Lambda_{\text{id}} \in \Theta_1] \leq \frac{\alpha(\mu - 1)\sigma_{\text{tag}}}{2^k} + \frac{\beta(\mu - 1)q_{\text{tag}}}{2^{n+1}} + \frac{\alpha q_{\text{tag}}\sigma_{\text{tag}}}{2^{2k}} + \frac{\beta q_{\text{tag}}^2}{2^{k+n}}.$$

CONDITION (C-2). We define two subsets $\Theta_{2,1}$ and $\Theta_{2,2}$ of $\Theta_2$. Subset $\Theta_{2,1}$ consists of transcripts $\tau$ such that there exist distinct queries $(i, N, U, V)$ and $(i, N', U', V')$ in $\tau_{\text{tag}}$ such that $N = N'$ and $V = V'$. Subset $\Theta_{2,2}$ consists of transcripts $\tau$ such that there exists $(i, N, U, V)$ and $(i', N', U', V')$ in $\tau_{\text{tag}}$ such that $i \neq i'$, $K_{\text{out}}^i = K_{\text{out}}^{i'}$, $N = N'$, and $V = V'$. We start with $\Theta_{2,1}$. For any two distinct $(i, N, U, V)$ and $(i, N', U', V')$ in $\tau_{\text{tag}}$, one has $V = V'$ with probability $2^{-n}$ since tags are uniformly random and independent in the ideal world. There are at most $(\mu - 1)q_{\text{tag}}/2$ unordered pairs of queries sharing the same nonce, hence by the union bound $\Pr[\Lambda_{\text{id}} \in \Theta_{2,1}] \leq (\mu - 1)q_{\text{tag}}/2^{n+1}$. Consider now $\Theta_{2,2}$. For any two distinct $(i, N, U, V)$ and $(i', N', U', V')$ in $\tau_{\text{tag}}$ such that $i \neq i'$, $K_{\text{out}}^i = K_{\text{out}}^{i'}$ with probability $2^{-k}$ and $V = V'$ with probability $2^{-n}$. Summing over the $q_{\text{tag}}(q_{\text{tag}} - 1)/2$ unordered pairs of queries, one has $\Pr[\Lambda_{\text{id}} \in \Theta_{2,2}] \leq q_{\text{tag}}^2/2^{k+n+1}$. All in all,

$$\Pr[\Lambda_{\text{id}} \in \Theta_2] \leq \frac{(\mu - 1)q_{\text{tag}}}{2^{n+1}} + \frac{q_{\text{tag}}^2}{2^{k+n+1}}.$$

CONDITION (C-3). For any $(i, N, U, V) \in \tau_{\text{tag}}$ and $(K, T, X, Y) \in \tau_{\text{ic}}$, the probability that $K_{\text{out}}^i = K$ is $2^{-k}$ and the probability that $H_{K_{\text{in}}^i}(U) = X$ is at most $\gamma(\text{len}(U)) = \alpha\text{len}(U)/2^k + \beta/2^n$. Summing over all pairs of queries, one has

$$\Pr[\Lambda_{\text{id}} \in \Theta_3] \leq q_{\text{ic}} \sum_{i=1}^{u} \sum_{N \in \{0,1\}^{\nu}} \sum_{j=1}^{q_{i,N}} \left( \frac{\alpha\ell_{i,N,j}}{2^{2k}} + \frac{\beta}{2^{k+n}} \right) \leq \frac{\alpha q_{\text{ic}}\sigma_{\text{tag}}}{2^{2k}} + \frac{\beta q_{\text{ic}}q_{\text{tag}}}{2^{k+n}}.$$

CONDITION $(C\text{-}4)$. We can view each query $(i, N, U, V) \in \tau_{\text{tag}}$ as throwing a ball $V$ uniformly at random into $2^n$ bins. In total, we throw $q_{\text{tag}} \leq 2^n$ balls and by Lemma 1 each bin contains at most $2n$ balls, except with probability at most $2^{-n}$. Then, for each query $(K, T, X, Y) \in \tau_{\text{ic}}$, there are at most $2n$ queries $(i, N, U, V) \in \tau_{\text{tag}}$ such that $V = Y$ and the probability that $K_{\text{out}}^i = K$ for one of those $2n$ queries is at most $2n/2^k$. Summing over all $q_{\text{ic}}$ ideal cipher queries yields

$$\Pr[\Lambda_{\text{id}} \in \Theta_4] \leq \frac{1}{2^n} + \frac{2n q_{\text{ic}}}{2^k}.$$

CONDITION $(C\text{-}5)$. We define two subsets $\Theta_{5,1}$ and $\Theta_{5,2}$ of $\Theta_5$. Subset $\Theta_{5,1}$ consists of transcripts $\tau$ such that there exist $(i, N, U, V) \in \tau_{\text{tag}}$ and $(i, N', U', V') \in \tau_{\text{ver}}$ such that $N = N'$, $H_{K_{\text{in}}^i}(U) = H_{K_{\text{in}}^i}(U')$, and $V = V'$. Subset $\Theta_{5,2}$ consists of transcripts $\tau$ such that there exist $(i, N, U, V) \in \tau_{\text{tag}}$ and $(i', N', U', V') \in \tau_{\text{ver}}$ such that $i \neq i'$, $K_{\text{out}}^i = K_{\text{out}}^{i'}$, $N = N'$, $H_{K_{\text{in}}^i}(U) = H_{K_{\text{in}}^{i'}}(U')$, and $V = V'$.

We start with $\Theta_{5,1}$. We fix any queries $(i, N, U, V) \in \tau_{\text{tag}}$ and $(i, N', U', V') \in \tau_{\text{ver}}$ and distinguish two cases. If the verification query $\text{VER}(i, N', U', V')$ was made *after* the tag query $\text{TAG}(i, N, U)$, then since we assumed that $\mathcal{A}$ does not make pointless queries, either $U \neq U'$ or $V \neq V'$ (otherwise we would have $(i, N', U', V') = (i, N, U, V)$). In the former case, $H_{K_{\text{in}}^i}(U) = H_{K_{\text{in}}^i}(U')$ with probability at most $\delta_{U,U'}$, while in the latter case the condition cannot be satisfied. If the verification query $\text{VER}(i, N', U', V')$ was made *before* the tag query $\text{TAG}(i, N, U)$, then $V$ is uniformly random and independent from $V'$ and hence $V = V'$ with probability $2^{-n}$. In both cases, the condition is satisfied with probability at most $\max\{\delta_{U,U'}, 2^{-n}\} = \delta_{U,U'}$ by our assumption that $\delta \geq 2^{-n}$. Fix some user $i$ and nonce $N$ and consider the $j$-th tag query and the $j'$-th verification query for $(i, N)$. Then the condition is satisfied for this pair of queries with probability at most

$$\frac{\alpha \max\{\ell_{i,N,j}, \ell'_{i,N,j'}\}}{2^k} + \frac{\beta}{2^n} \leq \frac{\alpha(\ell_{i,N,j} + \ell'_{i,N,j'})}{2^k} + \frac{\beta}{2^n} \leq \frac{\alpha(\ell_{\max}^{\text{tag}} + \ell'_{i,N,j'})}{2^k} + \frac{\beta}{2^n}.$$

By the union bound over all users $i$, nonces $N$, and pairs of tag and verification queries for $(i, N)$, we have

$$\begin{aligned}
\Pr[\Lambda_{\text{id}} \in \Theta_{5,1}] &\leq \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} \sum_{j=1}^{q_{i,N}} \sum_{j'=1}^{q'_{i,N}} \left( \frac{\alpha(\ell_{\max}^{\text{tag}} + \ell'_{i,N,j'})}{2^k} + \frac{\beta}{2^n} \right) \\
&\leq \frac{\alpha \mu \ell_{\max}^{\text{tag}}}{2^k} \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} q'_{i,N} + \frac{\alpha \mu}{2^k} \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} \sum_{j'=1}^{q'_{i,N}} \ell'_{i,N,j} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad + \frac{\beta \mu}{2^n} \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} q'_{i,N} \\
&\leq \frac{\alpha \mu \ell_{\max}^{\text{tag}} q_{\text{ver}}}{2^k} + \frac{\alpha \mu \sigma_{\text{ver}}}{2^k} + \frac{\beta \mu q_{\text{ver}}}{2^n}.
\end{aligned}$$

47

Consider now $\Theta_{5,2}$. For any $(i, N, U, V) \in \tau_{\text{tag}}$ and $(i', N', U', V') \in \tau_{\text{ver}}$ such that $i \neq i'$, $K_{\text{out}}^i = K_{\text{out}}^{i'}$ with probability $2^{-k}$ and $H_{K_{\text{in}}^i}(U) = H_{K_{\text{in}}^{i'}}(U')$ with probability at most $\gamma_{U,U'}$. Summing over all pairs of queries, we have

$$\Pr[\Lambda_{\text{id}} \in \Theta_{5,2}] \leq \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} \sum_{j=1}^{q_{i,N}} \sum_{i'=1}^{u} \sum_{N' \in \{0,1\}^\nu} \sum_{j'=1}^{q'_{i',N'}} \left( \frac{\alpha \ell'_{i',N',j'}}{2^{2k}} + \frac{\beta}{2^{k+n}} \right)$$

$$\leq \frac{\alpha q_{\text{tag}} \sigma_{\text{ver}}}{2^{2k}} + \frac{\beta q_{\text{tag}} q_{\text{ver}}}{2^{k+n}}.$$

All in all,

$$\Pr[\Lambda_{\text{id}} \in \Theta_5] \leq \frac{\alpha \mu \ell_{\max}^{\text{tag}} q_{\text{ver}}}{2^k} + \frac{\alpha \mu \sigma_{\text{ver}}}{2^k} + \frac{\beta \mu q_{\text{ver}}}{2^n} + \frac{\alpha q_{\text{tag}} \sigma_{\text{ver}}}{2^{2k}} + \frac{\beta q_{\text{tag}} q_{\text{ver}}}{2^{k+n}}.$$

CONDITION (C-6). For any $(i, N, U, V) \in \tau_{\text{ver}}$ and $(K, T, X, Y) \in \tau_{\text{ic}}$, $K_{\text{out}}^i = K$ with probability $2^{-k}$ and $H_{K_{\text{in}}^i}(U) = X$ with probability at most $\gamma(\text{len}(U))$. By summing over all pairs, one has

$$\Pr[\Lambda_{\text{id}} \in \Theta_6] \leq q_{\text{ic}} \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} \sum_{j=1}^{q'_{i,N}} \left( \frac{\alpha \ell'_{i,N,j}}{2^{2k}} + \frac{\beta}{2^{k+n}} \right) \leq \frac{\alpha q_{\text{ic}} \sigma_{\text{ver}}}{2^{2k}} + \frac{\beta q_{\text{ic}} q_{\text{ver}}}{2^{k+n}}.$$

Collecting all probabilities, we obtain

$$\Pr[\Lambda_{\text{id}} \in \Theta] \leq \underbrace{\frac{\alpha(\mu - 1)\sigma_{\text{tag}}}{2^k} + \frac{\beta(\mu - 1)q_{\text{tag}}}{2^{n+1}} + \frac{\alpha q_{\text{tag}} \sigma_{\text{tag}}}{2^{2k}} + \frac{\beta q_{\text{tag}}^2}{2^{k+n}}}_{\Theta_1}$$

$$+ \underbrace{\frac{(\mu - 1)q_{\text{tag}}}{2^{n+1}} + \frac{q_{\text{tag}}^2}{2^{k+n+1}}}_{\Theta_2} + \underbrace{\frac{\alpha q_{\text{ic}} \sigma_{\text{tag}}}{2^{2k}} + \frac{\beta q_{\text{ic}} q_{\text{tag}}}{2^{k+n}}}_{\Theta_3} + \underbrace{\frac{1}{2^n} + \frac{2n q_{\text{ic}}}{2^k}}_{\Theta_4}$$

$$+ \underbrace{\frac{\alpha \mu \ell_{\max}^{\text{tag}} q_{\text{ver}}}{2^k} + \frac{\alpha \mu \sigma_{\text{ver}}}{2^k} + \frac{\beta \mu q_{\text{ver}}}{2^n} + \frac{\alpha q_{\text{tag}} \sigma_{\text{ver}}}{2^{2k}} + \frac{\beta q_{\text{tag}} q_{\text{ver}}}{2^{k+n}}}_{\Theta_5}$$

$$+ \underbrace{\frac{\alpha q_{\text{ic}} \sigma_{\text{ver}}}{2^{2k}} + \frac{\beta q_{\text{ic}} q_{\text{ver}}}{2^{k+n}}}_{\Theta_6}$$

$$\leq \frac{2n q_{\text{ic}}}{2^k} + \frac{\alpha(\mu - 1)\sigma_{\text{tag}}}{2^k} + \frac{\alpha \mu \sigma_{\text{ver}}}{2^k} + \frac{\alpha \mu \ell_{\max}^{\text{tag}} q_{\text{ver}}}{2^k}$$

$$+ \frac{1}{2^n} + \frac{\beta(\mu - 1)q_{\text{tag}}}{2^n} + \frac{\beta \mu q_{\text{ver}}}{2^n}$$

$$+ \frac{\alpha(q_{\text{tag}} + q_{\text{ic}})(\sigma_{\text{tag}} + \sigma_{\text{ver}})}{2^{2k}} + \frac{\beta(2q_{\text{tag}} + q_{\text{ic}})(q_{\text{tag}} + q_{\text{ver}})}{2^{k+n}}.$$

$$\leq \frac{2n q_{\text{ic}}}{2^k} + \frac{\alpha \mu \sigma_{\text{tag}}}{2^k} + \frac{\alpha(\mu + 1)\sigma_{\text{ver}}}{2^k} + \frac{\alpha \mu \ell_{\max}^{\text{tag}} q_{\text{ver}}}{2^k}$$

$$+ \frac{1}{2^n} + \frac{\beta \mu q_{\text{tag}}}{2^n} + \frac{\beta(\mu + 1)q_{\text{ver}}}{2^n}, \tag{9}$$

48

where for the first transition we used $\beta \geq 1$ and for the second one we used $2q_{\text{tag}} + q_{\text{ic}} \leq 2^k$.

GOOD TRANSCRIPTS PROBABILITY RATIO. Let $\tau = (\tau_{\text{ic}}, \tau_{\text{tag}}, \tau_{\text{ver}}, \mathbf{K})$ be a good transcript and let $u$ be the length of $\mathbf{K}$. For any $(K, T) \in \{0,1\}^k \times \{0,1\}^t$, let $\mathcal{X}_{\text{ic}}(K, T)$ denote the set of inputs $X \in \{0,1\}^n$ such that there exists $Y$ with $(K, T, X, Y) \in \tau_{\text{ic}}$ and $\mathcal{Y}_{\text{ic}}(K, T)$ denote the set of outputs $Y \in \{0,1\}^n$ such that there exists $X$ with $(K, T, X, Y) \in \tau_{\text{ic}}$. Note that since $\tau$ is good, for any $(K, T, X) \in \{0,1\}^k \times \{0,1\}^t \times \{0,1\}^n$, there is at most one query $(i, N, U, V) \in \tau_{\text{tag}}$ such that $K = K_{\text{out}}^i$, $T = 0^{t-\nu}|N$, and $X = H_{K_{\text{in}}^i}(U)$ as otherwise condition (C-1) would be satisfied. For any $(K, T) \in \{0,1\}^k \times \{0,1\}^t$, define

$$\mathcal{X}_{\text{tag}}(K, T) \coloneqq \{X \in \{0,1\}^n \colon \exists (i, N, U, V) \in \tau_{\text{tag}} \colon$$
$$K = K_{\text{out}}^i \wedge T = 0^{t-\nu}\|N \wedge X = H_{K_{\text{in}}^i}(U)\}.$$

Note that by the previous observation, one has $\sum_{(K,T)} |\mathcal{X}_{\text{tag}}(K, T)| = q_{\text{tag}}$. Similarly, for any $(K, T, Y) \in \{0,1\}^k \times \{0,1\}^t \times \{0,1\}^n$, there is at most one query $(i, N, U, V) \in \tau_{\text{tag}}$ such that $K = K_{\text{out}}^i$, $T = 0^{t-\nu}|N$, and $Y = V$ as otherwise condition (C-2) would be satisfied. For any $(K, T) \in \{0,1\}^k \times \{0,1\}^t$, define

$$\mathcal{Y}_{\text{tag}}(K, T) \coloneqq \{Y \in \{0,1\}^n \colon \exists (i, N, U, V) \in \tau_{\text{tag}} \colon$$
$$K = K_{\text{out}}^i \wedge T = 0^{t-\nu}\|N \wedge Y = V\}.$$

Because $\tau$ is good, for any $(K, T)$, $\mathcal{X}_{\text{ic}}(K, T) \cap \mathcal{X}_{\text{tag}}(K, T) = \emptyset$ (as otherwise condition (C-3) would be satisfied) and $\mathcal{Y}_{\text{ic}}(K, T) \cap \mathcal{Y}_{\text{tag}}(K, T) = \emptyset$ (as otherwise condition (C-4) would be satisfied). Hence, for each pair $(K, T)$, $\tau_{\text{ic}}$ and $\tau_{\text{tag}}$ together impose a set of $|\mathcal{X}_{\text{ic}}(K, T)| + |\mathcal{X}_{\text{enc}}(K, T)|$ equations on the random permutation $E_{\text{ic}}(K, T, \cdot)$ internally sampled by the ideal cipher of the form $E_{\text{ic}}(K, T, X) = Y$ where all $X$'s are distinct and all $Y$'s are distinct. On the other hand, $\tau_{\text{ver}}$ imposes a set of $q_{\text{ver}}$ inequalities on $E_{\text{ic}}$ of the form $E_{\text{ic}}(K', T', X') \neq Y'$ (namely, every query $(i, N, U, V) \in \tau_{\text{ver}}$ translate to inequality $E_{\text{ic}}(K_{\text{out}}^i, 0^{t-\nu}|N, H_{K_{\text{in}}^i}(U)) \neq V$ that are "consistent" with previous equalities in the sense that for any equality $E_{\text{ic}}(K, T, X, Y)$ and inequality $E_{\text{ic}}(K', T', ', X') \neq Y'$, if $(K, T) = (K', T')$, then either $X \neq X'$ or $Y \neq Y'$ (otherwise condition (C-5) or (C-6) would be satisfied, depending on whether the equality stems from a TAG query or an IC query).

We say that a TBC $E \in \text{TBC}(k, t, n)$ is compatible with $\tau_{\text{ic}}$, $\tau_{\text{tag}}$, and $\tau_{\text{ver}}$ if

- for every $(K, T, X, Y) \in \tau_{\text{ic}}$, $E(K, T, X) = Y$,
- for every $(I, N, U, V) \in \tau_{\text{tag}}$, $E(K_{\text{out}}^i, 0^{t-\nu}\|N, H_{K_{\text{in}}^i}(U)) = V$,
- for every $(i, N, U, V) \in \tau_{\text{ver}}$, $E(K_{\text{out}}^i, 0^{t-\nu}\|N, H_{K_{\text{in}}^i}(U)) \neq V$,

and we let $\text{Comp}(\tau_{\text{ic}}, \tau_{\text{tag}}, \tau_{\text{ver}})$ denote the set of such TBCs and

$$p(\tau_{\text{ic}}, \tau_{\text{tag}}, \tau_{\text{ver}}) \coloneqq \Pr[E \leftarrow_\$ \text{TBC}(k, t, n) \colon E \in \text{Comp}(\tau_{\text{ic}}, \tau_{\text{tag}}, \tau_{\text{ver}})].$$

Then, according to Lemma 3 from [CLS17], one has

$$p(\tau_{\mathrm{ic}}, \tau_{\mathrm{tag}}, \tau_{\mathrm{ver}}) \geq \left( \prod_{\substack{K \in \{0,1\}^k \\ T \in \{0,1\}^t}} \prod_{i=0}^{q(K,T)-1} \frac{1}{2^n - i} \right) \cdot \left( 1 - \frac{q_{\mathrm{ver}}}{2^n - \max\{q(K,T)\}} \right)$$

$$\geq \left( \prod_{\substack{K \in \{0,1\}^k \\ T \in \{0,1\}^t}} \prod_{i=0}^{q(K,T)-1} \frac{1}{2^n - i} \right) \cdot \left( 1 - \frac{q_{\mathrm{ver}}}{2^n - q_{\mathrm{ic}} - \mu} \right)$$

where $q(K,T) := |\mathcal{X}_{\mathrm{ic}}(K,T)| + |\mathcal{X}_{\mathrm{tag}}(K,T)|$ and for the second inequality we used that $\max\{q(K,T)\} \leq q_{\mathrm{ic}} + \mu$. From this it follows that

$$\Pr[\Lambda_{\mathrm{re}} = \tau] \geq \frac{1}{|\mathcal{K}_{\mathrm{in}}|^u} \cdot \frac{1}{2^{uk}} \cdot p(\tau_{\mathrm{ic}}, \tau_{\mathrm{tag}}, \tau_{\mathrm{ver}}),$$

where the first term accounts for the random choice of keys $(K_{\mathrm{in}}, K_{\mathrm{out}})$ and the second term for the probability that $E_{\mathrm{ic}}$ satisfies the constraints imposed by the transcripts $\tau_{\mathrm{ic}}$, $\tau_{\mathrm{tag}}$, and $\tau_{\mathrm{ver}}$. On the other hand, since in the ideal world the $q_{\mathrm{tag}}$ tags are uniformly random and independent, one has

$$\Pr[\Lambda_{\mathrm{id}} = \tau] = \frac{1}{|\mathcal{K}_{\mathrm{in}}|^u} \cdot \frac{1}{2^{uk}} \cdot \frac{1}{2^{q_{\mathrm{tag}}n}} \cdot \left( \prod_{\substack{K \in \{0,1\}^k \\ T \in \{0,1\}^t}} \prod_{i=0}^{|\mathcal{X}_{\mathrm{ic}}(K,T)|-1} \frac{1}{2^n - i} \right)$$

Hence,

$$\frac{\Pr[\Lambda_{\mathrm{re}} = \tau]}{\Pr[\Lambda_{\mathrm{id}} = \tau]}$$

$$= 2^{q_{\mathrm{tag}}n} \cdot \left( \prod_{\substack{K \in \{0,1\}^k \\ T \in \{0,1\}^t}} \prod_{i=0}^{|\mathcal{X}_{\mathrm{tag}}(K,T)|-1} \frac{1}{2^n - |\mathcal{X}_{\mathrm{ic}}(K,T)| - i} \right) \left( 1 - \frac{q_{\mathrm{ver}}}{2^n - q_{\mathrm{ic}} - \mu} \right)$$

$$= \left( 1 - \frac{q_{\mathrm{ver}}}{2^n - q_{\mathrm{ic}} - \mu} \right) \cdot \prod_{\substack{K \in \{0,1\}^k \\ T \in \{0,1\}^t}} \prod_{i=0}^{|\mathcal{X}_{\mathrm{tag}}(K,T)|-1} \frac{2^n}{2^n - |\mathcal{X}_{\mathrm{ic}}(K,T)| - i}$$

$$\geq 1 - \frac{q_{\mathrm{ver}}}{2^n - q_{\mathrm{ic}} - \mu}, \tag{10}$$

where for the last inequality we used that $\sum_{(K,T)} |\mathcal{X}_{\mathrm{tag}}(K,T)| = q_{\mathrm{tag}}$.

CONCLUDING. The theorem follows by combining Theorem 1 with Equations (9) and (10) and using $q_{\mathrm{ic}} + \mu \leq 2^n/2$ and $\beta \geq 1$ which implies $q_{\mathrm{ver}}/(2^n - q_{\mathrm{ic}} - \mu) \leq 2\beta q_{\mathrm{ver}}/2^n$.

IMPROVED BOUND WHEN BOUNDING THE PER-USER DATA COMPLEXITY. Finally, we prove the "Moreover" part of the theorem. Assume that the total length of all TAG queries for any user is at most $B$. We modify how we upper bound $\Pr[\Lambda_{\mathrm{id}} \in \Theta_{5,1}]$. Indeed, since for any $(i, N)$, $\sum_{j=1}^{q_{i,N}} \ell_{i,N,j} \leq B$, one has

$$
\begin{aligned}
\Pr[\Lambda_{\mathrm{id}} \in \Theta_{5,1}] &\leq \sum_{i=1}^{u} \sum_{N \in \{0,1\}^{\nu}} \sum_{j=1}^{q_{i,N}} \sum_{j'=1}^{q'_{i,N}} \left( \frac{\alpha(\ell_{i,N,j} + \ell'_{i,N,j'})}{2^k} + \frac{\beta}{2^n} \right) \\
&\leq \frac{\alpha B}{2^k} \sum_{i=1}^{u} \sum_{N \in \{0,1\}^{\nu}} q'_{i,N} + \frac{\alpha \mu}{2^k} \sum_{i=1}^{u} \sum_{N \in \{0,1\}^{\nu}} \sum_{j'=1}^{q'_{i,N}} \ell'_{i,N,j} \\
&\qquad\qquad\qquad\qquad\qquad\qquad + \frac{\beta \mu}{2^n} \sum_{i=1}^{u} \sum_{N \in \{0,1\}^{\nu}} q'_{i,N} \\
&\leq \frac{\alpha B q_{\mathrm{ver}}}{2^k} + \frac{\alpha \mu \sigma_{\mathrm{ver}}}{2^k} + \frac{\beta \mu q_{\mathrm{ver}}}{2^n}.
\end{aligned}
$$

## D    Proof of Theorem 6

Assume towards a contradiction that $H[E]$ is not $\delta$-sAU, i.e., there exists $\ell$ and $(A, M)$ and $(A', M')$ in $\{0,1\}^{\leq L} \times \{0,1\}^{\leq L}$ with $(A, M) \neq (A', M')$ and $\max\{\mathsf{len}(A, M), \mathsf{len}(A', M')\} \leq \ell$ such that

$$
\delta' := \Pr[K \leftarrow_{\$} \mathcal{K} : H[E]_K(A, M) = H[E]_K(A', M')] > \delta.
$$

Let us show that there exists a non-empty set $\mathcal{S}$ of pairs $(T, X) \in \{0,1\}^t \times \{0,1\}^n$ of size at most $2\ell$ such that $H[E]_K(A, M) = H[E]_K(A', M')$ is equivalent to

$$
\sum_{(T,X) \in \mathcal{S}} E_K^T(X) = 0^n.
$$

The theorem will follow by defining $\mathcal{A}'$ as the adversary having this set $\mathcal{S}$ hardwired in its code and making oracle queries $\mathrm{ENC}(T, X)$ for all pairs $(T, X) \in \mathcal{S}$ except an arbitrary one $(T_0, X_0)$, computing $Y = \sum_{(T,X) \in \mathcal{S} \setminus (T_0, X_0)} \mathrm{ENC}(T, X)$, and returning $((T_0, X_0), Y)$ as "forgery". Then $\mathcal{A}'$ makes at most $2\ell$ oracle queries, runs in time at most $\alpha \ell$ for some small constant $\alpha$ independent of $E$, and wins the UNP game with advantage $\delta' > \delta(\ell)$, a contradiction.

Let us denote $\mathcal{S}_0$ and $\mathcal{S}_0'$ the set of pairs $(T, X)$ such that:

$$
H[E]_K(A, M) = \sum_{(T,X) \in \mathcal{S}_0} E_K^T(X), \text{ and } H[E]_K(A', M') = \sum_{(T,X) \in \mathcal{S}_0'} E_K^T(X).
$$

By definition of $H[E]$, the tweaks involved in the computation of each hash value are pairwise distinct, and both $\mathcal{S}_0$ and $\mathcal{S}_0'$ are non-empty. Thus, the only way for

the set $\mathcal{S}$ to be empty is that one has $\mathcal{S}_0 = \mathcal{S}_0'$, since the pairs that belong to $\mathcal{S}_0 \cap \mathcal{S}_0'$ will be the only one that will cancel each other. What remains to be shown is that there exists at least one pair $(T, X)$ such that either $(T, X) \in \mathcal{S}_0 \setminus \mathcal{S}_0'$, or $(T, X) \in \mathcal{S}_0' \setminus \mathcal{S}_0$. To this end, we distinguish several cases. If $(A, M) = \epsilon$, then $(A', M') \neq \epsilon$, and $H[E]_K(A, M) = E^{\langle 4 \rangle_3 | 0^{t-3}}(0^n)$. Moreover, no tweak with the prefix $\langle 4 \rangle_3$ can appear in $\mathcal{S}_0'$, which means that $(\langle 4 \rangle_3 | 0^{t-3}, 0^n) \in \mathcal{S}_0 \setminus \mathcal{S}_0'$; the case where $(A', M') = \epsilon$ can be treated similarly. We now assume that $(A, M), (A', M') \neq \epsilon$. Consider the case where $A \neq A'$. Let us denote $\ell_A$ (resp. $\ell_{A'}$) the length of $A$ (resp. $A'$) in $m + n$-bit block, and denote $A_i$ (resp. $A_i'$) the $i$-th $n + m$-bit block of $A$ (resp. $A'$). Several subcases can occur.

- $\ell_A > \ell_{A'}$ (the case where $\ell_A < \ell_{A'}$ can be treated similarly): the pair $(T, X) = (\langle i \rangle_3 || \langle \ell_A - 1 \rangle_c || \lceil A_{\ell_A - 1} \rceil_m, \lfloor A_{\ell_A - 1} \rfloor_n)$, where $i \in \{0, 1\}$ depending whether $A$ was padded or not, is the only one involving a tweak with the prefix $\langle i \rangle_3 || \langle \ell_A - 1 \rangle_c$, which means that $(T, X) \in \mathcal{S}_0 \setminus \mathcal{S}_0'$.
- $\ell_A = \ell_{A'}$, and there exists $i \in \{0, \ldots, \ell_A - 2\}$ such that $A_i \neq A_i'$: in that case, one has $(\langle 0 \rangle_3 || \langle i \rangle_c || \lceil A_i \rceil_m, \lfloor A_i \rfloor_n) \in \mathcal{S}_0 \setminus \mathcal{S}_0'$, since otherwise we would have $A_i = A_i'$.
- $\ell_A = \ell_{A'}$, and $A_i = A_i'$ for $i$ in $\{0, \ldots, \ell_A - 2\}$: either $|A_{\ell_A - 1}| \neq |A_{\ell_{A'} - 1}'|$, or $|A_{\ell_A - 1}| = |A_{\ell_{A'} - 1}'|$, and $A_{\ell_A - 1} \neq A_{\ell_{A'} - 1}'$. In both cases, one necessarily has $(T, X) = (\langle i \rangle_3 || \langle \ell_A - 1 \rangle_c || \lceil A_{\ell_A - 1} \rceil_m, \lfloor A_{\ell_A - 1} \rfloor_n) \in \mathcal{S}_0 \setminus \mathcal{S}_0'$. Indeed, in the former case, either both blocks are padded, in which case the padding rule implies that $\mathsf{ozp}(A_{\ell_A - 1}) \neq \mathsf{ozp}(A_{\ell_{A'} - 1}')$, or one block is padded and the other is not, which means that different tweak prefixes are used. In the latter case, the same tweak prefixes are used, but $(T, X) \in \mathcal{S}_0'$ would imply $A_{\ell_A - 1} = A_{\ell_{A'} - 1}'$.

Finally, if $A = A'$, then one necessarily has $M \neq M'$, and we can apply the same argument to this new case, which ends the proof of Theorem 6.

## E Security of GNSIV

In this section, we study the security of $\mathsf{GNSIV}$ in the case where two independent keys are used (one for hashing, and a second one for tag finalization and encryption). Slightly abusing our notation, we write $\mathsf{GNSIV}[E, f_T, f_X, H]$ for the mode of operation presented in Figure 7, where the hash function $H[E]$ defined in Section 4.2 is replaced by a generic hash function $H$. We prove the following result.

**Theorem 7 (mu security of GNSIV).** *Let $k$, $t$, and $n$ be positive integers, $\mathcal{K}_{\text{in}}$, $\mathcal{K}_{\text{out}}$, $\mathcal{A}$ and $\mathcal{M}$ be non-empty sets with $\mathcal{K}_{\text{in}}$, $\mathcal{K}_{\text{out}}$ finite, $\mathsf{len} \colon \mathcal{A} \times \mathcal{M} \to \mathbb{N}$ be some length function, $E \in \text{TBC}(k, t, n)$ be a tweakable block cipher modeled as an ideal tweakable cipher $(\text{IC}, \text{IC}^{-1})$, and $H \colon \mathcal{K}_{\text{in}} \times \mathcal{A} \times \mathcal{M} \to \{0, 1\}^n$ be a keyed hash function. Let $\nu$ be an integer such that $\nu \leq t$. Assume that $H$ is $\delta$-sAU and $\gamma$-uniform (w.r.t $\mathsf{len}$) for $\delta(\ell) = \gamma(\ell) = \alpha \ell / 2^k + \beta / 2^n$ with $\beta \geq 1$ (and hence $\delta \geq 2^{-n}$ and $\gamma \geq 2^{-n}$). Let $q_{\text{ic}}$, $q_{\text{enc}}$, $\sigma_{\text{enc}}$, $q_{\text{dec}}$, $\sigma_{\text{dec}}$, $\ell_{\max}^{\text{enc}}$, and $\mu$ be positive*

*integers such that $q_{\mathrm{enc}} \leq 2^n$, $2q_{\mathrm{enc}} + q_{\mathrm{ic}} \leq 2^k$, and $q_{\mathrm{ic}} + q_{\mathrm{enc}} + q_{\mathrm{dec}} \leq 2^n/2$. Then, for any (computationally unbounded) adversary $\mathcal{A}$ against the mu-nAE security of $\mathsf{GNSIV}[E, f_T, f_X, H]$ making at most $q_{\mathrm{ic}}$ queries in total to $\mathrm{IC}$ or $\mathrm{IC}^{-1}$, $q_{\mathrm{enc}}$ queries to $\mathrm{ENC}$ of total length (as measured by* len*) at most $\sigma_{\mathrm{enc}}$ and no query longer than $\ell_{\mathrm{max}}^{\mathrm{enc}}$, $q_{\mathrm{dec}}$ queries to $\mathrm{DEC}$ of total length (as measured by* len*) at most $\sigma_{\mathrm{dec}}$, and such that any (user, nonce) pair $(i, N)$ appears at most $\mu$ times in its $\mathrm{ENC}$ queries, one has*

$$\mathsf{Adv}_{\mathsf{GNSIV}[E, f_T, f_X, H]}^{\mathrm{mu\text{-}nae}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathsf{GCTR}[E, f_T, f_X]}^{\mathrm{mu\text{-}nive}}(n)(q_{\mathrm{ic}}, q_{\mathrm{enc}}, \sigma_{\mathrm{enc}}, \ell_{\mathrm{max}}^{\mathrm{enc}}, \mu)$$
$$+ \mathsf{Adv}_{\mathsf{NaT}[H, E]}^{\mathrm{mu\text{-}nprmac}}(n)(q_{\mathrm{ic}}, q_{\mathrm{enc}}, \sigma_{\mathrm{enc}}, q_{\mathrm{dec}}, \sigma_{\mathrm{dec}}, \ell_{\mathrm{max}}^{\mathrm{enc}}, \mu),$$

*where the first term is the bound from Theorem 2 with $\sigma_{\mathrm{enc}}$ and $\ell_{\mathrm{max}}^{\mathrm{enc}}$ substituted respectively to $\sigma$ and $\ell_{\mathrm{max}}$ and the second term is the bound from Theorem 3 with $q_{\mathrm{enc}}, \sigma_{\mathrm{enc}}, q_{\mathrm{dec}}, \sigma_{\mathrm{dec}}, \ell_{\mathrm{max}}^{\mathrm{enc}}$ substituted respectively to $q_{\mathrm{tag}}, \sigma_{\mathrm{tag}}, q_{\mathrm{ver}}, \sigma_{\mathrm{ver}}, \ell_{\mathrm{max}}^{\mathrm{tag}}$.*

The proof of Theorem 7 uses the H-coefficients technique. Fix a deterministic adversary $\mathcal{A}$ and assume without loss of generality that $\mathcal{A}$ makes exactly $q_{\mathrm{ic}}$ queries to $\mathrm{IC}$ or $\mathrm{IC}^{-1}$, $q_{\mathrm{enc}}$ queries to $\mathrm{ENC}$ and $q_{\mathrm{dec}}$ queries to $\mathrm{DEC}$, and that it never makes a pointless query, where a pointless query is either:

- a repeated query to $\mathrm{IC}$, $\mathrm{IC}^{-1}$, $\mathrm{ENC}$ or $\mathrm{DEC}$
- a query $\mathrm{IC}(K, T, X)$ if there was a previous query $\mathrm{IC}^{-1}(K, T, Y)$ that returned $X$ or a query $\mathrm{IC}^{-1}(K, T, Y)$ if there was a previous query $\mathrm{IC}(K, T, X)$ that returned $Y$,
- a query $\mathrm{ENC}(i, N, A, M)$ or $\mathrm{DEC}(i, N, A, V, C)$ such that $i > u$, where $u$ is the current value of the counter keeping track of NEW queries,
- a query $\mathrm{DEC}(i, N, A, V, C)$ if there was a previous query $\mathrm{ENC}(i, N, A, M)$ that returned $(V, C)$.

The transcript consists of three types of queries:

- queries to the ideal cipher that we record as a list $\tau_{\mathrm{ic}}$ containing tuples $(K, T, X, Y)$ such that $\mathcal{A}$ made either a query $\mathrm{IC}(K, T, X)$ that returned $Y$ or a query $\mathrm{IC}^{-1}(K, T, Y)$ that returned $X$;
- queries to $\mathrm{ENC}$ that we record as a list $\tau_{\mathrm{enc}}$ containing tuples $(i, N, A, M, V, C)$ such that $\mathcal{A}$ made a query $\mathrm{ENC}(i, N, A, M)$ that returned the tag $V$ and the ciphertext $C$;
- queries to $\mathrm{DEC}$ that we record as a list $\tau_{\mathrm{dec}}$ containing all tuples $(i, N, A, V, C)$ such that were queried to $\mathrm{DEC}$ (we do not keep track of the answers since we are interested in attainable transcripts, i.e., transcripts that can be obtained in the ideal world in which all queries to $\mathrm{DEC}$ return $\perp$).

We do not keep track explicitly of NEW queries, but when $\mathcal{A}$ has finished interacting with the oracles, we reveal all keys $\mathbf{K} = ((K_{\mathrm{in}}^1, K_{\mathrm{out}}^1), \ldots, (K_{\mathrm{in}}^u, K_{\mathrm{out}}^u))$ generated by calls to NEW, where $u$ is the final value of the counter keeping track of NEW queries. Besides, in the case where the length of an encryption query is not a multiple of $n$, we assume that, during the encryption pass, the last block

is padded with enough zeros so that it has length $n$ before returning the oracle answer. Before defining bad transcript, we can remark that, for any transcript $\tau$, we can derive a transcript $\tau_{\mathrm{tag}}$ of the interaction of $\mathcal{A}$ with the authentication pass of nAE as follows: $\tau_{\mathrm{tag}}$ contains the tuples $(i, N, A, M, V)$ for every query $(i, N, A, M, V, C)$ in $\tau_{\mathrm{enc}}$. We now define the following events:

- we say that $\tau$ is NaT-bad if $(\tau_{\mathrm{tag}}, \tau_{\mathrm{ic}}, \mathbf{K})$ satisfies one of the conditions (C-1) to (C-4) from Section 4;
- we say that $\tau$ is GCTR-bad if $(\tau_{\mathrm{enc}}, \tau_{\mathrm{ic}}, \mathbf{K})$ satisfies any of the conditions (C-1) to (C-4) from Section 3.

Now, assuming that $\tau$ is neither NaT-bad nor GCTR-bad, we are going to release additional information to the attacker as follows:

- in the real world, for every $(i, N, A, V, C)$ in $\tau_{\mathrm{dec}}$, we are going to release the corresponding plaintext $M$ by running the decryption algorithm of GNSIV;
- in the ideal world, we will release a dummy plaintext as follows: for every ciphertext block, if the corresponding $E(K, \langle 6 \rangle_3 \| f_T(N, V, j), (f_X(N, V, j))$ is already known (either from a query in $\tau_{\mathrm{ic}}$ or $\tau_{\mathrm{enc}}$), then the existing value is used; otherwise, we simulate what happens in the real world by drawing, uniformly at random and without replacement, an output in the set of values $Y$ that do not appear as outputs in $\tau_{\mathrm{ic}}$ or $\tau_{\mathrm{enc}}$ for the considered (key,tweak) pair (this is possible as $\tau$ is neither NaT-bad nor GCTR-bad)[16].

Note that this sampling is possible due to the fact that $\tau$ is neither NaT-bad nor GCTR-bad. After this step, queries from $\tau_{\mathrm{dec}}$ will appear as tuples $(i, N, A, M, V, C)$, where $M$ corresponds to the computed plaintext. If $\tau$ is NaT-bad or GCTR-bad, we simply assume that $M = \epsilon$ for all decryption queries.

A transcript $\tau$ will be said bad if one of the following condition is satisfied:

(C-1) $\tau$ is NaT-bad;

(C-2) $\tau$ is GCTR-bad;

(C-3) There exist queries $(i, N, A, M, V, C) \in \tau_{\mathrm{enc}}$ and $(i', N', A', M', V', C') \in \tau_{\mathrm{dec}}$ such that

$$\begin{cases} K_{\mathrm{out}}^i = K_{\mathrm{out}}^{i'} \\ N = N' \\ H_{K_{\mathrm{in}}^i}(A, M) = H_{K_{\mathrm{in}}^{i'}}(A', M') \\ V = V'. \end{cases}$$

(C-4) There exist queries $(i, N, A, M, V, C) \in \tau_{\mathrm{dec}}$ and $(K, T, X, Y) \in \tau_{\mathrm{ic}}$ such that

$$\begin{cases} K_{\mathrm{out}}^i = K \\ 0^{t-\nu} \| N = T \\ H_{K_{\mathrm{in}}^i}(A, M) = X \\ V = Y. \end{cases}$$

---

[16] Note that this is equivalent to sampling the output of an ideal cipher, conditioned on it being compatible with the one from the real world.

Otherwise, we say that $\tau$ is good and let $\Theta_{\mathrm{bad}}$, resp. $\Theta_{\mathrm{good}}$ denote the set of bad, resp. good transcripts. Note that the last two conditions are well-defined thanks to the fact that we release the plaintexts that correspond to every decryption query.

PROBABILITY OF BAD TRANSCRIPTS. First, we upper bound the probability of bad transcripts in the ideal world. We consider each condition in turn, letting $\Theta_i$ denote the set of transcripts satisfying condition (C-$i$), $i \in [\![1, 4]\!]$.

We assume that the queries transcript involves $u$ users, and for every $i \in [\![1, u]\!]$ and $N \in \{0, 1\}^{\nu}$ we let $q_{i,N}$ denote the number of ENC queries involving user $i$ and nonce $N$ (with $q_{i,N} = 0$ if there were no such queries). When $q_{i,N} \geq 1$, we also let $\ell_{i,N,j}$, $1 \leq j \leq q_{i,N}$, denote the length of the $j$-th ENC query for user $i$ with nonce $N$ and we assume that queries are reordered such that $\ell_{i,N,1} \leq \ell_{i,N,2} \leq \cdots \leq \ell_{i,N,q_{i,N}}$. Note that by our assumptions, we have

$$q_{i,N} \leq \mu \text{ for every } (i, N), \tag{11}$$

$$\sum_{i=1}^{u} \sum_{N \in \{0,1\}^{\nu}} q_{i,N} = q_{\mathrm{enc}}, \tag{12}$$

$$\text{and } \sum_{i=1}^{u} \sum_{N \in \{0,1\}^{\nu}} \sum_{j=1}^{q_{i,N}} \ell_{i,N,j} \leq \sigma_{\mathrm{enc}}. \tag{13}$$

Similarly, for every $i \in [\![1, u]\!]$ and $N \in \{0, 1\}^{\nu}$ we let $q'_{i,N}$ denote the number of DEC queries involving user $i$ and nonce $N$ (with $q'_{i,N} = 0$ if there were no such queries) and when $q'_{i,N} \geq 1$, we let $\ell'_{i,N,j}$, $1 \leq j \leq q'_{i,N}$, denote the length of the $j$-th DEC query for user $i$ with nonce $N$ and we assume that queries are reordered such that $\ell'_{i,N,1} \leq \ell'_{i,N,2} \leq \cdots \leq \ell'_{i,N,q'_{i,N}}$. Then

$$\sum_{i=1}^{u} \sum_{N \in \{0,1\}^{\nu}} q'_{i,N} = q_{\mathrm{dec}} \tag{14}$$

$$\text{and } \sum_{i=1}^{u} \sum_{N \in \{0,1\}^{\nu}} \sum_{j=1}^{q'_{i,N}} \ell'_{i,N,j} \leq \sigma_{\mathrm{dec}}. \tag{15}$$

In the following, for $((A, M), (A', M')) \in (\mathcal{A} \times \mathcal{M})^2$, we let

$$\delta_{(A,M),(A',M')} := \max\{\delta(\mathsf{len}(A, M)), \delta(\mathsf{len}(A', M'))\}$$
$$= \alpha \max\{\mathsf{len}(A, M), \mathsf{len}(A', M')\}/2^k + \beta/2^n$$
$$\gamma_{(A,M),(A',M')} := \min\{\gamma(\mathsf{len}(A, M)), \gamma(\mathsf{len}(A', M'))\}$$
$$= \alpha \min\{\mathsf{len}(A, M), \mathsf{len}(A', M')\}/2^k + \beta/2^n.$$

CONDITIONS (C-1) AND (C-2). As the random variable involved in these event have the same probability distribution in the ideal world as in Section 4 for (C-1) and Section 3 for (C-2), we can directly reuse the same upper bounds.

CONDITION (C-3). We define two subsets $\Theta_{3,1}$ and $\Theta_{3,2}$ of $\Theta_3$. Subset $\Theta_{3,1}$ consists of transcripts $\tau$ such that there exist queries $(i, N, A, M, V, C)$ in $\tau_{\text{enc}}$ and $(i, N', A', M', V', C')$ in $\tau_{\text{dec}}$ such that $N = N'$, $V = V'$ and $H_{K_{\text{in}}^i}(A, M) = H_{K_{\text{in}}^i}(A', M')$. Subset $\Theta_{1,2}$ consists of transcripts $\tau$ such that there exist queries $(i, N, A, M, V, C)$ in $\tau_{\text{enc}}$ and $(i', N', A', M', V', Y')$ in $\tau_{\text{dec}}$ such that $i \neq i'$, $K_{\text{out}}^i = K_{\text{out}}^{i'}$, $N = N'$, $V = V'$ and $H_{K_{\text{in}}^i}(A, M) = H_{K_{\text{in}}^{i'}}(A', M')$.

We start with $\Theta_{3,1}$. Fix $(i, N, A, M, V, C)$ in $\tau_{\text{enc}}$ and $(i, N', A', M', V', C')$ in $\tau_{\text{dec}}$ such that $N = N'$. Assume that the decryption query occurred after the encryption query, and that $V' = V$. Since $\mathcal{A}$ never makes pointless queries, then one has $(A', C') \neq (A, C)$. If $A \neq A'$, the probability that $H_{K_{\text{in}}^i}(A, M) = H_{K_{\text{in}}^i}(A', M')$ is smaller than $\delta_{(A,M),(A',M')}$. If $A = A'$, then one necessarily has $C' \neq C$. Since $N = N'$ and $V = V'$, this implies that $M \neq M'$, as $M$ and $M'$ are the plaintexts of respectively $C$ and $C'$ under the same IV and nonce value. Thus, the probability that $H_{K_{\text{in}}^i}(A, M) = H_{K_{\text{in}}^i}(A', M')$ is also smaller than $\delta_{(A,M),(A',M')}$. Now assume that the encryption query occurred after the decryption query. Then, the probability that $V = V'$ is exactly $2^{-n}$. Then the condition is satisfied for this pair of queries with probability at most

$$\frac{\alpha \max\{\ell_{i,N,j}, \ell'_{i,N,j'}\}}{2^k} + \frac{\beta}{2^n} \leq \frac{\alpha(\ell_{i,N,j} + \ell'_{i,N,j'})}{2^k} + \frac{\beta}{2^n} \leq \frac{\alpha(\ell_{\max}^{\text{enc}} + \ell'_{i,N,j'})}{2^k} + \frac{\beta}{2^n}.$$

By the union bound over all users $i$, nonces $N$, and pairs of encryption and decryption queries for $(i, N)$, we have, like in the study of the set $\Theta_{5,1}$ in the proof of Theorem 3,

$$\Pr[\Lambda_{\text{id}} \in \Theta_{3,1}] \leq \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} \sum_{j=1}^{q_{i,N}} \sum_{j'=1}^{q'_{i,N}} \left( \frac{\alpha(\ell_{\max}^{\text{enc}} + \ell'_{i,N,j'})}{2^k} + \frac{\beta}{2^n} \right)$$
$$\leq \frac{\alpha \mu \ell_{\max}^{\text{enc}} q_{\text{dec}}}{2^k} + \frac{\alpha \mu \sigma_{\text{dec}}}{2^k} + \frac{\beta \mu q_{\text{dec}}}{2^n}.$$

We now consider $\Theta_{3,2}$. Fix $(i, N, A, M, V, C)$ in $\tau_{\text{enc}}$ and $(i', N', A', M', V', C')$ in $\tau_{\text{dec}}$ such that $i \neq i'$. Then one clearly has $K_{\text{out}}^i = K_{\text{out}}^{i'}$ with a probability smaller than $2^{-k_2}$. Moreover, the probability that $H_{K_{\text{in}}^i}(A, M) = H_{K_{\text{in}}^{i'}}(A', M')$ is at most $\gamma_{(A,M),(A',M')}$. Summing over the at most $q_{\text{enc}} q_{\text{dec}}$ pairs of queries, we get

$$\Pr[\Lambda_{\text{id}} \in \Theta_{3,2}] \leq \frac{\alpha q_{\text{enc}} \sigma_{\text{dec}}}{2^{2k}} + \frac{\beta q_{\text{enc}} q_{\text{dec}}}{2^{k+n}}.$$

All in all,

$$\Pr[\Lambda_{\text{id}} \in \Theta_3] \leq \frac{\alpha \mu \ell_{\max}^{\text{enc}} q_{\text{dec}}}{2^k} + \frac{\alpha \mu \sigma_{\text{dec}}}{2^k} + \frac{\beta \mu q_{\text{dec}}}{2^n} + \frac{\alpha q_{\text{enc}} \sigma_{\text{dec}}}{2^{2k}} + \frac{\beta q_{\text{enc}} q_{\text{dec}}}{2^{k+n}}.$$

CONDITION (C-4). For any $(i, N, A, M, V, C) \in \tau_{\text{dec}}$ and $(K, T, X, Y) \in \tau_{\text{ic}}$, $K_{\text{out}}^i = K$ with probability $2^{-k_2}$ and $H_{K_{\text{in}}^i}(A, M) = X$ with probability at most

56

$\gamma(\mathsf{len}(A, M))$. By summing over all pairs, one has

$$\Pr[\Lambda_{\mathrm{id}} \in \Theta_4] \leq q_{\mathrm{ic}} \sum_{i=1}^{u} \sum_{N \in \{0,1\}^\nu} \sum_{j=1}^{q'_{i,N}} \left( \frac{\alpha \ell'_{i,N,j}}{2^{2k}} + \frac{\beta}{2^{k+n}} \right) \leq \frac{\alpha q_{\mathrm{ic}} \sigma_{\mathrm{dec}}}{2^{2k}} + \frac{\beta q_{\mathrm{ic}} q_{\mathrm{dec}}}{2^{k+n}}.$$

Collecting all probabilities, we obtain that $\Pr[\Lambda_{\mathrm{id}} \in \Theta]$ is the sum of the probabilities of bad transcripts in the proof of Theorem 2 and Theorem 3 (with substitutions as indicated in the theorem statement).

GOOD TRANSCRIPT PROBABILITY RATIO. Fix a good transcript $\tau$. We are going to divide the information about the ideal cipher that are contained in the transcript into several multisets:

- for each query to the ideal cipher, add a tuple $(K, T, X, Y)$ to $S_1$;
- for each encryption query $(i, N, A, M, V, C)$, add to $S_2$ the tuples

$$(K_{\mathrm{out}}^i, \langle 6 \rangle_3 \| f_T(N, V, j), f_X(N, V, j), M_j \oplus C_j),$$

where $M = M_1 \| \cdots \| M_{\ell_i}$, $C = C_1 \| \cdots \| C_{\ell_i}$, $j = 1, \ldots, \ell_i$, and $|M_k| = |C_k| = n$ for all $k = 1, \ldots, \ell_i$;

- for each encryption query $(i, N, A, M, V, C)$, add to $S_3$ the tuple

$$(K_{\mathrm{out}}^i, \langle 5 \rangle_3 \| 0^{t-3-\nu} \| N, H_{K_{\mathrm{in}}^i}(A, M), V);$$

- for each decryption query $(i, N, A, M, V, C)$, add to $S_4$ the tuples

$$(K_{\mathrm{out}}^i, \langle 6 \rangle_3 \| f_T(N, V, j), f_X(N, V, j), M_j \oplus C_j),$$

where $M = M_1 \| \cdots \| M_{\ell_i}$, $C = C_1 \| \cdots \| C_{\ell_i}$, $j = 1, \ldots, \ell_i$, and $|M_k| = |C_k| = n$ for all $k = 1, \ldots, \ell_i$, and as long as the tuple did not already belong to $S_1 \cup S_2 \cup S_4$;

- for each decryption query $(i, N, A, M, V, C)$, add to $S_5$ the tuple

$$(K_{\mathrm{out}}^i, \langle 5 \rangle_3 \| 0^{t-3-\nu} \| N, H_{K_{\mathrm{in}}^i}(A, M), V),$$

if it did not already belong to $S_5$.

Thanks to the fact that $\tau$ is a good transcript, the domain separation, and the way $S_4$ and $S_5$ are generated, then the multisets $S_1, \ldots, S_5$ do not contain any duplicated entries and are pairwise disjoint. Besides, the set $S_1 \cup S_2 \cup S_3 \cup S_4$ does not contain any distinct triples $(K, T, X, Y)$ and $(K', T', X', Y')$ such that $K = K'$, $T = T'$, and either $X = X'$ or $Y = Y'$.

Let us now compute the ratio $\Pr[\Lambda_{\mathrm{re}} = \tau] / \Pr[\Lambda_{\mathrm{id}} = \tau]$. The event $\Lambda_{\mathrm{id}} = \tau$ can be broken down into several events:

- `Key` corresponds to the event where the key vector $\mathbf{K}$ has the correct value;
- `IdealPrim` corresponds to the ideal cipher being compatible with $S_1$ (i.e. $\mathrm{IC}(K, T, X) = Y$ for every tuple $(K, T, X, Y)$ in $S_1$);

- `IdealEnc` corresponds to the ciphertexts and authentication tags indicated by $\tau$ being the actual values revealed (note that this corresponds to $n(|S_2|+|S_3|)$ bits);
- `IdealDec` corresponds to the sampled queries agreeing with $S_4$.

Similarly, the event $\Lambda_{\mathrm{re}} = \tau$ can be broken down into several events:

- `Key` corresponds to the event where the key vector $\mathbf{K}$ has the correct value;
- `RealPrim` corresponds to the ideal cipher being compatible with $S_1$;
- `RealEnc` corresponds to the ideal cipher being compatible with $S_2 \cup S_3$;
- `RealDec` corresponds to the ideal cipher being compatible with $S_4$;
- `RealVer` corresponds to the ideal cipher *not* being compatible with any tuple in $S_5$ (i.e. $\mathrm{IC}(K,T,X) \neq Y$ for every tuple $(K,T,X,Y)$ in $S_5$).

Note that the first event is common between both worlds (and independent from the second one),
$$\Pr[\texttt{RealPrim}] = \Pr[\texttt{IdealPrim}]$$
and

$$\Pr[\texttt{RealDec}|\texttt{Key} \cap \texttt{RealPrim} \cap \texttt{RealEnc}] =$$
$$\Pr[\texttt{IdealDec}|\texttt{Key} \cap \texttt{IdealPrim} \cap \texttt{IdealEnc}].$$

Moreover, after ordering $S_2 \cup S_3$ (with an arbitrary ordering), one has

$$\frac{\Pr[\texttt{RealEnc}|\texttt{Key} \cap \texttt{RealPrim}]}{\Pr[\texttt{IdealEnc}|\texttt{Key} \cap \texttt{IdealPrim}]} = \frac{2^{n(|S_2|+|S_3|)}}{\prod_{i=1}^{|S_2 \cup S_3|}(2^n - n_i)} \geq 1,$$

where $n_i$ denotes the number of occurrences of $(K_i, T_i)$ in the first $i-1$ elements of $S_2 \cup S_3$. Hence, one has

$$\frac{\Pr[\Lambda_{\mathrm{re}} = \tau]}{\Pr[\Lambda_{\mathrm{id}} = \tau]} \geq \Pr[\texttt{RealVer}|\texttt{RealDec} \cap \texttt{RealEnc} \cap \texttt{RealPrim} \cap \texttt{Key}]$$

$$\geq 1 - \frac{q_{\mathrm{dec}}}{2^n - q_{\mathrm{ic}} - q_{\mathrm{enc}} - q_{\mathrm{dec}}} \tag{16}$$

$$\geq 1 - \frac{2\beta q_{\mathrm{dec}}}{2^n} \tag{17}$$

where for the last inequality we used $q_{\mathrm{ic}} + q_{\mathrm{enc}} + q_{\mathrm{dec}} \leq 2^n/2$ and $\beta \geq 1$ Combining the probability of bad transcripts and Equation 17 with Theorem 1 concludes the proof of Theorem 7.