TEDT2 – Highly Secure Leakage-resilient TBC-based Authenticated Encryption

Eik List

Bauhaus-Universität Weimar, Weimar, Germany
<firstname>.<lastname>(at)uni-weimar.de

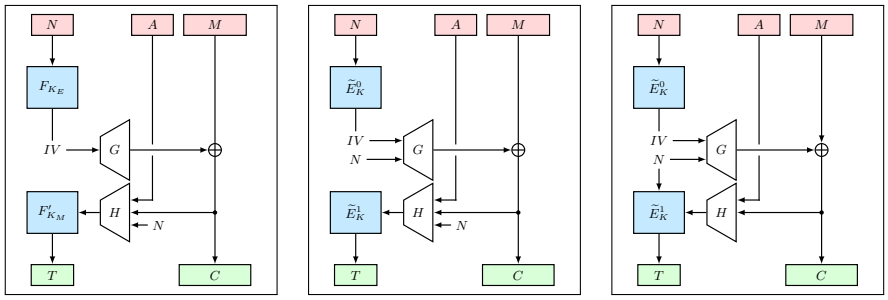
Abstract. Leakage-resilient authenticated encryption (AE) schemes received considerable attention during the previous decade. Two core security models of bounded and unbounded leakage have evolved, where the latter has been motivated in a very detailed and practice-oriented manner. In that setting, designers often build schemes based on (tweakable) block ciphers due to the small state size, such as the recent two-pass AE scheme TEDT from TCHES 1/2020. TEDT is interesting due to its high security guarantees of $O(n - \log(n^2))$-bit integrity under leakage and similar AE security in the black-box setting. Though, a detail limited it to provide only $n/2$-bit privacy under leakage.
In this work, we extend TEDT to TEDT2 in three aspects with the help of a tweakable block cipher with a $3n$-bit tweakey: we (1) adopt the idea from the design team of Romulus of replacing TEDT's previous internal hash function with Naito's MDPH, (2) move the nonce from the hash to the tag-generation function both for more efficiency, and (3) strengthen the security of the encryption to obtain beyond-birthday-bound security also under leakage.

Keywords: Symmetric-key cryptography · authenticated encryption · provable security · leakage resilience

1 Introduction

1.1 Leakage-resilient Authenticated Encryption

Authenticated encryption (AE) has been established as an invaluable cryptographic primitive [5,52] for various practical use cases that need the protection of both authenticity and the confidentiality of transmitted data. While the usual security notions treat the primitives as black boxes to the adversary, side channels [38,39] are a highly important threat to many systems. The protection of primitives against side-channel leakage – be it due to timing, memory accesses, power consumption, induced faults, or electromagnetic radiation – is usually left to the implementors and engineers. On a hardware level, the signal can be blurred by noise or special circuits, whereas on the implementation level, countermeasures include masking (i. e., secret sharing) [13,27] or shuffling [56]. Since side-channel protection is often inhibitive in terms of area, additional power consumption, and efficiency, a line of research has been devoted to developing leakage-resilient schemes. The interested reader can find in-depth surveys in [8,37].



(a) FGHF′ [18]. (b) TEDT [9]. (c) TEDT2.

Fig. 1: High-level comparison on existing two–pass designs and our proposal.

Schools of Thought. The literature on confidentiality with leakage could be categorized into three approaches: (1) "only computation leaks" (OCL) [43] with bounded leakage [26], (2) absence of oracles and hard-to-invert leakage [57], and (3) efficient simulatability of leakage [55]. The latter is still unsolved [40].

The former direction started with the framework by Barwell et al. [4] that contained notions capturing arbitrary non-adaptive leakage in the bounded-leakage setting. The approach has found widespread adoption, e.g. in [18,22,24]. Characteristic is that security is lacking while leakage occurs, but is guaranteed again once the leakage ends. Thus, schemes can provide nonce-misuse robustness in the sense of [53]. The second school of thought can be located by the group around Standaert. The school considers unbounded leakage with leveled implementations [48]. It has evolved stepwise with early focus on integrity [48], integrity with decryption leakage [11], and the composition with confidentiality [10], along to attempts to include misuse-resilience [9,30], to the recent summary at CRYPTO'20 [8]. In contrast to the bounded-leakage school, their notions cannot provide nonce-misuse resistance, but only resilience in the sense of [2].

Recent Schemes. The portfolio of leakage-resilient AE schemes has grown significantly recently, with focus on permutation-based designs like ISAP [19,20] and the generic Sponge and Duplex [18,22] in the OCL line of research. In contrast, the unbounded-leakage direction preferred leveled implementations with a few calls to a strongly protected primitive and the majority of computations to a more efficient, less protected primitive. Proposals using this approach often employed only a (tweakable) block cipher [9,30], but several permutation-based designs [7,12,31] with few calls to a protected block cipher followed. Thus, leveled implementations of permutation-based schemes are a viable option, as has been pointed out implicitly by [1] and explicitly by [8] and allows for efficient leakage-resilient implementations of lightweight standards such as Ascon [21]. Though, permutation-based schemes may employ a very small rate for the nonce absorption in the key-derivation phase, as in FGHF′ or ISAP [18,19]. Leveled block-cipher-based constructions such as TEDT are therefore interesting for potentially higher efficiency. Moreover, tweakable block ciphers (TBCs) could lead

Table 1: Comparison between existing (T)BC-based leakage-resilient AE schemes and our proposal. Security in bits, #primitive calls for messages of at most m n-bit blocks and at most a-block associated data. • = see CCAmL2, – = not available, (x) = probably x-bit security, but no proof is known, $^{(*)}$ = keyed hashing, $^{(\dagger)}$ = can be one call less depending on the hash-input length.

	Black-box bit Security			Leakage bit Security			#Primitive calls			
Scheme	CCA	CI	MR	CCAmL1	CCAmL2	CIML2	Enc	Hash	KDF	TGF
1 Pass										
TET [9]	(n)	(n)	–	$(n/2)$	–	(n)	$2m$	$2a$	1	1
AET-LR [28]	n	$n/2$	–	n	–	$(n/2)$	m	$a/2$ $^{(*)}$	1	1
2 Pass										
Romulus-LR-TEDT [35]	$n-\log(n^2)$	$n-\log(n^2)$	–	•	$n/2$	$n-\log(n^2)$	$2m$	$a+m+3^{(\dagger)}$	1	1
TEDT [9]	$n-\log(n^2)$	$n-\log(n^2)$	–	•	$n/2$	$n-\log(n^2)$	$2m$	$2a+2m+4$	1	1
TEDT2 **[This work]**	$n-\log(n)$	$n-\log(n)$	–	•	$n-\log(n)$	$n-\log(n)$	$2m$	$a+m+2^{(\dagger)}$	2	1
3 Pass										
FEMALE [30]	$n/2$	$n/2$	$n/2$	•	$n/2$	$n/2$	$4m$	$2a+2m+8$	2	1

to schemes with even smaller internal states than sponges (see, e.g., [45]). In this work, our focus will be on leakage-resilient TBC-based schemes, but acknowledging that they can share a similar high-level structure with permutation-based counterparts. Since the high-level view helps understand concepts, we will take a brief look at FGHF′ and TEDT in the following.

Based on the analysis of Encrypt-then-MAC under leakage by [4], Degabriele et al. [18] suggested FGHF′, where the acronym reflects the structure. The result of a key-derivation function F takes the nonce and produces an IV for a pseudorandom stream generator G. H hashes the resulting ciphertext, nonce, and associated data and forwards the output to a keyed function F' to generate the tag. The high-level structure is similar in other designs, e.g., ISAP or TEDT. The latter, which stands for Tweakable Encrypt-Digest-and-Tag is built from a TBC and comes with strong security guarantees, a single small-size primitive, and a single key. It has only a few structural differences compared to FGHF′: TEDT employs a TBC, an invertible tag-generation function for integrity under decryption leakage, and uses the nonce as IV to G, as illustrated in Figure 1.

1.2 Research Questions

TEDT is interesting for its efficiency and the leveled approach that spares the expensive protections for most primitive calls. However, the low-level view in the analysis by Guo et al. [30] may be hard to have a clear view of all details. We can identify three aspects of improvements, where we could use (1) a more efficient hash function, (2) the nonce in the finalization for more efficient authentication, and (3) a $2n$-bit tweakey in the encryption for higher security under leakage.

Firstly, TEDT employed Hirose's compression function with Merkle-Dåmgard strengthening [32] for hashing. Compared to TEDT, we can use Naito's proposal MDPH$[\widetilde{\pi}]$ from [44] that had also been suggested for Romulus-LR-TEDT [35] and AET-LR [28]. Like Romulus-LR-TEDT, we also suggest using a $3n$-bit TBC.

Thus, our proposal can process a $2n$-bit message block with each iteration of two primitive calls. Thus, the hash-function rate increases from 1/2 to 1.

Secondly, both FGHF′ and TEDT process all inputs to the public hash function H nonce, associated data, and ciphertext. A similar approach is followed in the instantiation of FGHF′ and in ISAP, which use the nonce as an initialization vector. Using a TBC with $3n$-bit tweakey, we can spare to process the nonce during hashing and use it in the tag-generation function (TGF) instead.

Thirdly, TEDT uses two primitive calls per message block: one to derive a new key for the subsequent block and one to produce a keystream block that is added to the current message block. The resulting rate-1/2 encryption provided $O(n - \log(n))$-bit security in the single-user black-box setting, but only $n/2$-bit security under leakage due to collisions and a hybrid argument. We generalize the encryption to use a larger tweak efficiently. Using a TBC based on the TWEAKEY framework by Jean et al. [36], we obtain a longer tweakey for higher security, whose encryption need two primitive calls for the tweakey update per message block. To compensate for the additional call, we use the tweakey for processing two message blocks. We obtain a more secure rate-1/2 construction that provides $O(n-\log(n))$-bit security in both the black-box setting and under leakage, where we adopt from TEDT the assumption that the distinguishing advantage for the XORs of the plain/ciphertexts with the PRG keystream does not endanger the security.

Outline. The remainder of this work is structured as follows: After Section 2 gives general preliminaries, Section 3 provides a design rationale of our improvements before Section 4 describes our proposal. Section 5 explains our used security model before Sections 6 and 7 summarize the results of the security analysis. Section 8 concludes this work. The analysis details are provided in Appendix C for the qCIML2 proof, and in Appendix D for the qCPA$mL2 proof of TEDT2 with RCTR.

2 Preliminaries

General Notations. We use uppercase characters for variables and functions, lowercase characters for indices, calligraphic characters ($\mathcal{X}$, $\mathcal{Y}$, ...) for sets and spaces, and bold characters ($\mathbf{X}$, $\mathbf{Y}$, ...) for vectors, matrices, and adversaries. For a non-negative integer x, we define $[x] =^{\text{def}} \{1, \ldots, x\}$ and $[0..x] =^{\text{def}} \{0, \ldots, x\}$. $\mathbb{F}_q^n$ denotes the n-dimensional extension of the field with characteristic q, where the elements of $\mathbb{F}_2^n$ can be represented as bit strings and ε is the empty string. For a list $\mathbf{L}$, we define [] as the empty list and $\mathbf{L} \xleftarrow{\cup} x$ denotes appending an element x to $\mathbf{L}$. Given sets $\mathcal{X}$ and $\mathcal{Y}$, we define $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$ for the set of all functions $F : \mathcal{X} \to \mathcal{Y}$, $\widetilde{\mathsf{Perm}}(\mathcal{T}, \mathcal{X})$ for the set of all tweakable permutations over $\mathcal{X}$, and $\mathsf{TBC}(\mathcal{K}, \mathcal{T}, \mathcal{X})$ for the sets of all tweakable block ciphers with key space $\mathcal{K}$ and tweak space $\mathcal{T}$ over $\mathcal{X}$. We use $\mathcal{X}^{\leq x} =^{\text{def}} \bigcup_{i=0}^{x} \mathcal{X}^i$. We define $X_1, X_2, \ldots \twoheadleftarrow \mathcal{X}$ for random uniform sampling X_1, X_2, ..., independently from each other and other samplings from $\mathcal{X}$. Furthermore, we define n-bit strings for arbitrary n

as $X = (X_{n-1}, \dots X_0)$ where X_i is the i-th least significant bit. We denote by $\text{MSB}_c(X)$ and $\text{LSB}_c(X)$ the c least significant bits of X.

Distinguishers. An adversary is a computationally unbounded algorithm that shall win a security game against a challenger. In this work, we focus on adversaries that are distinguishers. A distinguisher $\mathbf{A}$ is given access to one of two worlds and shall output a decision bit at the end of its interaction that shall denote which setting it interacted with. The challenger chooses one of the worlds by a fair coin toss at the start and provides $\mathbf{A}$ with access to either the real world $\mathcal{O}^1$ and an ideal world $\mathcal{O}^0$ with identical interfaces. We define

$$\underset{\mathbf{A}}{\Delta}(\underbrace{O_1^1, \dots, O_r^1}_{\mathcal{O}^1}; \underbrace{O_1^0, \dots, O_r^0}_{\mathcal{O}^0})(\mathbf{A}) \stackrel{\text{def}}{=} \left|\Pr\left[\mathbf{A}^{O_1^1,\dots,O_r^1} \Rightarrow 1\right] - \Pr\left[\mathbf{A}^{O_1^0,\dots,O_r^0} \Rightarrow 1\right]\right|,$$

where the probabilities are over the coins in the game, if any. Later, we will use labeled oracles, such as $\Delta_{\mathbf{A}}(\mathcal{E}_K, \mathcal{D}_K\,; \$, \perp)$, where we use $\perp$ as a function that always outputs the $\perp$ symbol as the indicator for a failed decryption: $\perp(X) = \perp$ for all X. We will use O_j to mean the j-th oracle in the sequence in each world, e.g., O_1 will refer to $\mathcal{E}_K$ or \$. We consider computationally unbounded distinguishers whose complexities are measured only by the number of queries to their oracles. Moreover, we assume that adversaries do not ask duplicate queries or queries to which they already know the answer. W.l.o.g., we focus on deterministic distinguishers since for any probabilistic distinguisher, there exists a deterministic one with at least the same success probability, cf. [23].

Notion Conventions. For a notion X, we write $\mathbf{Adv}^X_{\Pi}(\mathbf{A})$ for the advantage of $\mathbf{A}$ on some scheme Π. We define that $\mathbf{A}$ is a $(r_1, \dots, r_k)$-X-adversary for a notion X if $\mathbf{A}$ uses at most the resources $r_1, \dots, r_k$ (certain types of queries or blocks). We write $\mathbf{Adv}^X_{\Pi}(r_1, \dots, r_k) \stackrel{\text{def}}{=} \max_{\mathbf{A}}\left\{\mathbf{Adv}^X_{\Pi}(\mathbf{A})\right\}$ for the maximum advantage over all $(r_1, \dots, r_k)$-X-adversaries $\mathbf{A}$ on Π.

Query Restrictions. The security models we consider contain query restrictions that are necessary to prevent trivial wins of the adversary. We use $O_i \not\rightsquigarrow O_j$ to say that $\mathbf{A}$ must not ask the result of an earlier query to O_i in a later query to O_j. We write $O_{i,N} \not\rightsquigarrow O_{j,N}$ to indicate that $\mathbf{A}$ must not ask a query with a nonce N to $O_{j,N}$ if N was used in an earlier query to O_i. For sets of oracles $\mathcal{S}_i$, $\mathcal{S}_j$, we write $\mathcal{S}_i \not\rightsquigarrow \mathcal{S}_j$ for $O_i \not\rightsquigarrow O_j$ for each combination of $O_i, O_j \in \mathcal{S}_i \times \mathcal{S}_j$. For example, $O_1 \not\rightsquigarrow \{O_1, O_2\}$ means that a result from O_1 must not be used as input to O_1 or O_2. Similarly, we write $\mathcal{S}_i \not\leftsquigarrow \mathcal{S}_j$ that a query to any oracle in $\mathcal{S}_j$ must not have occured earlier to any oracle in $\mathcal{S}_i$. Finally, we denote as $O_i \upharpoonright O_j$ that O_j accepts only those queries that have been used earlier as queries to O_i. This will be useful for models with several leaking oracles that allow $\mathbf{A}$ to collect additional leakage traces for earlier queries.

Nonce-based Authenticated Encryption. Let $\mathcal{K}$, $\mathcal{N}$, $\mathcal{A}$, $\mathcal{M}$, $\mathcal{C}$, $\mathcal{T}$ be non-empty sets or spaces for keys, nonces, associated data, messages, ciphertexts,

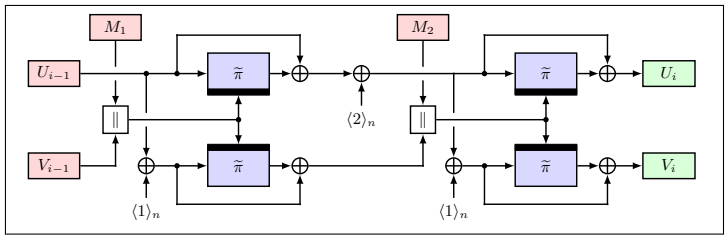


Fig. 2: Naito's hash function $\mathsf{MDPH}[\widetilde{\pi}]$ [44], based on the double-block-length compression function [32] and the MDP mode [33].

Table 2: Number of primitive calls of $\widetilde{\pi} \in \mathsf{TBC}(\mathbb{F}_2^n, \mathbb{F}_2^t, \mathbb{F}_2^n)$ in the hash functions. a and m denote the number of n-bit message blocks after padding each.

				$(a+m) \bmod 2$	
Scheme	Hash	MAC	t	0	1
TEDT [9]	Hirose [32]	HaT	n	$2a+2m+4$	
TEDT	Hirose [32]	HaT	$2n$	$a+m+4$	$a+m+5$
Romulus-LR-TEDT [35]	MDPH [33,44]	HaT	$2n$	$a+m+2$	$a+m+3$
TEDT2 **[This work]**	MDPH [33,44]	NHaT	$2n$	$a+m+2$	$a+m+1$

and tags, respectively. Following [46], a nonce-based AE (nAE) scheme consists of a pair of deterministic algorithms $\mathcal{E} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M} \to \mathcal{C} \times \mathcal{T}$ and $\mathcal{D} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \cup \{\perp\}$ for encryption and decryption, respectively. We assume correctness and tidiness: For all $K, N, A, M \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}$, it holds that $\mathcal{D}_K^{N,A}(\mathcal{E}_K^{N,A}(M)) = M$, and for all $(K, N, A, C, T) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T}$ where $\exists M \in \mathcal{M}$ s.t. $\mathcal{E}_K^{N,A}(M) = (C, T)$, it holds that $\mathcal{E}_K^{N,A}(\mathcal{D}_K^{N,A}(C, T)) = (C, T)$. The common notion is nAE security. $O_{1,N} \not\equiv O_{1,N}$ states that $\mathbf{A}$ must respect nonces.

Definition 1 (nAE Security [46]). Let $\Pi = (\mathcal{E}, \mathcal{D})$ be an nAE scheme and $K \twoheadleftarrow \mathcal{K}$. Then, the nAE advantage of an adversary $\mathbf{A}$ on Π is defined as $\mathbf{Adv}^{\mathsf{nAE}}_{\Pi_K}(\mathbf{A}) \stackrel{\text{def}}{=} \Delta_{\mathbf{A}}(\mathcal{E}_K, \mathcal{D}_K\,;\,\$, \perp)$, where $O_1 \not\rightarrow O_2$ and $O_{1,N} \not\equiv O_{1,N}$.

3 Design Rationale

This section describes our improvements for TEDT2. Prior, we briefly recall the necessary elements of TEDT.

TEDT. From a high-level perspective, TEDT encrypts a message M as

$$IV \leftarrow \widetilde{E}_K(N),\ C \leftarrow G[\widetilde{E}](IV, N) \oplus M,\ (U, V) \leftarrow H[\widetilde{E}](N, A, C),\ T \leftarrow \widetilde{E}_K^V(U)$$

under a secret key K and a nonce N. G and H are based on the same TBC $\widetilde{E} \in \mathsf{TBC}(\mathbb{F}_2^n, \mathbb{F}_2^n, \mathbb{F}_2^n)$, where G is a variant of the Bellare-Yee rekeying PRG [6]. In

contrast to FGHF′, TEDT uses an invertible tag-generation function F' inspired by [11]: instead of computing a leaking tag for a decryption query, the scheme inverts ${F'}^{-1}(T)$ and compares the output with the hash of nonce, associated data, and ciphertext. Since FGHF′ and other permutation-based schemes output only a fraction of the state as tag, they are usually not efficiently invertible.

Reducing the Hash Function. TEDT2 employs three ways for more efficient hashing compared to TEDT. It adopts the use of a TBC with $3n$-bit tweakey from AET-LR [28] and Romulus-LR-TEDT [35], processing $2n$ bits of message material by each hash-function iteration with two calls, saving half of the primitive calls. Moreover, it adopts "Merkle-Dåmgard with permutation", MDPH [33]. In [44], Naito showed its indifferentiability for up to $O(2^n/n)$ queries when instantiated with the compression function from [32]. The construction is illustrated for two blocks in Figure 2. Compared to [32] with Merkle-Dåmgard strengthening (MDS) [16,42], MDPH$[\widetilde{\pi}]$ spares a compression-function call and allows smaller key-tweak inputs than Hirose's compression function with MDS [32]. Finally, TEDT2 need not hash the nonce. If the number of n-bit blocks of the padded hash-function input is even, TEDT2 saves an iteration (i.e., two calls) for messages of random length on average, which is detailed in Table 2.

3.1 Strengthening the Authentication and Hashing More Efficiently

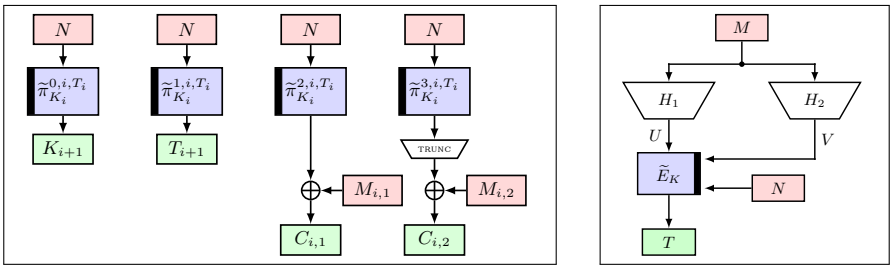
Unkeyed hashing avoids the need for strong leakage protection. Though, the absence of a key allows an offline adversary to evaluate the hash function separately. For authentication, TEDT employed a variant of Cogliati et al.'s MAC Hash-as-Tweak (HaT) [15], which provides n-bit security independent of nonces, but with unkeyed hashing. Given an $n+t$-bit hash, it uses the n-bit part as state and the t-bit part as the tweak in a tweakable block cipher to generate the tag. Since we have a TBC with an $(n+t)$-bit tweak, we can employ the nonce in the finalization. We call the resulting MAC Nonce-and-Hash-as Tweak (NHaT).

3.2 Strengthening The Encryption

The Encryption in TEDT. One iteration of the PRG G in TEDT computes

$$K_{i+1} = \widetilde{\pi}_{K_i}^{PK}(N \,\|\, \langle i \rangle_{\lfloor n/4 \rfloor - 1} \,\|\, 0) \quad \text{and} \quad C_i = \widetilde{\pi}_{K_i}^{PK}(N \,\|\, \langle i \rangle_{\lfloor n/4 \rfloor - 1} \,\|\, 1)\,,$$

where PK is a user-dependent public constant. G provides beyond-birthday-bound security in the black-box setting [9], but the bound is tight under leakage using a hybrid argument of the form $\sigma \cdot \mathbf{Adv}^{\mathsf{LUP\text{-}2}}_{F[\widetilde{\pi}]}(p, \sigma)$, where $F[\widetilde{\pi}]$ represents one iteration of the PRG, σ the total number of blocks by the adversary and p the number of leakage measures per iteration. This is the bottleneck of TEDT due to the n-bit key size since $\mathbf{Adv}^{\mathsf{LUP\text{-}2}}_{F[\widetilde{\pi}]}(p, \sigma) \in O\left(\sigma/2^n\right)$. Over all σ blocks of the adversary, the term leads to a birthday bound of $O\left(\sigma^2/2^n\right)$.



(a) RCTR$[\widetilde{\pi}]$ mode in TEDT2. **(b)** NHaT$[H_1, H_2, \widetilde{E}]$.

Fig. 3: Encryption (left) and tag-generation function (right) of TEDT2.

Modes. Our aim for a mode was to obtain n-bit security under leakage. We assume an ephemeral-key scheme with (1) n-bit CPA security under nonce-respecting adversaries and leakage and (2) unpredictability of the iteration in $O(\sigma/2^{2n})$ to allow the use of a hybrid argument in the CCA analysis, and (3) a rate of at least $1/2$ comparable with G in TEDT. For security, we suggest a $2n$-bit tweakey in a TWEAKEY-based primitive that treats both n-bit tweakey words similarly as secrets. During our studies, we considered five modes in total: (1) Generalized TET [9] (GTET), (2) Generalized FEMALE [29] (GFBE), (3) Rekeying counter mode (RCTR), (4) Rekeying OCB (ROCB), and (5) Rekeying OTR (ROTR). We study them in more detail in Appendix F. For TEDT2, we opted for the RCTR, which is illustrated for $r = 2$ message blocks per iteration in Figure 3a. Thus, an iteration needs two calls the the primitive to derive the subsequent tweakey (K_{i+1}, T_{i+1}) from the previous $2n$-bit tweakey (K_i, T_i). To compensate the additional primitive call without lowering the rate, we use the tweakey for two primitive calls to derive $(C_{i,1}, C_{i,2})$. While this provides a side-channel adversary with six instead of three traces, it must recover a $2n$-bit tweakey compared to TEDT. While GTET and GFBE have a higher rate of $r/(r+1)$, RCTR has the advantage of being a PRG, which simplifies the decryption and is a direct extension of the PRG in TEDT.

The tweakey could be expanded further to $3n$ or $4n$ bits, etc. given a primitive with a larger tweak at the cost of increased state size. Such primitives have been announced by Peyrin [49] for more efficient hashing. Such a primitive will be slightly slower for encryption, but more efficient if the rate can be increased further. Though, this should be considered in detail under the concrete side-channel analysis, which cannot be addressed satisfactorily in the present work.

4 Definition of TEDT2

Primitive and Domains. We instantiate TEDT2 with a tweakable block cipher $\widetilde{E} \in \mathsf{TBC}(\mathbb{F}_2^n, \mathbb{F}_2^{2n}, \mathbb{F}_2^n)$. Concretely, we suggest Skinny-64-192, Skinny-128-384, or Deoxys-BC-128-384. We assume a TWEAKEY-based block cipher where key and tweak words are treated (almost) equivalently and in a generalizable

Table 3: Domain parameters of TEDT2.

Part	Domains	Rationale
ENCRYPT	$\{0,1,2,3,4,5\}$	For key, tweak, full and partial message blocks
KDF	$\{6,7\}$	Two calls
TGF	$\{8\}$	

manner. The tweak allows us to have a single primitive for all occasions using domains for the different purposes of key derivation, encryption, hashing, and tag generation. We assume that key derivation and tag generation use strongly protected implementations of $\widetilde{E}$, e.g. against simple (SPA) and differential-power analysis (DPA), and all other calls to $\widetilde{E}$ use a less protected implementation, e.g. against only SPA (cf. [8]).

Sets and Primitive. Define positive integers $k = \tau = n$, $d = 4$, and $\nu = n - d$. Let $\mathcal{K} = \mathbb{F}_2^k$, $\mathcal{N} = \mathbb{F}_2^\nu$, $\mathcal{A} = \mathbb{F}_2^{\leq n \cdot a_{\max}}$, $\mathcal{M} = \mathcal{C} = \mathbb{F}_2^{\leq n \cdot m_{\max}}$, and $\mathcal{T} = \mathbb{F}_2^\tau$ be spaces for keys, nonces, associated data, messages, ciphertexts, and authentication tags, respectively. We define a domain space $\mathcal{D} = \mathbb{F}_2^d$ and a compound tweak space $\mathcal{T}_D = \mathcal{D} \times \mathcal{T}_1 \times \mathcal{T}_2 = \mathbb{F}_2^{2n}$, where $\mathcal{T}_1 = \mathbb{F}_2^{n-d}$ and $\mathcal{T}_2 = \mathbb{F}_2^n$. Thus, we define the nonce space as $\mathbb{F}_2^{n-d}$, to have d bits for the domain. We use the domains from Table 3 encoded as d-bit integers, e.g., $\langle 8 \rangle_d = \texttt{1000}$. We will often use block indices, where we assume that they are encoded as $n-d$-bit integers, like domains. We define TEDT2 for at most $a_{\max} = 2^{n/2}$ n-bit blocks of associated data and at most $m_{\max} = 2^{n/2}$ blocks per message and at most $2^{n/2}$ messages.

Encryption and Decryption. The encryption $\mathcal{E}[\widetilde{E}]_K$ expects a nonce, associated data, and message $(N, A, M) \in \mathcal{N} \times \mathcal{A} \times \mathcal{M}$ and encrypts M under a key $K \in \mathcal{K}$ and the nonce to a tuple of ciphertext $C \in \mathcal{C}$ and tag $T \in \mathcal{T}$ such that $|M| = |C|$ and returns (C, T). The decryption algorithm $\mathcal{D}[\widetilde{E}]_K$ expects a nonce, associated data, ciphertext, and a tag $(N, A, C, T) \in \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T}$. If the tuple is deemed invalid, the decryption outputs $\perp$. Otherwise, it decrypts C under the key $K \in \mathcal{K}$ and the nonce to the single possible message $M \in \mathcal{M}$ such that $\mathcal{E}[\widetilde{E}]_K^{N,A}(M) = (C, T)$ and outputs M. The algorithms are correct and tidy. Algorithm 1 defines the encryption and decryption procedures.

5 Security Model

Comparison to [9,29,30]. We follow the framework of unbounded leakage under oracle-free hard-to-invert leakage functions [9,30] since it captures leakage in all queries. The notions follow a convention of [PI,CI,CPA,CCA][m,M,-][L⟨i⟩] (for plaintext/ciphertext integrity, chosen-plain-/ciphertext attack), where m means nonce-misuse resilience, i.e., nonces may repeat except in challenge queries. L⟨i⟩ indicates leakage in i oracles; L2 means leakage in en- and decryption. Though, we differ in three minor aspects from their notions.

Algorithm 1 Definition of TEDT2.

11: **function** $\mathcal{E}[\widetilde{E}]_K^{N,A}(M)$
12: $K_E \leftarrow \mathrm{KDF}[\widetilde{E}]_K(N)$
13: $C \leftarrow \textsc{Encrypt}[\widetilde{E}]_{K_E}(M)$
14: $T \leftarrow \mathrm{TGF}[\widetilde{E}]_K(N, A, C)$
15: **return** (C, T)

16: **function** $\mathrm{KDF}[\widetilde{E}]_K(N)$
17: **return** $\widetilde{E}_K^{6,0,N}(0), \widetilde{E}_K^{7,0,N}(0)$

20: **function** $\mathrm{TGF}[\widetilde{E}]_K(N, A, C)$
21: $X \leftarrow \textsc{concat}_n(A, C)$
22: $(U, V) \leftarrow \textsc{Hash}[\widetilde{E}](X)$
23: **return** $\widetilde{E}_K^{8,N,V}(U)$

25: **function** $\textsc{Encrypt}[\widetilde{E}]_K^N(M)$
26: $(K_1, T_1) \leftarrow K$
27: $(M_1, \ldots, M_m) \xleftarrow{2n} M$
28: **for** $i \leftarrow 1..m-1$ **do**
29: $K_{i+1} \leftarrow \widetilde{E}_{K_i}^{0,i,T_i}(N)$
30: $T_{i+1} \leftarrow \widetilde{E}_{K_i}^{1,i,T_i}(N)$
31: $S_i \leftarrow \widetilde{E}_{K_i}^{2,i,T_i}(N) \,\|\, \widetilde{E}_{K_i}^{3,i,T_i}(N)$
32: $C_i \leftarrow S_i \oplus M_i$
33: $(d_1, d_2) \leftarrow \textsc{getDomainForM}(|M_m|)$
34: $S_m \leftarrow \widetilde{E}_{K_m}^{d_1,m,T_m}(N) \,\|\, \widetilde{E}_{K_m}^{d_2,m,T_m}(N)$
35: $C_m \leftarrow \textsc{trunc}_{|M_m|}(S_m) \oplus M_m$
36: **return** $(C_1 \,\|\, \cdots \,\|\, C_m)$

41: **function** $\textsc{concat}_n(X, Y)$
42: $X^* \leftarrow \textsc{padzeroes}_n(X)$
43: $Y^* \leftarrow \textsc{padzeroes}_n(Y)$
44: $L \leftarrow \langle |X| \rangle_{n/2} \,\|\, \langle |Y| \rangle_{n/2}$
45: **return** $X^* \,\|\, Y^* \,\|\, L$

46: **function** $\textsc{padzeroes}_x(X)$
47: $\ell \leftarrow |X| \bmod x$
48: **if** $\ell \equiv 0$ **then return** X
49: **return** $X \,\|\, 0^{x-\ell}$

51: **function** $\mathcal{D}[\widetilde{E}]_K(N, A, C, T)$
52: $K_E \leftarrow \mathrm{KDF}[\widetilde{E}]_K(N)$
53: **if** $\textsc{Verify}[\widetilde{E}]_K(N, A, C, T)$ **then**
54: **return** $\textsc{Decrypt}[\widetilde{E}]_{K_E}^N(C)$
55: **return** $\perp$

56: **function** $\textsc{Verify}[\widetilde{E}]_K(N, A, C, T)$
57: $X \leftarrow \textsc{concat}_n(A, C)$
58: $(U, V) \leftarrow \textsc{Hash}[\widetilde{E}](X)$
59: $U' \leftarrow \widetilde{D}_K^{8,N,V}(T)$
60: **return** $U = U'$

61: **function** $\textsc{Decrypt}[\widetilde{E}]_K^N(C)$
62: **return** $\textsc{Encrypt}[\widetilde{E}]_K^N(C)$

66: **function** $\textsc{Hash}[\widetilde{E}](M)$
67: $(M_{1,1}, M_{1,2}, \ldots, M_{m,1}, M_{m,2}) \xleftarrow{n} M$
68: $(U_0, V_0) \leftarrow (0^n, 0^n)$
69: **for** $i \leftarrow 1..m$ **do**
70: **if** $i = m$ **then** $U_{i-1} \leftarrow U_{i-1} \oplus \langle 2 \rangle$
71: $K_i \leftarrow V_{i-1}$
72: $T_i \leftarrow M_{i,1} \,\|\, M_{i,2}$
73: $W_i \leftarrow U_{i-1} \oplus \langle 1 \rangle$
74: $U_i \leftarrow \widetilde{E}_{K_i}^{T_i}(U_{i-1}) \oplus U_{i-1}$
75: $V_i \leftarrow \widetilde{E}_{K_i}^{T_i}(W_i) \oplus W_i$
76: **return** (U_m, V_m)

81: **function** $\textsc{trunc}_x(X)$
82: **if** $|X| \le x$ **then return** X
83: **return** $\textsc{msb}_x(X)$

86: **function** $\textsc{pad}_x(X)$
87: $\ell \leftarrow (|X| + 1) \bmod x$
88: **if** $\ell \equiv 0$ **then return** $X \,\|\, 1$
89: **return** $X \,\|\, 1 \,\|\, 0^{x-\ell}$

91: **function** $\textsc{getDomainForM}(\ell)$
92: **if** $\ell = 2n$ **then return** $(2, 3)$
93: **else if** $n \le \ell \wedge \ell < 2n$ **then return** $(2, 5)$
94: **else return** $(4, 5)$

First, the notions from [30] used only a single challenge query, where CCAmL2 was extended to a multi-challenge variant in [29]. We will use multi-challenge but single-user notions (and denote this by a q) throughout this work since they are much more common and make our results comparable with those for TEDT, which was proven under multi-challenge notions muCIML2 and muCCAmL2 [9].

Second, we replace the left-or-right style for confidentiality with a real-or-random style, where the ideal world samples a message at random. We make this explicit by a -$ in the notions. While left-or-right and real-or-random notions are roughly equally strong, the latter seems more natural for avoiding dependencies on how an adversary chooses alternative messages. We stress that our real-or-random definitions only sample the message at random but process it with the

Algorithm 2 The qCIML2 experiment, adapted from [9] and [29,30].

```
11: procedure INITIALIZE
12:     K ↞ 𝒦; Q ← ∅; b ← {0,1}
21: function FINALIZE(b')
22:     return b = b'
31: function Ê_K(N, A, M, Λ)
32:     R ↞ ℛ
33:     (C, T) ← E_K^{N,A}(M)
34:     Q ∪← {(N, A, C, T)}
35:     L ← Λ_K^{N,A}(M; R)
36:     return (C, T, L)
41: function D̂_K(N, A, C, T, Λ)
42:     R ↞ ℛ
43:     M ← D_K^{N,A}(C, T)
44:     L ← Λ_K^{N,A}(C, T; R)
45:     return (M, L)
51: function D̂^ch_K(N, A, C, T, Λ)
52:     R ↞ ℛ
53:     L ← Λ_K^{N,A}(C, T; R)
54:     if (N, A, C, T) ∈ Q ∨ b = 0 then return (⊥, L)
55:     M ← D_K^{N,A}(C, T)
56:     return (M, L)
```

same construction and key; they do not define an abstract ideal without the real construction since leakage of idealized objects is difficult to define (cf. [9,30,54]).

Third, we focus on information-theoretic distinguishers whose resources are bounded only by the numbers of queries and bits/blocks to the available oracles. Complexity-theoretic results can be derived in a straightforward manner.

We will write notions as distinguishing games. Note that we will usually add primitive oracles similar as in the ideal-cipher model in [9,30].

Leakage Functions. We inherit three usual assumptions that leakage functions Λ are (1) probabilistic, (2) oracle-free, and (3) not efficiently invertible from the notions of [9,30]. We use a non-empty random-coin set $\mathcal{R}$ and sample $R \twoheadleftarrow \mathcal{R}$ for every call to a leakage function (cf. [4]), which ensures that repeated calls may result in different leakage traces. We denote by $[L_i]_p = (L_1, \ldots, L_p)$ a p-element list of leakages L_i from the same leakage function Λ collected under independent random coins. Since leakage functions chosen by an adversary could compute some state in the future, they are usually prohibited from calling the primitives and are therefore called oracle-free (cf. [57]) to prevent future-computation attacks, where $\Lambda(K_i, \ldots)$ might otherwise leak outputs about K_j for a later occurring key $i < j$, which would render any confidentiality goal unachievable. This model reflects practice where leakages are a function of the primitive's in- and outputs. Moreover, we assume that leakage functions are not efficiently invertible in the sense of exponentially hard-to-invert functions [25].

We assume that leakage-function sets are used for the queries corresponding to their subscripts, i.e., $\mathcal{L}_E$ and $\mathcal{L}_D$ correspond $\mathcal{E}$ and $\mathcal{D}$ oracle(s), respectively. We mark leaking oracles by a hat, i.e. the leaking variant of $\mathcal{E}_K$ is $\widehat{\mathcal{E}}_K$, where leaking means that $\widehat{\mathcal{E}}_K$ takes a leakage function $\Lambda \in \mathcal{L}_E$ as an additional parameter that is called with the remaining parameters of $\mathcal{E}_K$ and random coins.

5.1 Notion for Authenticity

qCIML2 is a single-user variant of muCIML2 [9] and a distinguisher version of CIML2; every forgery allows distinguishing. qCIML2, as defined in Algorithm 2,

Algorithm 3 The qCCA$mL2 experiment, adapted from [9] and [29,30].

11: **procedure** INITIALIZE
12: $K \twoheadleftarrow \mathcal{K}$; $b \twoheadleftarrow \{0,1\}$
13: $\mathcal{Q}_N \leftarrow \emptyset$; $\mathcal{Q} \leftarrow \emptyset$
14: $\mathcal{Q}_N^{ch} \leftarrow \emptyset$; $\mathcal{Q}^{ch} \leftarrow \emptyset$

21: **function** $\widehat{\mathcal{E}}_K(N, A, M, \Lambda)$
22: **if** $N \in \mathcal{Q}_N^{ch}$ **then return** $\perp$
23: $R \twoheadleftarrow \mathcal{R}$
24: $(C, T) \leftarrow \mathcal{E}_K^{N,A}(M)$
25: $\mathcal{Q} \xleftarrow{\cup} \{(N, A, C, T)\}$; $\mathcal{Q}_N \xleftarrow{\cup} \{N\}$
26: $L \leftarrow \Lambda_K^{N,A}(M; R)$
27: **return** (C, T, L)

31: **function** $\widehat{\mathcal{D}}_K(N, A, C, T, \Lambda)$
32: **if** $(N, A, C, T) \in \mathcal{Q}^{ch}$ **then return** $\perp$
33: $R \twoheadleftarrow \mathcal{R}$
34: $M \leftarrow \mathcal{D}_K^{N,A}(C, T)$
35: $L \leftarrow \Lambda_K^{N,A}(C, T; R)$
36: **return** (M, L)

41: **function** FINALIZE(b')
42: **return** $b = b'$

51: **function** $\widehat{\mathcal{E}}_K^{ch}(N, A, M, \Lambda)$
52: **if** $N \in \mathcal{Q}_N^{ch} \vee N \in \mathcal{Q}_N \vee M = \varepsilon$ **then**
53: **return** $\perp$
54: $M^* \leftarrow M$
55: $R \twoheadleftarrow \mathcal{R}$
56: **if** $b = 0$ **then** $M^* \twoheadleftarrow \mathbb{F}_2^{|M|}$
57: $(C, T) \leftarrow \mathcal{E}_K^{N,A}(M^*)$
58: $\mathcal{Q}^{ch} \xleftarrow{\cup} \{(N, A, C, T)\}$; $\mathcal{Q}_N^{ch} \xleftarrow{\cup} \{N\}$
59: $L \leftarrow \Lambda_K^{N,A}(C, T; R)$
60: **return** (C, T, L)

66: **function** $\widehat{\mathcal{D}}_K^{ch}(N, A, C, T, \Lambda)$
67: **if** $(N, A, C, T) \in \mathcal{Q}^{ch}$ **then return** $\perp$
68: $R \twoheadleftarrow \mathcal{R}$
69: $L \leftarrow \Lambda_K^{N,A}(C, T; R)$
70: **return** L

splits the decryption queries into two oracles, one for collecting decryption leakage from earlier encryption queries, and one as a challenge oracle. Note that $\widehat{\perp}$ always outputs the decryption $\perp$, but also outputs decryption leakage.

Definition 2 (qCIML2). Let $\Pi = (\mathcal{E}_K, \mathcal{D}_K)$ be an nAE scheme, $K \twoheadleftarrow \mathcal{K}$, and $\mathcal{L}_E$ and $\mathcal{L}_D$ be sets of leakage functions. Let $\mathbf{A}$ be an adversary on Π. Then, the qCIML2 advantage of $\mathbf{A}$ on Π is defined as

$$\mathbf{Adv}^{\mathsf{qCIML2}}_{\mathcal{E}_K,\mathcal{D}_K,\mathcal{L}_E,\mathcal{L}_D}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\widehat{\mathcal{E}}_K, \widehat{\mathcal{D}}_K, \widehat{\mathcal{D}}_K^{ch}\,;\, \widehat{\mathcal{E}}_K, \widehat{\mathcal{D}}_K, \widehat{\perp}) \quad \text{where} \quad O_1 \not\rightsquigarrow O_3\,.$$

5.2 Notions for Confidentiality

We define qCCA$mL2 in Algorithm 3 as a real-or-random variant of mCCAmL2 [29] or as a single-user variant of muCCAmL2 [9]. In the ideal world, the challenge encryption oracle $\widehat{\$}_{\mathcal{E}}$ encrypts a random string of $|M|$ bits with the real construction to produce a ciphertext and leakage. The challenge decryption oracle is defined similarly as $\widehat{\$}_{\mathcal{D}}$. Though, in both worlds, $\widehat{\$}_{\mathcal{D}}$ accepts only queries that were previous outputs from $\widehat{\$}_{\mathcal{E}}$ (denoted as $O_3 \uparrow O_4$) and outputs only the corresponding decryption leakages and not the decryption to avoid trivial wins.

qCPA$mL2 is a real-or-random variant of the mCPAmL2 notion by [29], which differs from qCCA$mL2 in the fact that it has no non-challenge decryption oracle. Thus, it is defined in Algorithm 3 without Lines 31–36.

Definition 3 (qCCA$mL2 and qCPA$mL2). Let $\Pi = (\mathcal{E}_K, \mathcal{D}_K)$ be an nAE scheme, $K \twoheadleftarrow \mathcal{K}$, and $\mathcal{L}_E$ and $\mathcal{L}_D$ be sets of leakage functions. Then, the qCCA$mL2 advantage of an adversary $\mathbf{A}$ on Π is defined as

$$\mathbf{Adv}^{\mathsf{qCCA\$mL2}}_{\mathcal{E}_K,\mathcal{D}_K,\mathcal{L}_E,\mathcal{L}_D}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\widehat{\mathcal{E}}_K, \widehat{\mathcal{D}}_K, \widehat{\mathcal{E}}_K^{ch}, \widehat{\mathcal{D}}_K^{ch}\,;\, \widehat{\mathcal{E}}_K, \widehat{\mathcal{D}}_K, \widehat{\$}_{\mathcal{E}}, \widehat{\$}_{\mathcal{D}})\,,$$

where $O_{3,N} \not\succ O_{1,N}$, $\{O_{1,N}, O_{3,N}\} \not\succ O_{3,N}$, $O_3 \not\rightarrow O_2$, $O_3 \nuparrow O_4$. The qCPA$mL2 advantage of an adversary $\mathbf{A}$ on Π is defined as

$$\mathbf{Adv}^{\mathsf{qCPA\$mL2}}_{\mathcal{E}_K,\mathcal{D}_K,\mathcal{L}_E,\mathcal{L}_D}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\widehat{\mathcal{E}}_K, \widehat{\mathcal{E}}^{ch}_K, \widehat{\mathcal{D}}^{ch}_K ; \widehat{\mathcal{E}}_K, \widehat{\$}_{\mathcal{E}}, \widehat{\$}_{\mathcal{D}}) ,$$

where $O_{2,N} \not\succ O_{1,N}$, $\{O_{1,N}, O_{2,N}\} \not\succ O_{2,N}$, $O_2 \nuparrow O_3$.

We adapt two auxiliary notions, LUP-4 and XOR$, from [8,9] that reflect practical attacks under leakage regarding its non-invertability and the indistinguishability of XORs, respectively. Algorithms 4 and 5 defines their experiments.

Unpredictability. For TEDT, the LUP-2 game [9] modeled the unpredictability under leakage of a single iteration of the used PRG. LUP-4 generalizes the notion to the setting in RCTR. It takes a larger tweakey $K_0, T_0 \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ from the adversary, samples $K_1 \twoheadleftarrow \mathbb{F}_2^n$ and $T_1 \twoheadleftarrow \mathbb{F}_2^n$ and uses them for p decryptions, as well as p calls of one iteration of the PRG in TEDT2 that generates two n-bit outputs and K_2, T_2. $\mathbf{A}$ can query the encryption p times to collect a vector of input- and output leakages from all primitive calls except for the calls to $M_{0,1}$ and $M_{0,2}$, where it is not provided with input leakage. $\mathbf{A}$ outputs a set $\mathcal{K}'$ of q tuples (K_1, T_1) and wins iff the correct tweakey is contained.

Definition 4 (LUP-4). Let $\widetilde{\pi} \in \mathsf{TBC}(\mathbb{F}_2^n, \mathcal{T}_D, \mathbb{F}_2^n)$. Let $\mathcal{L}^{\mathsf{in}}$ and $\mathcal{L}^{\mathsf{out}}$ be sets of leakage functions. Let $\mathbf{A}$ be an adversary that provides $K_0, T_0 \in \mathbb{F}_2^n$ to and plays the LUP-4 experiment against $\widehat{\mathcal{E}}[\widetilde{\pi}]$, and outputs a set $\mathcal{K}' \subseteq (\mathbb{F}_2^n \times \mathbb{F}_2^n)^*$ with $|\mathcal{K}'| \leq q$. The LUP-4 advantage of $\mathbf{A}$ is defined as

$$\mathbf{Adv}^{\mathsf{LUP\text{-}4}}_{\widehat{\mathcal{E}}[\widetilde{\pi}]^{*,*,T_0}_{K_0},\mathcal{L}^{\mathsf{in}},\mathcal{L}^{\mathsf{out}}}(\mathbf{A}) \stackrel{\text{def}}{=} \Pr\left[(K_1, T_1) \in \mathcal{K}'\right] .$$

We define $\mathbf{Adv}^{\mathsf{LUP\text{-}4}}_{\widehat{\mathcal{E}}[\widetilde{\pi}],\mathcal{L}^{\mathsf{in}},\mathcal{L}^{\mathsf{out}}}(p, q)$ as the maximum of all LUP-4 adversaries $\mathbf{A}$ on $\widehat{\mathcal{E}}[\widetilde{\pi}]$ that ask at most p queries and output a set of at most q guesses.

Indistinguishability of XOR. An implementation that shall provide confidentiality must protect all operations. An XOR that leaks a single bit can destroy privacy, but the probability may be non-negligible [8,9,31]. However, it should be addressed in the security analysis. Their works proposed a notion of Left-or-Right XOR security that can be evaluated in practice on an isolated component. The XOR$ game in Algorithm 5 is our real-or-random variant thereof for consistency, where the real world processes a message $M \in \mathbb{F}_2^n$ chosen by the adversary, and the ideal world samples and processes a message $M^* \twoheadleftarrow \mathbb{F}_2^n$.

Definition 5 (XOR$). Let $\widetilde{\pi} \in \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$ and $K_1 \twoheadleftarrow \mathcal{K}$. Let $\mathcal{L}^{\mathsf{out}}$, $\mathcal{L}_1^{\oplus}$, and $\mathcal{L}^{\oplus}$ be sets of leakage functions. Let $\mathbf{A}$ be a adversary that plays the XOR$ experiment given in Algorithm 5 against $\widehat{\mathcal{E}}[\widetilde{\pi}]$. Then, the XOR$ advantage of $\mathbf{A}^b \Rightarrow b'$, interacting with world b and outputting b' is defined as $\mathbf{Adv}^{\mathsf{XOR\$}}_{\widehat{\mathcal{E}}[\widetilde{\pi}]_K,\mathcal{L}^{\mathsf{out}},\mathcal{L}^{\oplus}}(\mathbf{A}) \stackrel{\text{def}}{=} \left|\Pr\left[\mathbf{A}^1 \Rightarrow 1\right] - \Pr\left[\mathbf{A}^0 \Rightarrow 1\right]\right| .$

We define $\mathbf{Adv}^{\mathsf{XOR\$}}_{\widehat{\mathcal{E}}[\widetilde{\pi}]_K,\mathcal{L}^{\mathsf{out}},\mathcal{L}^{\oplus}}(p, q)$ for the maximum advantage over all XOR$ adversaries $\mathbf{A}$ on $\widehat{\mathcal{E}}[\widetilde{\pi}]_K$ that ask at most q queries under p measurements each.

Algorithm 4 LUP-4 experiment.

11: **procedure** INITIALIZE(K_0, T_0)
12: $K_1 \twoheadleftarrow \mathbb{F}_2^n$; $T_1 \twoheadleftarrow \mathbb{F}_2^n$
13: $M_{0,1} \leftarrow (\widetilde{E}_{K_0}^{0,0,T_0})^{-1}(K_1)$
14: $M_{0,2} \leftarrow (\widetilde{E}_{K_0}^{1,0,T_0})^{-1}(T_1)$

21: **function** LEAK$[\widetilde{E}](\Lambda^{\mathsf{in}}, \Lambda^{\mathsf{out}}, K, T, X, Y)$
22: $R^{\mathsf{in}}, R^{\mathsf{out}} \twoheadleftarrow \mathcal{R}$
23: $L^{\mathsf{in}} \leftarrow \Lambda^{\mathsf{in}}(K, T, X; R^{\mathsf{in}})$
24: $L^{\mathsf{out}} \leftarrow \Lambda^{\mathsf{out}}(K, T, Y; R^{\mathsf{out}})$
25: **return** $(L^{\mathsf{in}}, L^{\mathsf{out}})$

31: **function** FINALIZE($\mathcal{K}'$)
32: win $\leftarrow |\mathcal{K}'| \le q \wedge$
33: $(K_1, T_1) \in \mathcal{K}'$
34: **if** win **then**
35: **return** 1
36: **return** 0

41: **function** $\widehat{\mathcal{E}}[\widetilde{E}](N, \Lambda^{\mathsf{in}}, \Lambda^{\mathsf{out}})$
42: **for** $i \leftarrow 1..p$ **do**
43: $R_{0,1}^{\mathsf{out}}, R_{0,2}^{\mathsf{out}} \twoheadleftarrow \mathcal{R}$
44: $K_1 \leftarrow \widetilde{E}_{K_0}^{0,0,T_0}(M_{0,1})$; $T_1 \leftarrow \widetilde{E}_{K_0}^{1,0,T_0}(M_{0,2})$
45: $K_2 \leftarrow \widetilde{E}_{K_1}^{0,1,T_1}(N)$; $T_2 \leftarrow \widetilde{E}_{K_1}^{1,1,T_1}(N)$
46: $Z_{1,1} \leftarrow \widetilde{E}_{K_0}^{2,1,T_0}(N)$; $Z_{1,2} \leftarrow \widetilde{E}_{K_0}^{3,1,T_1}(N)$
47: $L_{0,1}^{\mathsf{out}} \leftarrow \Lambda_0^{\mathsf{out}}(K_0, (0,0,T_0), K_1; R_{0,1}^{\mathsf{out}})$
48: $L_{0,2}^{\mathsf{out}} \leftarrow \Lambda_0^{\mathsf{out}}(K_0, (1,0,T_0), T_1; R_{0,2}^{\mathsf{out}})$
49: $L_0 \leftarrow L_{0,1}^{\mathsf{out}}, L_{0,2}^{\mathsf{out}}$
50: $L_{1,1}^{\mathsf{in}}, L_{1,1}^{\mathsf{out}} \leftarrow \text{LEAK}[\widetilde{E}](\Lambda^{\mathsf{in}}, \Lambda^{\mathsf{out}}, K_1, (0,1,T_1), N, K_2)$
51: $L_{1,2}^{\mathsf{in}}, L_{1,2}^{\mathsf{out}} \leftarrow \text{LEAK}[\widetilde{E}](\Lambda^{\mathsf{in}}, \Lambda^{\mathsf{out}}, K_1, (1,1,T_1), N, T_2)$
52: $L_{1,3}^{\mathsf{in}}, L_{1,3}^{\mathsf{out}} \leftarrow \text{LEAK}[\widetilde{E}](\Lambda^{\mathsf{in}}, \Lambda^{\mathsf{out}}, K_1, (2,1,T_1), N, Z_{1,1})$
53: $L_{1,4}^{\mathsf{in}}, L_{1,4}^{\mathsf{out}} \leftarrow \text{LEAK}[\widetilde{E}](\Lambda^{\mathsf{in}}, \Lambda^{\mathsf{out}}, K_1, (3,1,T_1), N, Z_{1,2})$
54: $L_{1,1} \leftarrow L_{1,1}^{\mathsf{in}}, L_{1,1}^{\mathsf{out}}$; $L_{1,2} \leftarrow L_{1,2}^{\mathsf{in}}, L_{1,2}^{\mathsf{out}}$
55: $L_{1,3} \leftarrow L_{1,3}^{\mathsf{in}}, L_{1,3}^{\mathsf{out}}$; $L_{1,4} \leftarrow L_{1,4}^{\mathsf{in}}, L_{1,4}^{\mathsf{out}}$
56: **return** $(K_2, T_2, Z_{1,1}, Z_{1,2},$
57: $[L_0]_p, [L_{1,1}]_p, [L_{1,2}]_p, [L_{1,3}]_p, [L_{1,4}]_p)$

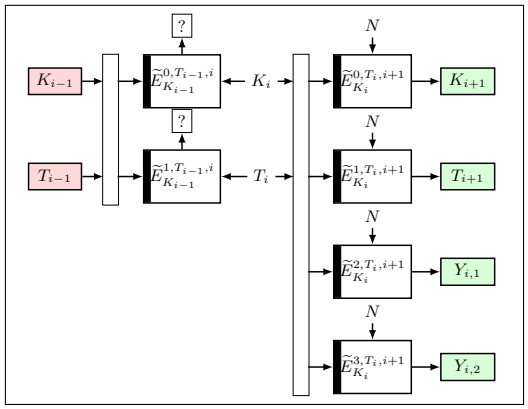


Fig. 4: The LUP-4 setting.

6 Authentication Security Analysis of TEDT2

TEDT2 inherits the single-user CIML2 security from [9]. Since it is similar to the proof of TEDT, we provide a cleaned and slightly adapted description in Appendix C As for TEDT, we assume that all intermediate values may leak completely, except for the key K of the key-derivation and tag-generation functions. The leakage-function sets are defined as singletons $\mathcal{L}_E = \{\Lambda_E\}$ and $\mathcal{L}_D = \{\Lambda_D\}$, where on input (N, A, M), Λ_E and Λ_D return

- (K, S, X, Y) for each primitive call of the PRG G, where $S = (D, T, U) \in \mathcal{T}_D$,
- (X, S, Y) for each call to the key-derivation function,
- (N, U, V, T) for each call to the tag-generation function, and
- (A, B) for each XOR of $A \oplus B$.

Algorithm 5 XOR$ experiment.

```
11: function Initialize(K, T, M, Λ^out, Λ^⊕)
12:     Y ←$ F_2^n; b ←$ {0,1}
13:     M* ← M
14:     if b = 0 then
15:         M* ←$ F_2^n
16:     X ← (Ẽ_K^T)^{-1}(Y)
21: function Finalize(b')
22:     return b = b'
31: function Ê[Ẽ]_K(Λ^out, Λ^⊕)
32:     R^out, R^⊕ ←$ R
33:     Y ← Ẽ_K^T(X);  C ← Y ⊕ M*
34:     L^out ← Λ^out(K, T, Y; R^out)
35:     L^⊕ ← Λ^⊕(Y, C; R^⊕)
36:     for i ← 2..p do
37:         R^out, R^⊕ ←$ R
38:         Y ← Ẽ_K^T(X);  M* ← C ⊕ Y
39:         L^out ← Λ^out(K, T, Y; R^out)
40:         L^⊕ ← Λ^⊕(Y, M*; R^⊕)
41:     return (C, [L^out]_p, [L^⊕]_p)
```

We follow the steps by Berti et al. [9]:

(1) We replace the KDF and TGF by ideal secret tweakable keyed permutations, independent of each other and all other permutation calls.
(2) Then, we study the calls to the TGF and upper bound the probability of partial collisions and partial multi-collisions in V as bad events.
(3) Third, we upper bound the probability of forgeries for good transcripts.

The result is given in Theorem 1. The proof can be found in Appendix C.

Theorem 1 (qCIML2 Security). Let $\widetilde{\pi} \in \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$, $K \twoheadleftarrow \mathcal{K}$, and $n \geq 4$. Let $\mathbf{A}$ be a qCIML2-adversary on $\Pi[\widetilde{\pi}]_K = \mathsf{TEDT2}[\widetilde{\pi}]_K$ that makes at most q_c construction queries of at most σ message blocks and σ_a associated-data blocks in total and q_p primitive queries. Let q_e be the numbers of encryption construction queries, q_d the cumulative decryption construction queries to the oracles and μ be the number of encryption queries with repeating nonces in non-challenge encryption queries. Let σ_p be the number of ideal-primitive calls in all primitive and construction queries of $\mathbf{A}$, where $\sigma \leq 3\sigma + \sigma_a + 2q_c + q_p \leq 2^{n-3}$. Let the sets of leakage functions $\mathcal{L}_E$ and $\mathcal{L}_D$ be as defined in Section 6. Then

$$\mathbf{Adv}^{\mathsf{qCIML2}}_{\Pi[\widetilde{\pi}]_K, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D}(q_e, q_d, \sigma, q_p) \leq \mathbf{Adv}^{\mathsf{TPRP}}_{\widetilde{\pi}}(2q_c) + \mathbf{Adv}^{\mathsf{STPRP}}_{\widetilde{\pi}}(q_e, q_d) + \frac{2n(q_p + q_d) + 2q_d + 3(\sigma + \sigma_a) + 1}{2^n} + \frac{9\sigma_p^2 + 2\sigma_p + 9q_p(q_d + \mu) + q_c^2}{2^{2n}}.$$

7 Encryption Security Analysis of TEDT2

We adopt the definitions of $\widetilde{\pi}$ and K from Section 6. The leakage-function sets are singletons $\mathcal{L}_E = \{\Lambda_E\}$ and $\mathcal{L}_D = \{\Lambda_D\}$, where on input (N, A, M), Λ_E outputs

– $\Lambda^{\mathsf{in}}(K, T, X)$ and $\Lambda^{\mathsf{out}}(K, T, Y)$ for each call of $\widetilde{\pi}_K^T(X) = Y$ in Encrypt.
– $\Lambda^{\oplus}(A, B)$ for each XOR of two values $A \oplus B$ by internal actions
– All intermediate values during hashing, i.e., the hash function is unprotected.

On input (N, A, C, T), Λ_D returns the values corresponding to the above that occur during regular decryption in Algorithm 1. We let $\mathcal{L}^{\mathsf{in}} = \{\Lambda^{\mathsf{in}}\}$, $\mathcal{L}^{\mathsf{out}} = \{\Lambda^{\mathsf{out}}\}$, and $\mathcal{L}^{\oplus} = \{\Lambda^{\mathsf{out}}\}$ be the leakage-function sets that contain the parts of Λ_E and Λ_D, respectively, that are used in the LUP-4 and XOR$ notions.

Theorem 2. Let $\mathbf{A}$ be an qCCA$mL2 adversary on $\Pi[\widetilde{\pi}]_K = \mathsf{TEDT2}[\widetilde{\pi}]_K$ that asks at most q_e encryption queries and q_d decryption queries of at most σ blocks in total and q_p primitive queries. Let $F[\widetilde{\pi}]$ be an iteration of $\mathsf{RCTR}[\widetilde{\pi}]$. Let $n \geq 4$, $\sigma \leq 2^{n-3}$, and let $\mathcal{L}_E$ and $\mathcal{L}_D$ be as defined at the top of Section 7. Then

$$\begin{aligned}\mathbf{Adv}^{\mathsf{qCCA\$mL2}}_{\Pi[\widetilde{\pi}]_K,\widetilde{\pi}^{\pm},\mathcal{L}_E,\mathcal{L}_D}(q_e,q_d,\sigma,q_p) &\leq \frac{4q_p(\sigma+q_c)+4(\sigma+q_c)^2}{2^{2n}} \\ &+ \mathbf{Adv}^{\mathsf{qCIML2}}_{\Pi[\widetilde{\pi}]_K,\widetilde{\pi}^{\pm},\mathcal{L}_E,\mathcal{L}_D}(q_e,q_d,\sigma,q_p) + 2\sigma \cdot \mathbf{Adv}^{\mathsf{LUP\text{-}4}}_{F[\widetilde{\pi}],\mathcal{L}^{\mathsf{in}},\mathcal{L}^{\mathsf{out}}}(p,\sigma+q_c+q_p) \\ &+ \sigma \cdot \mathbf{Adv}^{\mathsf{XOR\$}}_{F[\widetilde{\pi}],\mathcal{L}^{\mathsf{out}},\mathcal{L}^{\oplus}}(p,2\sigma+2q_c+q_p)\,.\end{aligned}$$

The latter term reflects side-channel tweakey recovery and has a birthday-bound complexity of

$$O\left(\sigma \cdot \frac{\sigma+q_c+q_p}{c \cdot 2^{2n}}\right).$$

Following the argument by Berti et al. [9], the three leakage traces per tweakey should not render the value of c significant. While TEDT2 doubles the number of available traces for every tweakey to six, two n-bit values must be recovered. Thus, we assume that the resulting term of $c \cdot 2^{2n}$ that reflects six traces remains insignificant. The term $\sigma \cdot \mathbf{Adv}^{\mathsf{XOR\$}}_{F[\widetilde{\pi}]}(p, 2q_c+2\sigma+q_p)$ represents the distinguishing advantage in the case of "minimal message manipulation" and is inevitable for schemes that employ XOR during encryption.

Proof. The proof can follow the steps of the muCCAmL2 proof for TEDT in [9]. All queries of $\mathbf{A}$ will be stored in a transcript $\tau = \tau_c \cup \tau_p$. In this context, τ_c consists of the construction queries of $\mathbf{A}$ and their corresponding responses. τ_p represents the primitive queries of $\mathbf{A}$ to $\widetilde{\pi}^{\pm}$ and their associated responses. We have to show that

(1) Encryption queries provide confidentiality.
(2) Leakages from the decryption oracle do not affect confidentiality.

From Theorem 1, we can upper bound the advantage that TEDT2 is qCIML2-secure. Thus, unless $\mathbf{A}$ can forge, non-trivial decryption queries will yield only the $\perp$ symbol and leak only the invalid decryptions from $(\widetilde{\pi}_K^{8,N,H(A,C)})^{-1}(T)$. We can capture this as follows.

$$\underset{\mathbf{A}}{\Delta}(\widehat{\mathcal{E}}_K, \widehat{\mathcal{D}}_K, \widehat{\mathcal{E}}^{ch}_K, \widehat{\mathcal{D}}^{ch}_K, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D\,;\, \widehat{\mathcal{E}}_K, \widehat{\mathcal{D}}_K, \widehat{\$}_{\mathcal{E}}, \widehat{\$}_{\mathcal{D}}, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D) \qquad (1)$$

with the restrictions $O_{3,N} \not\succ O_{1,N}$, $\{O_{1,N}, O_{3,N}\} \not\succ O_{3,N}$, $O_3 \not\rightsquigarrow O_2$, and $O_3 \not\uparrow O_4$. We can introduce an intermediate step of

$$\underset{\mathbf{A}}{\Delta}(\widehat{\mathcal{E}}_K, \widehat{\mathcal{D}}_K, \widehat{\mathcal{E}}^{ch}_K, \widehat{\mathcal{D}}^{ch}_K, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D\,;\, \widehat{\mathcal{E}}_K, \widehat{\perp}, \widehat{\mathcal{E}}^{ch}_K, \widehat{\mathcal{D}}^{ch}_K, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D)\,,$$

with the restrictions as for Equation (1) plus $O_1 \not\rightsquigarrow O_2$. Its left-hand side is equivalent to that in Equation (1) since every adversary that wanted to ask a query that was output from $O_1 \hookrightarrow O_2$ before could now use $O_3 \hookrightarrow O_4$ for this purpose. This setting can be reformulated as

$$\begin{aligned}&\underset{\mathbf{A}}{\Delta}(\widehat{\mathcal{E}}_K, \widehat{\mathcal{D}}_K, \widehat{\mathcal{D}}_K, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D \,;\, \widehat{\mathcal{E}}_K, \widehat{\mathcal{D}}_K, \widehat{\perp}, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D)\\ &\leq \mathbf{Adv}^{\mathsf{qCIML2}}_{\mathcal{E}[\widetilde{\pi}], \mathcal{D}[\widetilde{\pi}], \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D}(q_e, q_d, \sigma, q_p) \qquad (2)\end{aligned}$$

since the nonce restrictions to two encryption oracles are irrelevant. The remaining restrictions merge to $O_1 \not\rightsquigarrow O_3$ for Equation (2), which yields the qCIML2 notion. The remaining step is to upper bound the distinguishing advantage of

$$\begin{aligned}&\underset{\mathbf{A}}{\Delta}(\widehat{\mathcal{E}}_K, \widehat{\perp}, \widehat{\mathcal{E}}^{ch}_K, \widehat{\mathcal{D}}^{ch}_K, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D \,;\, \widehat{\mathcal{E}}_K, \widehat{\perp}, \widehat{\$}_{\mathcal{E}}, \widehat{\$}_{\mathcal{D}}, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D)\\ &= \underset{\mathbf{A}}{\Delta}(\widehat{\mathcal{E}}_K, \widehat{\mathcal{E}}^{ch}_K, \widehat{\mathcal{D}}^{ch}_K, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D \,;\, \widehat{\mathcal{E}}_K, \widehat{\$}_{\mathcal{E}}, \widehat{\$}_{\mathcal{D}}, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D)\\ &\leq \mathbf{Adv}^{\mathsf{qCPA\$mL2}}_{\mathcal{E}[\widetilde{\pi}], \mathcal{D}[\widetilde{\pi}], \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D}(q_e, q_d, \sigma, q_p) \qquad (3)\end{aligned}$$

with the restrictions $O_{2,N} \not\succ O_{1,N}$, $\{O_{1,N}, O_{2,N}\} \not\succ O_{2,N}$, and $O_2 \not\uparrow O_3$. The advantage in Equation (3) for TEDT2 can be upper bounded by Lemma 1. □

Lemma 1 (qCPA$mL2 Security of TEDT2 with RCTR). Let $\widetilde{\pi} \twoheadleftarrow \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$. Let $\mathbf{A}$ be an qCPA$mL2 adversary on $\mathit{\Pi}[\widetilde{\pi}]_K = \mathsf{TEDT2}[\widetilde{\pi}]_K$ that asks at most q_e encryption queries and q_d decryption queries of at most σ blocks in total and q_p primitive queries. Let $F[\widetilde{\pi}]$ be an iteration of $\mathsf{RCTR}[\widetilde{\pi}]$. Let $n \geq 2$, $\sigma \leq 2^{n-3}$, and let $\mathcal{L}_E$ and $\mathcal{L}_D$ be as defined at the top of Section 7. Then

$$\begin{aligned}\mathbf{Adv}^{\mathsf{qCPA\$mL2}}_{\mathit{\Pi}[\widetilde{\pi}]_K, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D}(q_e, q_d, \sigma, q_p) &\leq \frac{4q_p(\sigma + q_c) + 4(\sigma + q_c)^2}{2^{2n}}\\ &+ 2\sigma \mathbf{Adv}^{\mathsf{LUP\text{-}4}}_{F[\widetilde{\pi}], \mathcal{L}^{\mathsf{in}}, \mathcal{L}^{\mathsf{out}}}(p, \sigma + q_c + q_p) + \sigma \mathbf{Adv}^{\mathsf{XOR\$}}_{F[\widetilde{\pi}], \mathcal{L}^{\mathsf{out}}, \mathcal{L}^{\oplus}}(p, 2\sigma + 2q_c + q_p)\,.\end{aligned}$$

The proof is given in Appendix D.

Comparison of Security Bounds. Figure 5 compares the single-user CCAmL2 and CIML2 bounds for TEDT with the corresponding qCCA$mL2 and qCIML2 bounds for TEDT2. The plot uses $n = 128$ and $q_c = q_p = \sigma$ as an example. It contains the bounds for the LUP-2 and LUP-4 terms, respectively, but must omit the terms for XOR$ security since it strongly depends on the implementation and must be determined in practice. The lines that represent the qCCA$mL2 and qCIML2 security bounds for TEDT2 almost overlap, which is natural since the qCCA$mL2 bound contains the dominating bound for qCIML2 security. The difference in the terms of qCIML2 is largely due to the presentation. The main effect is the difference between the birthday-bound term for LUP-2 in the CCAmL2 bound of TEDT, while the $2n$-bit tweakey of TEDT2 leads to an improved bound.

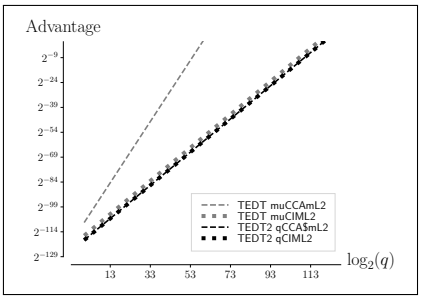


Fig. 5: Single-user CIML2 and CCAmL2 bounds of TEDT and qCIML2 and qCCA$mL2 bounds for TEDT2, for $n = 128$ and where $q_c = q_p = \sigma = q$ are used.

8 Discussion and Future Work

Under the umbrella aim of maintaining qCIML2 and qCCA$mL2 security, our core goals for TEDT2 were three-fold: to adopt a more efficient hash function, to render the authentication more efficient by moving the nonce to the tag-generation function, and to strengthen the encryption under leakage beyond the birthday bound. For this purpose, we adopted the more efficient hash function from Naito's result on MDPH and used a TBC with $3n$-bit tweakey to have a rate of about one while hashing. We could spare the effort for hashing the nonce and obtain higher security for the encryption mode and thus for the scheme under leakage, using a TWEAKEY-based TBC with $3n$-bit tweakey. We emphasize that the use of MDPH and a TBC with $3n$-bit tweakey has been also proposed for AET-LR and Romulus-LR-TEDT. We do not claim this adoption as a novel idea. Note that AET-LR used a more efficient sequential hashing of the associated data that used both state input and tweak, which is efficient and secure in the black-box setting but needs costly protection against DPAs [28] under leakage.

As an interesting by-product, we obtain an even more secure authentication in the black-box setting. In Appendix E, we also prove nMAC security of the MAC NHaT. We identify several potential future improvements: the efficiency of the hash function can be further increased with longer tweaks, as suggested by Peyrin [49]. Concerning the PRG, the rate of $1/2$ may limit the applicability of our proposal in high-performance settings. While the same tweakey could be used for more blocks, this would need a fine-grained study of the leakage. Alternatively, other PRGs such as a generalization of TET from [9] might appear useful. Thanks to having modular proofs, only the confidentiality proof would have to be revised in this case. Finally, while our focus was on security in the usual single-user setting, an interesting future work could consider multiple users.

Acknowledgments We thank all reviewers of Latincrypt 2021 for their highly fruitful comments and notes. We are highly thankful to Christoph Dobraunig who pointed out the potential of leveled permutation-based schemes and Avijit

Dutta for pointing out a slip in the computation of the good transcripts in the qCIML2 proof. The author was supported by DFG grant LU 608/9-1.

References

1. Megha Agrawal, Donghoon Chang, and Somitra Sanadhya. sp-AELM: Sponge based authenticated encryption scheme for memory constrained devices. In *ACISP*, pages 451–468. Springer, 2015.
2. Tomer Ashur, Orr Dunkelman, and Atul Luykx. Boosting Authenticated Encryption Robustness with Minimal Modifications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO III*, volume 10403 of *LNCS*, pages 3–33. Springer, 2017.
3. Manuel Barbosa and Pooya Farshim. Indifferentiable Authenticated Encryption. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO I*, volume 10991 of *LNCS*, pages 187–220. Springer, 2018.
4. Guy Barwell, Daniel P. Martin, Elisabeth Oswald, and Martijn Stam. Authenticated Encryption in the Face of Protocol and Side Channel Leakage. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT I*, volume 10624 of *LNCS*, pages 693–723. Springer, 2017.
5. Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In Tatsuaki Okamoto, editor, *ASIACRYPT*, volume 1976 of *LNCS*, pages 531–545. Springer, 2000.
6. Mihir Bellare and Bennet S. Yee. Forward-Security in Private-Key Cryptography. In Marc Joye, editor, *CT-RSA*, volume 2612 of *LNCS*, pages 1–18. Springer, 2003.
7. Davide Bellizia, Francesco Berti, Olivier Bronchain, Gaëtan Cassiers, Sébastien Duval, Chun Guo, Gregor Leander, Gaëtan Leurent, Itamar Levi, Charles Momin, Olivier Pereira, Thomas Peters, François-Xavier Standaert, Balazs Udvarhelyi, and Friedrich Wiemer. Spook: Sponge-Based Leakage-Resistant Authenticated Encryption with a Masked Tweakable Block Cipher. *IACR ToSC*, 2020(S1):295–349, 2020.
8. Davide Bellizia, Olivier Bronchain, Gaëtan Cassiers, Vincent Grosso, Chun Guo, Charles Momin, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Mode-Level vs. Implementation-Level Physical Security in Symmetric Cryptography - A Practical Guide Through the Leakage-Resistance Jungle. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO I*, volume 12170 of *LNCS*, pages 369–400. Springer, 2020.
9. Francesco Berti, Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. TEDT, a Leakage-Resist AEAD Mode for High Physical Security Applications. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(1):256–320, 2020.
10. Francesco Berti, François Koeune, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Leakage-Resilient and Misuse-Resistant Authenticated Encryption. *IACR Cryptol. ePrint Arch.*, 2016:996, 2016.
11. Francesco Berti, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. On Leakage-Resilient Authenticated Encryption with Decryption Leakages. *IACR ToSC*, 2017(3):271–293, 2017.
12. Gaëtan Cassiers, Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. SpookChain: Chaining a Sponge-Based AEAD with Beyond-Birthday Security. In Shivam Bhasin, Avi Mendelson, and Mridul Nandi, editors, *SPACE*, volume 11947 of *LNCS*, pages 67–85. Springer, 2019.

13. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *LNCS*, pages 398–412. Springer, 1999.
14. Shan Chen and John P. Steinberger. Tight Security Bounds for Key-Alternating Ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT*, volume 8441 of *LNCS*, pages 327–350. Springer, 2014. Full version at https://eprint.iacr.org/2013/222.
15. Benoît Cogliati, Jooyoung Lee, and Yannick Seurin. New Constructions of MACs from (Tweakable) Block Ciphers. *IACR Trans. Symmetric Cryptol.*, 2017(2):27–58, 2017.
16. Ivan Damgård. Collision Free Hash Functions and Public Key Signature Schemes. In David Chaum and Wyn L. Price, editors, *EUROCRYPT*, volume 304 of *LNCS*, pages 203–216. Springer, 1987.
17. Donald W. Davies and Wyn L. Price. Digital signatures, an update. *Proc. 5th Int. Conf. on Computer Communication*, pages 845–849, October 1984.
18. Jean Paul Degabriele, Christian Janson, and Patrick Struck. Sponges Resist Leakage: The Case of Authenticated Encryption. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT II*, volume 11922 of *LNCS*, pages 209–240. Springer, 2019.
19. Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, Robert Primas, and Thomas Unterluggauer. Isap v2.0. *IACR ToSC*, 2020(S1):390–416, 2020.
20. Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, and Thomas Unterluggauer. ISAP - Towards Side-Channel Secure Authenticated Encryption. *IACR ToSC*, 2017(1):80–105, 2017.
21. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2. Sep 15 2016. Selection of the CAESAR competition.
22. Christoph Dobraunig and Bart Mennink. Leakage Resilience of the Duplex Construction. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT III*, volume 11923 of *LNCS*, pages 225–255. Springer, 2019.
23. Christoph Dobraunig and Bart Mennink. Leakage Resilience of the ISAP Mode: a Vulgarized Summary. In *NIST LWC Workshop*, volume 2019, page 23, 2019.
24. Christoph Dobraunig and Bart Mennink. Leakage Resilient Value Comparison with Application to Message Authentication. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT II*, volume 12697 of *LNCS*, pages 377–407. Springer, 2021.
25. Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In Michael Mitzenmacher, editor, *STOC*, pages 621–630. ACM, 2009.
26. Stefan Dziembowski and Krzysztof Pietrzak. Leakage-Resilient Cryptography. In *FOCS*, pages 293–302. IEEE Computer Society, 2008.
27. Louis Goubin and Jacques Patarin. DES and Differential Power Analysis (The "Duplication" Method). In Çetin Kaya Koç and Christof Paar, editors, *CHES*, volume 1717 of *LNCS*, pages 158–172. Springer, 1999.
28. Chun Guo, Mustafa Khairallah, and Thomas Peyrin. AET-LR: Rate-1 Leakage-Resilient AEAD based on the Romulus Family. In *NIST LWC Workshop*, 2020.
29. Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Authenticated Encryption with Nonce Misuse and Physical Leakages: Definitions, Separation Results, and Leveled Constructions. *IACR Cryptol. ePrint Arch.*, 2018:484, 2018. version 20190711:105233.

30. Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Authenticated Encryption with Nonce Misuse and Physical Leakage: Definitions, Separation Results and First Construction - (Extended Abstract). In Peter Schwabe and Nicolas Thériault, editors, *LATINCRYPT*, volume 11774 of *LNCS*, pages 150–172. Springer, 2019.
31. Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Towards Low-Energy Leakage-Resistant Authenticated Encryption from the Duplex Sponge Construction. *IACR ToSC*, 2020(1):6–42, 2020.
32. Shoichi Hirose. Some Plausible Constructions of Double-Block-Length Hash Functions. In Matthew J. B. Robshaw, editor, *FSE*, volume 4047 of *LNCS*, pages 210–225. Springer, 2006.
33. Shoichi Hirose, Je Hong Park, and Aaram Yun. A Simple Variant of the Merkle-Damgård Scheme with a Permutation. In Kaoru Kurosawa, editor, *ASIACRYPT*, volume 4833 of *LNCS*, pages 113–129. Springer, 2007.
34. Viet Tung Hoang and Stefano Tessaro. Key-Alternating Ciphers and Key-Length Extension: Exact Bounds and Multi-user Security. In *CRYPTO*, pages 3–32. Springer, 2016.
35. Tetsu Iwata, Mustafa Khairallah, Kazuhiko Minematsu, and Thomas Peyrin. New Results on Romulus. *NIST LWC Workshop*, 2020.
36. Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT II*, volume 8874 of *LNCS*, pages 274–288. Springer, 2014.
37. Yael Tauman Kalai and Leonid Reyzin. A survey of leakage-resilient cryptography. In Oded Goldreich, editor, *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 727–794. ACM, 2019.
38. Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Neal Koblitz, editor, *CRYPTO*, volume 1109 of *LNCS*, pages 104–113. Springer, 1996.
39. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *LNCS*, pages 388–397. Springer, 1999.
40. Jake Longo, Daniel P. Martin, Elisabeth Oswald, Daniel Page, Martijn Stam, and Michael Tunstall. Simulatable Leakage: Analysis, Pitfalls, and New Constructions. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT I*, volume 8873 of *LNCS*, pages 223–242. Springer, 2014.
41. Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In Moni Naor, editor, *TCC*, volume 2951 of *LNCS*, pages 21–39. Springer, 2004.
42. Ralph C. Merkle. One Way Hash Functions and DES. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *LNCS*, pages 428–446. Springer, 1989.
43. Silvio Micali and Leonid Reyzin. Physically Observable Cryptography (Extended Abstract). In Moni Naor, editor, *TCC*, volume 2951 of *LNCS*, pages 278–296. Springer, 2004.
44. Yusuke Naito. Optimally Indifferentiable Double-Block-Length Hashing Without Post-processing and with Support for Longer Key Than Single Block. In Peter Schwabe and Nicolas Thériault, editors, *LATINCRYPT*, volume 11774 of *LNCS*, pages 65–85. Springer, 2019.
45. Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Lightweight Authenticated Encryption Mode Suitable for Threshold Implementation. In Anne Canteaut and

Yuval Ishai, editors, *EUROCRYPT II*, volume 12106 of *LNCS*, pages 705–735. Springer, 2020.
46. Chanathip Namprempre, Phillip Rogaway, and Thomas Shrimpton. Reconsidering Generic Composition. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT*, volume 8441 of *LNCS*, pages 257–274. Springer, 2014.
47. Jacques Patarin. The "Coefficients H" Technique. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *SAC*, volume 5381 of *LNCS*, pages 328–345. Springer, 2008.
48. Olivier Pereira, François-Xavier Standaert, and Srinivas Vivek. Leakage-Resilient Authentication and Encryption from Symmetric Cryptographic Primitives. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS*, pages 96–108. ACM, 2015.
49. Thomas Peyrin. Tweakable Block Cipher-Based Cryptography, Nov 12 2020.
50. Bart Preneel, René Govaerts, and Joos Vandewalle. Hash Functions Based on Block Ciphers: A Synthetic Approach. In Douglas R. Stinson, editor, *CRYPTO*, volume 773 of *LNCS*, pages 368–378. Springer, 1993.
51. Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with Composition: Limitations of the Indifferentiability Framework. In Kenneth G. Paterson, editor, *EUROCRYPT*, volume 6632 of *LNCS*, pages 487–506. Springer, 2011.
52. Phillip Rogaway. Authenticated-encryption with associated-data. In Vijayalakshmi Atluri, editor, *ACM CCS*, pages 98–107. ACM, 2002.
53. Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *LNCS*, pages 373–390. Springer, 2006.
54. François-Xavier Standaert. Towards an Open Approach to Side-Channel Resistant Authenticated Encryption. In Chip-Hong Chang, Ulrich Rührmair, Daniel E. Holcomb, and Patrick Schaumont, editors, *ACM*, page 1. ACM, 2019.
55. François-Xavier Standaert, Olivier Pereira, and Yu Yu. Leakage-Resilient Symmetric Cryptography under Empirically Verifiable Assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO I*, volume 8042 of *LNCS*, pages 335–352. Springer, 2013.
56. Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. Shuffling against Side-Channel Attacks: A Comprehensive Study with Cautionary Note. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *LNCS*, pages 740–757. Springer, 2012.
57. Yu Yu, François-Xavier Standaert, Olivier Pereira, and Moti Yung. Practical leakage-resilient pseudorandom generators. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM CCS*, pages 141–151. ACM, 2010.

A Changelog

Revision of November 29, 2022

- We revised the definition of the KDF so that $\widetilde{E}$ takes the nonce as a tweak instead of as a state input to strengthen its collision resistance.
- We corrected the proof of the interpolation ratio of good transcripts to occur in the real vs. in the ideal world in the proof of Lemma 6.
- We clarified in Section 1 that leveled purely permutation-based implementations are equally possible and well-suited as leveled TBC-based implementations.

B Notions

B.1 The H-coefficient Technique

The H-coefficient technique is a proof method by Patarin [47] that was modernized by Chen and Steinberger [14] and generalized by Hoang and Tessaro [34] in their expectation method, which allowed to derive the fundamental lemma as a corollary. A distinguisher $\mathbf{A}$ obtains outputs from a real world $\mathcal{O}^1$ or an ideal world $\mathcal{O}^0$, where the results of its interaction are collected in a transcript τ. The oracles can sample random coins before the experiment (often a key or an ideal primitive that is sampled beforehand) and are then deterministic [14]. A transcript τ is called attainable if $\Pr[\Theta_{\text{ideal}} = \tau] > 0$. Let Θ denote the set of all attainable transcripts. The fundamental Lemma of the H-coefficients technique, whose proof can be found e.g., in [14,47], states that we can split the set Θ into two disjoint sets GoodT and BadT and upper bound the distinguishing advantage as in Lemma 2. A proof can be found, e.g., in [14].

Lemma 2 ([14,47]). Assume, there exist $\epsilon_1, \epsilon_2 \geq 0$ s. t. for any transcript $\tau \in \text{GoodT}$, it holds $\Pr[\Theta_{\text{real}} = \tau] \,/\, \Pr[\Theta_{\text{ideal}} = \tau] \geq 1-\epsilon_1$ and $\Pr[\Theta_{\text{ideal}} \in \text{BadT}] \leq \epsilon_2$. Then, for all distinguishers $\mathbf{A}$, it holds that $\Delta_{\mathbf{A}}(\mathcal{O}^1 ; \mathcal{O}^0) \leq \epsilon_1 + \epsilon_2$.

B.2 Compatible Permutations

Let $\widetilde{\pi} \in \widetilde{\mathsf{Perm}}(\mathcal{T}, \mathcal{X})$ be a permutation for finite non-empty sets $\mathcal{T}$ and $\mathcal{X}$. Let a transcript $\tau = \{(T^i, X^i, Y^i)\}$, for $i \in [q]$, be a set of queries and responses. A permutation $\widetilde{\pi}$ is compatible to a transcript τ, if for all elements in τ, it holds that $Y^i = \widetilde{\pi}^{T^i}(X^i)$ and $X^i = (\widetilde{\pi}^{T^i})^{-1}(Y^i)$. We denote by

$$\mathsf{Comp}(\tau) \stackrel{\text{def}}{=} \left\{ \widetilde{\pi} \in \widetilde{\mathsf{Perm}}(\mathcal{T}, \mathcal{X}) : Y^i = \widetilde{\pi}^{T^i}(X^i), \forall i \in [q] \right\}$$

the set of all permutations $\widetilde{\pi}$ that are compatible to the transcript τ. Thus, if $\widetilde{\pi}$ is compatible to τ, it holds that $\widetilde{\pi} \in \mathsf{Comp}(\tau)$.

C qCIML2 Proof of TEDT2

This section summarizes the qCIML2 analysis of TEDT2. It is naturally very similar to that of TEDT of Berti et al. [9] and differs only in few aspects that will be highlighted. Throughout the section, we will use $\mathcal{K} = \mathbb{F}_2^n$, $\mathcal{T}_D = \mathcal{D} \times \mathcal{T}_1 \times \mathcal{T}_2 = \mathbb{F}_2^{2n}$, where $\mathcal{D} = \mathbb{F}_2^d$, $\mathcal{T}_1 = \mathbb{F}_2^{n-d}$, and $\mathcal{T}_2 = \mathbb{F}_2^n$.

C.1 Idealizing the KDF and TGF

The analysis will employ a sequence of lemmas. First, we replace the two calls in the key-derivation function $\text{KDF}[\widetilde{\pi}]_K$ by ideal tweakable block ciphers $\widetilde{\pi}_1, \widetilde{\pi}_2 \twoheadleftarrow \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$ and the tag-generation function by $\widetilde{\pi}_3 \twoheadleftarrow \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$ that cannot be queried separately. The difference between both settings is upper bounded by

$$\mathbf{Adv}^{\mathsf{TPRP}}_{\widetilde{\pi}}(2q_c) + \mathbf{Adv}^{\mathsf{STPRP}}_{\widetilde{\pi}}(q_e, q_d)\,. \tag{4}$$

C.2 Multi-semi-collision Resistance

While we have a nonce in the tag-generation function, in order to upper bound the security under nonce-misusing adversaries, we still need to consider the probability of semi-collisions in the hash. We have to consider an upper bound on r-multi-collisions of outputs from the Davies-Meyer compression function[1] to bound the probability to obtain multi-collision in values V. For this purpose, recall the definition of the Davies-Meyer compression function $\mathsf{DM}[\widetilde{\pi}] : \mathcal{K} \times \mathcal{T}_D, \times \mathbb{F}_2^n \to \mathbb{F}_2^n$:

$$\mathsf{DM}[\widetilde{\pi}](K, T, X) \stackrel{\text{def}}{=} \widetilde{\pi}_K^T(X) \oplus X \,.$$

For $\widetilde{\pi} \in \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$, let the single-input $\mathsf{DM}[\widetilde{\pi}](X)$ for $X = (K,\ T,\ M_{1,1},\ M_{1,2})$ be a short-hand version for $\mathsf{DM}[\widetilde{\pi}](K, (D, T, M_{1,1}), M_{1,2})$, where $D \in \mathcal{D}$ is a domain constant. Lemma 3 is given in [9]. The proof can be found ibidem.

Lemma 3 (r-collision Multi-collisions of DM [9]). For any adversary $\mathbf{A}$ that makes at most $q \leq 2^n/2$ queries to $\widetilde{\pi} \twoheadleftarrow \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$ and positive integer r, the probability to find r pairwise distinct $X_1, \ldots, X_r \in \mathbb{F}_2^n$ such that

$$\Pr\left[\mathsf{DM}[\widetilde{\pi}](X_1) = \cdots = \mathsf{DM}[\widetilde{\pi}](X_r)\right] \leq \frac{(2q)^r}{r! \cdot 2^{(r-1)n}} \,.$$

C.3 Collision-freeness of the Hash Function

The next step is to upper bound the probability of hash-function collisions and multi-collisions in one part of the hash output. We consider $\textsc{Hash}[\widetilde{\pi}]$ as in algorithm 1, where $\widetilde{\pi} \twoheadleftarrow \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$. Naturally, the structure of the analysis will be similar to that by Berti et al. [9], but the resulting bound will differ from that of TEDT.

We consider a transcript $\tau = \tau_h \cup \tau_p$, where $\tau_h = \{(U^i, V^i, M^i)\}$ consists of exactly the queries to the construction and $\tau_p = \{(K^i, T^i, X^i, Y^i)\}$ to queries to the primitive or to its inverse and the corresponding result $Y^i = \widetilde{\pi}_{K^i}^{T^i}(X^i)$. We adopt the strategy of matching-query adversaries from [9,32]. Every time $\mathbf{A}$ makes a primitive forward query to $Y \leftarrow \widetilde{\pi}_K^T(X)$, the challenger makes an additional matching query $Y' \leftarrow \widetilde{\pi}_K^T(X')$ with $X' = X \oplus 1$ (which reflects the primitive call in the lower lane of $\textsc{Hash}$) and stores (K, T, X, Y) and (K, T, X', Y') into the adversary's query transcript τ_p. Similarly, every time $\mathbf{A}$ makes a primitive backward query to $X \leftarrow (\widetilde{\pi}_K^T)^{-1}(Y)$, the challenger makes a matching query $Y' \leftarrow \widetilde{\pi}_K^T(X')$ with $X' = X \oplus 1$, and inserts (K, T, X, Y) and (K, T, X', Y') into τ. Thus, every time $\mathbf{A}$ makes a primitive query, the transcript contains both top and bottom queries. For calls M^i to the construction , we compute $(U^i, V^i) \leftarrow \textsc{Hash}[\widetilde{\pi}](M^i)$, add (U^i, V^i, M^i) to τ_h, and add primitive queries for each top-lane call in $\textsc{Hash}$ and the corresponding matching query for the bottom-lane call to τ_p, i.e., $(V_j^i, (M_{j,1}^i, M_{j,2}^i), U_j^i, Y_j^i)$ and $(V_j^i, (M_{j,1}^i, M_{j,2}^i), U_j^i \oplus 1, Y_j^i)$.

[1] Attributed to Meyer in [17] but well-known as Davies-Meyer scheme, cf. [50].

Let $\exists^*$ mean "there exist pairwise distinct". We define the hash-collision event

$$\mathsf{coll} \stackrel{\text{def}}{=} \exists^*(U^i, V^i, M^i), (U^j, V^j, M^j) \in \tau_c : \textsc{Hash}[\widetilde{\pi}](M^i) = \textsc{Hash}[\widetilde{\pi}](M^j)\,.$$

We further define

$$\mathsf{vmulticoll}(r) \stackrel{\text{def}}{=} \exists^*(U^1, V^1, M^1), \ldots, (U^r, V^r, M^r) \in \tau_c : V^1 = \cdots = V^r\,.$$

Lemma 4. Let $n \geq 4$ and $H[\widetilde{\pi}]$ be short for $\textsc{Hash}[\widetilde{\pi}]$ with $\widetilde{\pi} \twoheadleftarrow \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$. Let $\mathbf{A}$ be an adversary on $H[\widetilde{\pi}]$ so that τ_p contains at most $q_p \leq 2^n/8$ primitive queries. Then,

$$\Pr\left[\mathsf{coll} \vee \mathsf{vmulticoll}(n)\right] \leq \frac{5q^2 + 2q}{2^{2n}}\,.$$

Proof. We define three bad events during the interaction of $\mathbf{A}$ with its oracles:

- bad_1: A collision occurs between the outputs of compression-function calls. There exist four pairwise distinct queries $(V^i, (M_1^i, M_2^i), U^i, Y^i)$, $(V^i, (M_1^i, M_2^i), U^i \oplus 1, Y'^i)$, $(V^j, (M_1^j, M_2^j), U^j, Y^j)$, $(V^j, (M_1^j, M_2^j), U^j \oplus 1, Y'^j) \in \tau$ such that

$$U^i \oplus Y^i = U^j \oplus Y^j \quad \text{and} \quad (U^i \oplus 1) \oplus Y'^i = (U^j \oplus 1) \oplus Y'^j\,.$$

- bad_2: The initival vector of $\textsc{Hash}$ is hit, i.e., $\exists(U, V, (M_1, M_2), Y), (U \oplus 1, V, M_1, M_2, Y') \in \tau_p$ s.t.

$$\widetilde{\pi}_V^{M_1,M_2}(U) \oplus U = U_0 \quad \text{and} \quad \widetilde{\pi}_V^{M_1,M_2}(U \oplus 1) \oplus (U \oplus 1) = V_0\,.$$

- bad_3: An n-multi-collision occurs in the bottom row, i.e. $\exists^*(V^1, (M_1^1, M_2^1), U^1, Y^1), \ldots, (V^n, (M_1^n, M_2^n), U^n, Y^n) \in \tau_p$ such that $U^1 \oplus 1 \oplus Y^1 = \cdots = U^n \oplus 1 \oplus Y^n$.

bad_1. Since the inputs to the blocks differ, each output from $\widetilde{\pi}$ is sampled uniformly at random from a set of at least $2^n - 2q$ values. Thus, the probability for a collision in $U^i \oplus Y^i = U^j \oplus Y^j$ is at most $1/(2^n - 2q)$. A similar argument can be derived for a collision in V. Over all query pairs, it follows that

$$\Pr\left[\mathsf{bad}_1\right] \leq \sum_{1 \leq j < i \leq q} \Pr[H(X^i) = H(X^j)] \leq \frac{\binom{q}{2}}{(2^n - 2q)^2} \leq \frac{q^2}{2^{2n}}\,.$$

using $q \leq 2^n/8$.

bad_2. The probability that an output produces $Y = U$ and $Y' = U \oplus 1$ is at most $1/(2^n - 2q)$. This holds for both calls. Since $q \leq 2^n/8$, it follows that

$$\Pr\left[\mathsf{bad}_2\right] \leq \sum_{i=1}^{q} \frac{1}{(2^n - 2q)^2} \leq \frac{2q}{2^{2n}}\,.$$

bad_3. Given Lemma 3, we have that

$$\frac{(2q)^n}{n! \cdot 2^{(n-1)n}}$$

For $n \geq 4$, it holds that $n! \geq 2^n$. Thus,

$$\Pr\left[\mathsf{bad}_3\right] \leq \left(\frac{2q}{2^n}\right)^n \leq \frac{4q^2}{2^{2n}}\,.$$

The union bound yields

$$\Pr[\mathsf{bad}] \leq \sum_{i=1}^{3} \Pr[\mathsf{bad}_i] \leq \frac{5q^2 + 2q}{2^{2n}}\,.$$

It remains to show that every other transcript cannot lead to a collision or to an n-semi-collision in V, i.e., to any of the bad events. We define a good attainable transcript as a transcript that is not bad, i.e., no bad event occurs.

Absence of Collisions. Let distinct $(U, V, M), (U', V', M') \in \tau_c$ and m denote the number of $2n$-bit blocks of M and m' that of M'. Assume that M_{TAIL} is the maximal common suffix of M and M':

$$M \stackrel{\text{def}}{=} (M_{\text{HEAD}} \,\|\, M_i \,\|\, M_{\text{TAIL}})$$
$$M' \stackrel{\text{def}}{=} (M'_{\text{HEAD}} \,\|\, M'_j \,\|\, M_{\text{TAIL}})\,,$$

where the length of $|M_{\text{HEAD}}|$, $|M'_{\text{HEAD}}|$, and $|M_{\text{TAIL}}|$ are multiples of $2n$ bits. There exist two cases:

Case (1): Either $(M_{\text{HEAD}} \,\|\, M_i)$ or $(M'_{\text{HEAD}} \,\|\, M'_j)$ is empty. W.l.o.g., assume that $(M_{\text{HEAD}} \,\|\, M_i)$ is empty. Since U is not empty, M_{TAIL} is not empty. Furthermore, it holds that $H[\widetilde{\pi}](M'_{\text{HEAD}} \,\|\, M'_j) \neq (U_0, V_0) = (0^n, 0^n)$ since the transcript is good. Thus, the first-block calls in M_{TAIL} are different. Since we excluded non-trivial collisions in good transcripts, this leads to $(U, V) \neq (U', V')$.

Case (2): Neither $(M_{\text{HEAD}} \,\|\, M_i)$ nor $(M'_{\text{HEAD}} \,\|\, M'_j)$ is empty. Since we excluded non-trivial collisions in good transcripts, this leads to

$$H[\widetilde{\pi}](M_{\text{HEAD}} \,\|\, M_i) \neq H[\widetilde{\pi}](M'_{\text{HEAD}} \,\|\, M'_j)\,.$$

If tail is empty, we are done. If tail is not empty, no collision can follow since the transcript is good.

Absence of n-Multi-collisions. It remains to show that there is no n-multi-collision in a good transcript. Let $(M_1^i, \ldots, M_{m^i}^i) \xleftarrow{2n} M^i$ denote the i-th message in $2n$-bit blocks after padding and length encoding. Since the transcript is good and no message is empty, the tuples consisting of the respective final message input blocks and the final chaining values, $(M_{m^i}, U_{m^i-1}, V_{m^i-1})$, are necessarily pairwise distinct. By the definition of the compression function, an n-multi-semi-collision is equivalent to an n-multi-collision of the Davies-Meyer compression function. Since we assume that the transcript is good, this cannot occur. □

C.4 Composition Theorem

We restate the theorem here to aid the reader.

Theorem 3 (qCIML2 Security). Let $\widetilde{\pi} \in \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$, $K \twoheadleftarrow \mathcal{K}$, and $n \geq 4$. Let $\mathbf{A}$ be a qCIML2-adversary on $\Pi[\widetilde{\pi}]_K = \mathsf{TEDT2}[\widetilde{\pi}]_K$ that makes at most q_c construction queries of at most σ message blocks and σ_a associated-data blocks in total and q_p primitive queries. Let q_e be the numbers of encryption construction queries, q_d the cumulative decryption construction queries to the oracles and μ be the number of encryption queries with repeating nonces in non-challenge encryption queries. Let σ_p be the number of ideal-primitive calls in all primitive and construction queries of $\mathbf{A}$, where $\sigma \leq 3\sigma + \sigma_a + 2q_c + q_p \leq 2^{n-3}$. Let the sets of leakage functions $\mathcal{L}_E$ and $\mathcal{L}_D$ be as defined in Section 6. Then

$$\begin{aligned}\mathbf{Adv}^{\mathsf{qCIML2}}_{\Pi[\widetilde{\pi}]_K, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D}(q_e, q_d, \sigma, q_p) &\leq \mathbf{Adv}^{\mathsf{TPRP}}_{\widetilde{\pi}}(2q_c) + \mathbf{Adv}^{\mathsf{STPRP}}_{\widetilde{\pi}}(q_e, q_d) \\ &+ \frac{2n(q_p + q_d) + 2q_d + 3(\sigma + \sigma_a) + 1}{2^n} + \frac{9\sigma_p^2 + 2\sigma_p + 9q_p(q_d + \mu) + q_c^2}{2^{2n}}\,.\end{aligned}$$

Proof. The proof follows the steps of that for TEDT by [9].

Transcript. We define a transcript

$$\tau \stackrel{\text{def}}{=} (\{K\}, \tau_c, \tau_p, \tau_{\text{KDF}}, \tau_{\text{TGF}})$$

that contains the queries of $\mathbf{A}$ and their corresponding responses:

- τ_p contains all primitive queries (K^i, S^i, X^i, Y^i) of $\mathbf{A}$ and their corresponding responses.
- For all construction queries of $\mathbf{A}$, τ_{KDF} consists of exactly all queries to the ideal key-derivation function $(\widetilde{\pi}_1)^{S^i}_{K^i}(N^i)$ and $(\widetilde{\pi}_2)^{S^i}_{K^i}(N^i)$ and the responses.
- Similarly, for all construction queries, τ_{TGF} consists of exactly the queries to the ideal tag-generation function and the responses: (U^i, V^i, N^i, T^i), where $T^i \leftarrow (\widetilde{\pi}_3)^{N^i, V^i}_{K^i}(U^i)$.
- Moreover, for all construction queries of $\mathbf{A}$, the challenger considers all internal queries $(K^i_j, S^i_j, X^i_j, Y^i_j)$ to $\widetilde{\pi}$,

$$Y^i_j \leftarrow \widetilde{\pi}^{S^i_j}_{K^i_j}(X^i_j) \quad \text{or its inverse} \quad X^i_j \leftarrow (\widetilde{\pi}^{S^i_j}_{K^i_j})^{-1}(Y^i_j)\,,$$

  and stores them in a transcript τ_c.

In both the real and ideal world, the challenger samples $K \twoheadleftarrow \mathcal{K}$ and defines the contents of τ_{KDF} and τ_{TGF} as in the real world. Similarly, all primitive queries are answered by $\widetilde{\pi}$ and stored in τ_p, and all construction queries to the encryption and to the non-challenge decryption oracles are answered by the construction as in the real world.

All such queries are given to $\mathbf{A}$. We assume that the KDF and the TGF components do not leak the long-term key K. We only add the long-term key

$K \twoheadleftarrow \mathcal{K}$ to the transcript as a usual proof technique to bound the transcript probability. However, K is not given to $\mathbf{A}$.

The number of primitive queries for encrypting an a-block associated data A and an m-block message M (where we consider n-bit blocks)

- in the hash function is $m+a+2$ since at most two additional calls are added by the added message length;
- in the encryption process is $2m$.

Hence, at most $3m + a + 2$ primitive queries are added to τ_c for such a tuple of message and associated data. The same holds for valid decryption queries. At most $m + a + 2$ primitive calls occur in invalid decryption queries. Thus, the number of primitive queries, σ_p, is at most $\sigma_p \leq 3\sigma + \sigma_a + 2q_c + q_p$ in construction and additional primitive queries. Furthermore, we consider a list of $\tau_h = \{(X^i, U^i, V^i)\}$ for $i \in [q_c]$ for the hash-function inputs X^i and outputs (U^i, V^i). Note that $X^i = \textsc{concat}_n(A^i, C^i)$ is never the empty string since the length has been appended.

We define the maximal multiplicity of values V as

$$\ell_V \stackrel{\text{def}}{=} \max_{V \in \mathbb{F}_2^n} \left|\left\{(X^i, U^i, V^i) \in \tau_h : V^i = V\right\}\right| .$$

We define an attainable transcript τ as bad if any of the following events occur:

- bad_1: There exists $V \in \mathbb{F}_2^n$ such that $(X, U, V) \in \tau_h$ and $\ell_V \geq n$.
- bad_2: There exist distinct $(X^i, U^i, V^i), (X^j, U^j, V^j) \in \tau_h$ with $X^i \neq X^j$ but $(U^i, V^i) = (U^j, V^j)$.
- bad_3: There exists $(K^i, S^i, X^i, Y^i) \in \tau_{\text{KDF}}$ and $(K^j, S^j, X^j, Y^j) \in \tau_p$ such that $(K^i, S^i, X^i) = (K^j, S^j, X^j)$ or $(K^i, S^i, Y^i) = (K^j, S^j, Y^j)$.
- bad_4: There exists $(K^i, S^i, X^i, Y^i) \in \tau_{\text{TGF}}$ and $(K^j, S^j, X^j, Y^j) \in \tau_p$ such that $(K^i, S^i, X^i) = (K^j, S^j, X^j)$ or $(K^i, S^i, Y^i) = (K^j, S^j, Y^j)$.
- bad_5: There exists a collision of a TGF, KDF, encryption, or primitive query with a primitive query used in the hash function.
- bad_6: There exist
  - a TGF query $(K^i, U^i, V^i, N^i, T^i) \in \tau_{\text{TGF}}$ that was generated from a query $(N^i, A^i, M^i, C^i, T^i)$ and
  - a hash query $(X^j, U^j, V^j) \in \tau_h$ with $X^j = \textsc{pad}(A^j, C^j)$ and $H(X^j) = (U^j, V^j) = (U^i, V^i)$ and $(\widetilde{\pi}_3)_K^{N^i, V^i}(U^i) = T^i$.
- bad_7: There exist distinct queries (K^i, S^i, X^i, Y^i) and $(K^j, S^j, X^j, Y^j) \in \tau_{\text{KDF}}$, such that $(K^i, S^i) \neq (K^j, S^j)$ but $Y^i = Y^j$.

Otherwise, we call τ good. We define BadT for the set of all bad attainable transcripts in the ideal world and GoodT for the set of all good attainable transcripts.

C.5 Bad Transcripts

Lemma 5. It holds that

$$\Pr[\Theta_{\text{ideal}} \in \textsc{BadT}] \leq \frac{2n(q_p + q_d) + 3(\sigma + \sigma_a) + 1}{2^n} + \frac{9\sigma_p^2 + 2\sigma_p + 9q_p(q_d + \mu) + q_c^2}{2^{2n}} .$$

Proof. In the following, we upper bound the probability of each bad event.

bad_1 and bad_2. For bad_1, we have an multi-semi-collision of n colliding values in V that have been generated by the Davies-Meyer construction inside $\textsc{Hash}[\widetilde{\pi}]$. We know this event from Lemma 3. Lemma 4 includes both bad events. We have to insert σ_p for the number of primitive queries in the construction hash and primitive queries. Thus, we can upper bound the probability of this event as

$$\Pr\left[\mathsf{bad}_1 \vee \mathsf{bad}_2\right] \leq \frac{5\sigma_p^2 + 2\sigma_p}{2^{2n}} .$$

bad_3. Among the two KDF primitives, the q_p primitive queries can target only one of them each. Since all KDF queries employ the same key and the adversary needs to guess and hit the correct key, the probability is at most

$$\Pr\left[\mathsf{bad}_3\right] \leq \frac{q_p}{2^n} .$$

bad_4. Similarly as for bad_2, the transcript contains q_p primitive queries and q_c queries that call the tag-generation function exactly once each. Note that no internal encryption or decryption query in the encryption or decryption process can hit K by accident due to the domain separation. We can condition on the absence of more than n-multi-collisions. Hence, each primitive query can cover at most n values, and can collide only with nonce-repeating or decryption queries. Since the adversary needs to guess and hit the correct key, the probability is upper bounded by

$$\Pr\left[\mathsf{bad}_4 \,|\neg \mathsf{bad}_1\right] \leq \frac{n \cdot q_p}{2^n} .$$

bad_5. The hash function is the only part of $\mathsf{TEDT2}$ wherein domain separation is absent. Hence, collisions in the tweakeys between primitive calls in the hash function and other primitive queries or primitive calls in KDF, encryption, or TGF primitive queries can occur. There exist two kinds of queries: (1) initial hash queries and (2) intermediate hash queries. Each of them can collide with the tweakeys in TGF, KDF, or in primitive calls in the encryption.

Collisions with Initial Hash Queries. For initial-value queries, the probability to hit the long-term secret K is

$$\Pr\left[V_0 = K\right] = \frac{1}{2^n}$$

for the single initial value. To hit a tweakey of any of at most $2\sigma + 2q_c$ primitive calls in encryption queries with the key V_0 is

$$\sum_{i\in[q_c]} \sum_{j\in[m^i]} \Pr\left[K^i_j = V_0\right] \leq \frac{2\sigma + 2q_c + 1}{2^n}\,.$$

Hence, for initial-value queries, the probability is at most

$$\frac{2\sigma + 2q_c + 1}{2^n}\,.$$

Collisions with Intermediate Hash Queries. For the up to $\sigma + \sigma_a \leq \sigma_p$ intermediate-hash queries, the probability to hit the key and the nonce input of any of at most $2\sigma + 2q_c \leq \sigma_p$ primitive calls in encryption queries is at most

$$\Pr\left[(U^i_j, V^i_j) = (N^{i'}, K^{i'}_{j'})\right] \leq \frac{(\sigma + \sigma_a)\cdot 2(\sigma + q_c)}{(2^n - \sigma)^2} \leq \frac{4\sigma_p^2}{2^{2n}}\,.$$

The probability to hit the long-term secret K in the TGF or KDF with any intermediate hash query is at most

$$\sum_{i\in[q_c]} \sum_{j\in[m^i]} \Pr\left[V^i_j = K\right] = \frac{\sigma + \sigma_a}{2^n}$$

since the single secret has to be hit. It follows that

$$\Pr\left[\mathsf{bad}_5\right] \leq \frac{4\sigma_p^2}{2^{2n}} + \frac{3\sigma + 3\sigma_a + 1}{2^n}\,.$$

bad_6. It remains to upper bound the probability that $\mathbf{A}$ forges successfully, for a transcript wherein the previous events $\mathsf{bad}_1, \ldots, \mathsf{bad}_4$ do not occur. Then, there exists (1) a TGF query $(K^i, U^i, V^i, N^i, T^i) \in \tau_{\mathrm{TGF}}$ that was generated from a query $(N^i, A^i, M^i, C^i, T^i)$ and (2) a hash query $(X^j, U^j, V^j) \in \tau_h$ with $X^j = \textsc{pad}(A^j, C^j)$ and $H(X^j) = (U^j, V^j) = (U^i, V^i)$ and $(\widetilde{\pi}_3)^{N^i, V^i}_K(U^i) = T^i$.

The hash query has produced two primitive queries $(U^k, V^k, (X^k_1, X^k_2), Y^k)$ and $(U^k \oplus 2 \oplus 1, V^k, (X^k_1, X^k_2), Y'^k)$ such that

$$U^k \oplus Y^k = U^j(U^k \oplus 2 \oplus 1) \oplus Y^k = V^j$$

and $X^k = (X^k_{1,1}, X^k_{1,2}, \ldots, X^k_{m^i,1}, X^k_{m^i,2})$. We distinguish between two cases.

Case 1: The TGF Query Followed the Primitive Queries. If the TGF query $(K^i, U^i, V^i, N^i, T^i)$ was in forward direction, it would have been result of an encryption query. However, since we excluded hash collisions, it can also not be due to an earlier distinct encryption query $(N^j, A^j, M^j, C^j, T^j)$; if it resulted from the same query as an earlier encryption query, it would be a prohibited trivial decryption query. Hence, the TGF query must be in backward direction.

For a transcript wherin bad_1 through bad_4 did not occur, the number of earlier hash queries (X^k, U^k, V^k) with $V^k = V^i$ is at most n. This implies that the number of target values U is also at most n. For each of them, the value U is chosen uniformly at random from a set of size at least $2^n - \sigma$ and the probability that it is correct is therefore

$$\Pr\left[\left((\widetilde{\pi}_3)_K^{N^i,V^i}\right)^{-1}(T^i) = U^i\right] \leq \frac{1}{2^n - \sigma} \leq \frac{2}{2^n}\,,$$

and over at most n queries that could be hit by q_d decryption queries

$$\Pr\left[\text{Case 1}\right] \leq \frac{2nq_d}{2^n}\,.$$

Case 2: The Primitive Queries Followed The TGF Query. In this case, the hash function output must match the values of the TGF query. For $\widetilde{\pi}$, the probability to hit U^i and V^i for any primitive queries is at most

$$\Pr\left[\begin{matrix} U^i = \widetilde{\pi}_{X_1^k}^{V^k,X_2^k}(U^k) \oplus U^k \\ V^i = \widetilde{\pi}_{X_1^k}^{V^k,X_2^k}(U^k \oplus 2 \oplus 1) \oplus U^k \oplus 2 \oplus 1 \end{matrix}\right] \leq \frac{1}{(2^n - \sigma)^2} \leq \frac{4}{2^{2n}}\,.$$

At most $2q_p$ primitive queries can occur. While they can match potentially all q_d decryption TGF queries, in encryption direction, they can only match any of μ nonce-repeating TGF queries. Therefore

$$\Pr\left[\text{Case 2}\right] \leq \frac{8q_p(q_d + \mu)}{2^{2n}}\,.$$

It follows that

$$\Pr\left[\mathsf{bad}_6 \;\middle|\; \bigwedge_{i=1}^{5} \neg\mathsf{bad}_i\right] \leq \frac{2nq_d}{2^n} + \frac{8q_p(q_d + \mu)}{2^{2n}}\,.$$

bad_7. Finally, we treat the case that two construction queries produced equal initial keys for the encryption procedure. In the real world, the KDF computes the initial value (K_1^i, T_1^i) as $(\widetilde{\pi}_1)_{K^i}^{S^i}(0^n) \,\|\, (\widetilde{\pi}_2)_{K^i}^{S^i}(0^n)$, where S^i contains the full nonce N^i. Since the long-term keys and tweak domains do not change, it holds that $K^i = K^j$ and it must hold that $S^i \neq S^j$, which implies $N^i \neq N^j$. The probability for distinct construction queries i, j with $S^i \neq S^j$ to have $(K_1^i, T_1^i) = (K_1^j, T_1^j)$ can be upper bounded by

$$\Pr\left[\mathsf{bad}_7 \;\middle|\; \bigwedge_{i=1}^{6} \neg\mathsf{bad}_i\right] \leq \frac{q_c^2 + q_c q_p}{2^{2n}}\,.$$

Then, the claim in Lemma 5 follows from

$$\begin{aligned}\Pr[\mathsf{bad}] &\le \sum_{i=1}^{3} \Pr[\mathsf{bad}_i] + \Pr[\mathsf{bad}_4 | \neg\mathsf{bad}_1] + \Pr[\mathsf{bad}_5] \\ &\quad + \Pr\left[\mathsf{bad}_6 \;\middle|\; \bigwedge_{i=1}^{5} \neg\mathsf{bad}_i\right] + \Pr[\mathsf{bad}_7] \\ &\le \frac{5\sigma_p^2 + 2\sigma_p}{2^{2n}} + \frac{q_p}{2^n} + \frac{nq_p}{2^n} + \frac{4\sigma_p^2}{2^{2n}} + \frac{3\sigma + 3\sigma_a + 1}{2^n} \\ &\quad + \frac{2nq_d}{2^n} + \frac{8q_p(q_d+\mu)}{2^{2n}} + \frac{q_c(q_c+q_p)}{2^{2n}} \\ &\le \frac{2n(q_p+q_d) + 3(\sigma+\sigma_a) + 1}{2^n} + \frac{9\sigma_p^2 + 2\sigma_p + 9q_p(q_d+\mu) + q_c^2}{2^{2n}}\,. \quad \square\end{aligned}$$

C.6 Good Transcripts

Lemma 6. For an attainable good transcript τ and $q_c \le 2^{n-1}$, it holds that

$$\frac{\Pr[\Theta_{\text{real}} = \tau]}{\Pr[\Theta_{\text{ideal}} = \tau]} \ge 1 - \frac{2q_d}{2^n}\,.$$

Proof. First, we consider the probability of the transcript in the ideal world. Since the key is sampled uniformly at random, the probability for each possible value of K is $|\mathcal{K}|^{-1}$. We define

$$\nu \stackrel{\text{def}}{=} \left|\left\{N^i\right\}_{i\in[q_c]}\right|$$

for the number of pairwise distinct nonces over all construction queries. We further define marginal variables Θ^x_{real} and Θ^x_{ideal} with $x \in \{\text{KDF}, \text{TGF}, p, c\}$ for the partial transcripts τ_x.

We use a set $\mathcal{KS}_{c,p} =^{\text{def}} \{(K^i, S^i) : (K^i, S^i, X^i, Y^i) \in \tau_c \cup \tau_p, \text{ for any } X^i, Y^i \in \mathbb{F}_2^n\}$ and $r =^{\text{def}} |\mathcal{KS}_{c,p}|$. It holds that $r \le \sigma_p$ and the inequality is strict whenever distinct queries i, j with $(K^i, S^i) = (K^j, S^j)$ exist. In the following, we use $\mathsf{KS}_{c,p}$ to be a sequence of exactly the elements in $\mathcal{KS}_{c,p}$, where we reorder its elements lexicographically ascendingly (only for the sake concreteness) by interpreting them $2n$-bit integers. Then, we can write

$$\ell^i_{c,p} \stackrel{\text{def}}{=} \left|\left\{(K, S, X, Y) \in \tau_c \cup \tau_p : (K, S) = (K^i, S^i)\right\}\right|$$

for the number of queries with tweakey (K^i, S^i), for all $i \in [r]$. Note that $\sum_{i=1}^{r} \ell^i_{c,p} = |\tau_c \cup \tau_p|$.

Similarly, we create a second such set $\mathcal{KS}_{\text{TGF}} =^{\text{def}} \{(K^i, S^i) : (K^i, S^i, X^i, Y^i) \in \tau_{\text{TGF}}, \text{ for any } X^i, Y^i \in \mathbb{F}_2^n\}$, $s =^{\text{def}} |\mathcal{KS}_{\text{TGF}}|$, and define

$$\ell^i_{\text{TGF}} \stackrel{\text{def}}{=} \left|\left\{(K, S, X, Y) \in \tau_{\text{TGF}} : (K, S) = (K^i, S^i)\right\}\right|$$

for the number of queries with tweakey (K^i, S^i), for all $i \in [s]$. Note that $\sum_{i=1}^{r} \ell^i_{\text{TGF}} = |\tau_{\text{TGF}}|$.

Ideal World. The ideal world calls two distinct permutations for each nonce. Hence, the probability in the ideal world to generate τ_{KDF} is

$$\Pr[\Theta^{\mathrm{KDF}}_{\mathrm{ideal}} = \tau_{\mathrm{KDF}}] = \frac{1}{(2^{2n})^{\nu}} \,.$$

The probability of permutation-query outputs of $\tau_c \cup \tau_p$, over all key-tweak tuples, is

$$\Pr[\Theta^{p}_{\mathrm{ideal}} = \tau_c \cup \tau_p] = \prod_{i=1}^{r} \frac{1}{(2^n)_{\ell^i_{c,p}}} \,.$$

Similarly, in a good transcript, the probability of the outputs of τ_{TGF} in encryption direction only would be

$$\Pr[\Theta^{\mathrm{TGF}}_{\mathrm{ideal}} = \tau_{\mathrm{TGF}}] = \prod_{i=1}^{s} \frac{1}{(2^n)_{\ell^i_{\mathrm{TGF}}}} \,.$$

Note that the challenge-decryption queries return $\perp$ every time with probability one. Thus, the probability of the ideal transcript is given by

$$\Pr\left[\Theta_{\mathrm{ideal}} = \tau\right] = \frac{1}{|\mathcal{K}|} \cdot \frac{1}{(2^{2n})^{\nu}} \cdot \prod_{i=1}^{r} \frac{1}{(2^n)_{\ell^i_{c,p}}} \cdot \prod_{i=1}^{s} \frac{1}{(2^n)_{\ell^i_{\mathrm{TGF}}}} \,.$$

Real World. For the real world, the key is sampled as before and has probability of $|\mathcal{K}|^{-1}$ to occur. Moreover, the same probability of outputs for the KDF holds in the real world:

$$\Pr[\Theta^{\mathrm{KDF}}_{\mathrm{real}} = \tau_{\mathrm{KDF}}] = \frac{1}{(2^{2n})^{\nu}} \,.$$

Similarly as in the ideal world, all inputs to $\widetilde{\pi}$ in the real-world encryption have probability

$$\Pr[\Theta^{c,p}_{\mathrm{ideal}} = \tau_c \cup \tau_p] = \prod_{i=1}^{s} \frac{1}{(2^n)_{\ell^i_{c,p}}} \,.$$

It remains to treat the results of the tag-generation function. In a good transcript, we have to consider the probability of the outputs of τ_{TGF} in encryption direction, and the fact that all fresh decryption queries will produce $\perp$. For the latter, it must hold that the random tweakable permutation $\widetilde{\pi}_3$ does not map the inputs to the provided value of T^i. Note that, in a good transcript, there are no duplicate queries and no collision of inputs $(K^i, S^i, X^i) = (K^j, S^j, X^j)$ or $(K^i, S^i, T^i) = (K^j, S^j, T^j)$ for distinct construction queries i and j. We make the distinguisher stronger than it is and assume it can keep the tweak constant for all challenge queries. The probability that $\widetilde{\pi}_3$ would not produce T for the

i-th construction query can be upper bounded by $1 - \frac{1}{2^n-(i-1)}$ and by $1 - \frac{q_d}{2^n-q_c}$ over all such queries. Following a similar argument as [15], we can

$$\Pr[\Theta^{\mathrm{TGF}}_{\mathrm{ideal}} = \tau_{\mathrm{TGF}}] \geq \prod_{i=1}^{s} \frac{1}{(2^n)_{\ell^i_{\mathrm{TGF}}}} \left(1 - \frac{q_d}{2^n - q_c}\right) .$$

Note that the challenge-decryption queries return $\perp$ every time with probability one. We obtain for the interpolation ratio

$$\begin{aligned}\frac{\Pr[\Theta_{\mathrm{real}} = \tau]}{\Pr[\Theta_{\mathrm{ideal}} = \tau]} &\geq \frac{|\mathcal{K}|(2^{2n})^{\nu} \cdot \prod_{i=1}^{r} (2^n)_{\ell^i_{c,p}} \cdot \prod_{i=1}^{s} (2^n)_{\ell^i_{\mathrm{TGF}}} \left(1 - \frac{q_d}{2^n-q_c}\right)}{|\mathcal{K}| \cdot (2^{2n})^{\nu} \cdot \prod_{i=1}^{s} (2^n)_{\ell^i_{c,p}} \cdot \prod_{i=1}^{s} (2^n)_{\ell^i_{\mathrm{TGF}}}} \\ &\geq 1 - \frac{q_d}{2^n - q_c} \geq 1 - \frac{2q_d}{2^n} .\end{aligned}$$

As a result, the claim in Lemma 6 follows.

D qCPA$mL2 Analysis of TEDT2 with RCTR[$\widetilde{\pi}$]

We restate the lemma to aid the reader.

Lemma 6 (qCPA$mL2 Security of TEDT2 with RCTR). Let $\widetilde{\pi} \twoheadleftarrow \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$. Let $\mathbf{A}$ be an qCPA$mL2 adversary on $\mathit{\Pi}[\widetilde{\pi}]_K = \mathsf{TEDT2}[\widetilde{\pi}]_K$ that asks at most q_e encryption queries and q_d decryption queries of at most σ blocks in total and q_p primitive queries. Let $F[\widetilde{\pi}]$ be an iteration of $\mathsf{RCTR}[\widetilde{\pi}]$. Let $n \geq 2$, $\sigma \leq 2^{n-3}$, and let $\mathcal{L}_E$ and $\mathcal{L}_D$ be as defined at the top of Section 7. Then

$$\begin{aligned}\mathbf{Adv}^{\mathsf{qCPA\$mL2}}_{\mathit{\Pi}[\widetilde{\pi}]_K, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D}(q_e, q_d, \sigma, q_p) &\leq \frac{4q_p(\sigma + q_c) + 4(\sigma + q_c)^2}{2^{2n}} \\ &+ 2\sigma \mathbf{Adv}^{\mathsf{LUP\text{-}4}}_{F[\widetilde{\pi}], \mathcal{L}^{\mathsf{in}}, \mathcal{L}^{\mathsf{out}}}(p, \sigma + q_c + q_p) + \sigma \mathbf{Adv}^{\mathsf{XOR\$}}_{F[\widetilde{\pi}], \mathcal{L}^{\mathsf{out}}, \mathcal{L}^{\oplus}}(p, 2\sigma + 2q_c + q_p) .\end{aligned}$$

Proof. The proof follows similar steps as the muCCAmL2 proof by [9] on TEDT. Though, we consider only a single user and we can safely restrict decryption queries to such that are results of previous encryption queries since here, we assume that the adversary is unable to produce a valid forgery. In the following, we employ a sequence of games $\mathcal{G}_1$ through $\mathcal{G}_4$. In that sequence, Game $\mathcal{G}_1$ represents the left-hand side of Equation (3),

$$\underset{\mathbf{A}}{\Delta}(\widehat{\mathcal{E}}_K, \widehat{\mathcal{E}}^{ch}_K, \widehat{\mathcal{D}}^{ch}_K, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D \,;\, \widehat{\mathcal{E}}_K, \widehat{\$}_{\mathcal{E}}, \widehat{\$}_{\mathcal{D}}, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D) ,$$

with the restrictions $O_{2,N} \not\succ O_{1,N}$, $\{O_{1,N}, O_{2,N}\} \not\succ O_{2,N}$, and $O_2 \nuparrow O_3$. Game $\mathcal{G}_4$ represents the right-hand side of Equation (3). We will move stepwise from Game $\mathcal{G}_i$ to $\mathcal{G}_{i+1}$ for $i = 1..3$. For each step, we define an adversary $\mathbf{A}_i$ whose goal is to distinguish between Game $\mathcal{G}_i$ to $\mathcal{G}_{i+1}$. We write $\Delta\mathcal{G}_{i,j}$ for the maximal advantage for all adversaries $\mathbf{A}_i$ to distinguish between Game $\mathcal{G}_i$ and $\mathcal{G}_j$, where all adversaries use equal query resources as $\mathbf{A}$.

We introduce an IdealPRG instead of RCTR$[\widetilde{\pi}]$ in the process. We denote the latter as RealPRG$[\widetilde{\pi}]$. We use a shorthand notation for the games, where we focus only on the PRG used in the oracles O_1, O_2, O_3 and indicate by M that real messages are encrypted, and by \$ that random messages are encrypted:

- $\Delta\mathcal{G}_{1,4} =^{\text{def}} \Delta_{\mathbf{A}}(\mathsf{RealPRG}[\widetilde{\pi}](M)\,;\,\mathsf{RealPRG}[\widetilde{\pi}](\$))$
- $\Delta\mathcal{G}_{1,2} =^{\text{def}} \Delta_{\mathbf{A}}(\mathsf{RealPRG}[\widetilde{\pi}](M)\,;\,\mathsf{IdealPRG}[\widetilde{\pi}](M))$
- $\Delta\mathcal{G}_{2,3} =^{\text{def}} \Delta_{\mathbf{A}}(\mathsf{IdealPRG}[\widetilde{\pi}](M)\,;\,\mathsf{IdealPRG}[\widetilde{\pi}](\$))$
- $\Delta\mathcal{G}_{3,4} =^{\text{def}} \Delta_{\mathbf{A}}(\mathsf{IdealPRG}[\widetilde{\pi}](\$)\,;\,\mathsf{RealPRG}[\widetilde{\pi}](\$))$

Note that the advantage of the final step is already considered in the second step by $\Delta\mathcal{G}_{1,2}$. Using the triangle inequality, we can upper bound the advantage by

$$(3) \le \Delta\mathcal{G}_{1,4} \le \sum_{i=1}^{3} \Delta\mathcal{G}_{i,i+1}\,.$$

Difference $\Delta\mathcal{G}_{1,2}$. Next, we define an ideal PRG as on the right-hand side of Algorithm 6. For each construction query (N^j, M^j), it

1. Samples all keys and tweaks K_i^j, T_i^j, as well as all would-be primitive outputs $Y_{i,1}^j$, $Y_{i,2}^j$ uniformly and independently at random from $\mathbb{F}_2^n$,
2. Computes C^j as $C_i^j = \text{TRUNC}_{|M_i^j|}(Y_i^j) \oplus M_i^j$,
3. Computes the leakages $L^{\oplus}$, L^{in}, and L^{out} with the help of its primitive oracle $\widetilde{\pi}$ and $\widetilde{\pi}^{-1}$, collects the would-be leakages into a vector $\mathbf{L}^j$, and
4. Outputs $(\mathbf{L}^j, C^j)$.

bad Query. We define bad as the event that the tweakey (K, S) used in a call to $\widetilde{\pi}_K^S$ for any block in a construction query collides with any tweakey with a call to $\widetilde{\pi}^{\pm}$ in a primitive query or with a distinct block of a construction query. Formally, the first case is fulfilled iff there exists a construction query (N^j, M^j, C^j) with $(M_1^j, \ldots, M_{m^j}^j) \xleftarrow{2n} M^j$ and a primitive query (K^k, S^k, X^k, Y^k) such that for some $i \in [0..m^j]$, $j \in [q_c]$, and $k \in [q_p]$, it holds that $(K_i^j, S_i^j) = (K^k, S^k)$. There are 2σ calls to the primitives in RCTR if all queries have a multiple of $2n$ blocks. There can be up to $2q_c$ further primitive calls if the final $2n$-bit block of each message is partial. The second event is fulfilled iff there exists a second construction query $(N^{j'}, M^{j'}, C^{j'})$ with $(M_1^{j'}, \ldots, M_{m^{j'}}^{j'}) \xleftarrow{2n} M^{j'}$ such that $(K_i^j, S_i^j) = (K_{i'}^{j'}, S_{i'}^{j'})$ for any $j' \neq j$ and $i' \in [0..m^{j'}]$. Here, there can be a collision of tweakeys among those $2(\sigma + q_c)$ encrypted blocks. Over all queries, the probability is upper bounded by

$$\Pr[\mathsf{bad}] \le \frac{q_p \cdot 2(\sigma + q_c)}{2^{2n}} + \frac{\binom{2(\sigma+q_c)}{2}}{2^{2n}} \le \frac{q_p \cdot 2(\sigma + q_c) + 2(\sigma + q_c)^2}{2^{2n}}\,.$$

If no collision occurs among the guesses, then all tweakeys are fresh. In this case, there is would be no difference between both games $\mathcal{G}_1$ with the RealPRG and $\mathcal{G}_2$

Algorithm 6 Games $\mathcal{G}_1$ to $\mathcal{G}_3$. IdealPRG replaces RealPRG in $\mathcal{G}_2$. Note that the primitive oracles $\widetilde{\boldsymbol{\pi}}^{\pm}$ and leakage functions belong to all games. The boxed statements belong only to Game $\mathcal{G}_3$ and $\mathcal{G}_4$.

procedure INITIALIZE(τ)
  $b \leftarrow \{0,1\}$; $K \twoheadleftarrow \mathcal{K}$; $\mathcal{Q} \leftarrow \emptyset$; bad $\leftarrow$ false

function REALPRG$[\widetilde{\pi}]^N(M)$
  $\mathbf{L} \leftarrow []$
  $\mathcal{Q} \xleftarrow{\cup} \{(K,(6,0,0)),(K,(7,0,0))\}$
  $K_1 \leftarrow \widetilde{\boldsymbol{\pi}}_K^{6,0,0}(N)$
  $T_1 \leftarrow \widetilde{\boldsymbol{\pi}}_K^{7,0,0}(N)$
  $(M_1,\ldots,M_m) \xleftarrow{2n} M$
  for $i \leftarrow 1..m$ **do**
    for $d \leftarrow 0..3$ **do** $\mathcal{Q} \xleftarrow{\cup} (K_i,(d,i,T_i))$
    $K_{i+1} \leftarrow \widetilde{\boldsymbol{\pi}}_{K_i}^{0,i,T_i}(N)$
    $T_{i+1} \leftarrow \widetilde{\boldsymbol{\pi}}_{K_i}^{1,i,T_i}(N)$
    $Y_i \leftarrow \widetilde{\boldsymbol{\pi}}_{K_i}^{2,T_i,i}(N) \,\|\, \widetilde{\boldsymbol{\pi}}_{K_i}^{3,T_i,i}(N)$
    $\boxed{M_i \twoheadleftarrow \mathbb{F}_2^{|M_i|}}$
    $C_i \leftarrow \text{TRUNC}_{|M_i|}(Y_i) \oplus M_i$
    $L_i \leftarrow ($
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (0,i,T_i), N, K_{i+1})$,
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (1,i,T_i), N, T_{i+1})$,
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (2,i,T_i), N, Y_{i,1})$,
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (3,i,T_i), N, Y_{i,2})$,
    LEAKXOR$(L^{\oplus}, M_{i,1}, Y_{i,1}, C_{i,1})$,
    LEAKXOR$(L^{\oplus}, M_{i,2}, Y_{i,2}, C_{i,2}))$
    $\mathbf{L} \xleftarrow{\cup} L_i$
  $C \leftarrow C_1 \,\|\, \cdots \,\|\, C_m$
  return $(\mathbf{L}, C)$

function $\widetilde{\boldsymbol{\pi}}_K^{d,i,T}(X)$
  if $(K,(d,i,T)) \in \mathcal{Q}$ **then** bad $\leftarrow$ true
  return $\widetilde{\pi}_K^{d,i,T}(X)$

function $(\widetilde{\boldsymbol{\pi}}_K^{d,i,T})^{-1}(Y)$
  if $(K,(d,i,T)) \in \mathcal{Q}$ **then** bad $\leftarrow$ true
  return $(\widetilde{\pi}_K^{d,i,T})^{-1}(Y)$

procedure INITIALIZE(τ)
  $b \leftarrow \{0,1\}$; $K \twoheadleftarrow \mathcal{K}$; $\mathcal{Q} \leftarrow \emptyset$; bad $\leftarrow$ false

function IDEALPRG$^N(M)$
  $\mathbf{L} \leftarrow []$
  $\mathcal{Q} \xleftarrow{\cup} \{(K,(6,0,0)),(K,(7,0,0))\}$
  $K_1 \twoheadleftarrow \mathbb{F}_2^n$
  $T_1 \twoheadleftarrow \mathbb{F}_2^n$
  $(M_1,\ldots,M_m) \xleftarrow{2n} M$
  for $i \leftarrow 1..m$ **do**
    for $d \leftarrow 0..3$ **do** $\mathcal{Q} \xleftarrow{\cup} (K_i,(d,i,T_i))$
    $K_{i+1} \twoheadleftarrow \mathbb{F}_2^n$
    $T_{i+1} \twoheadleftarrow \mathbb{F}_2^n$
    $Y_{i,1} \twoheadleftarrow \mathbb{F}_2^n$
    $Y_{i,2} \twoheadleftarrow \mathbb{F}_2^n$
    $\boxed{M_i \twoheadleftarrow \mathbb{F}_2^{|M_i|}}$
    $C_i \leftarrow \text{TRUNC}_{|M_i|}(Y_i) \oplus M_i$
    $L_i \leftarrow ($
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (0,i,T_i), N, K_{i+1})$,
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (1,i,T_i), N, T_{i+1})$,
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (2,i,T_i), N, Y_{i,1})$,
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (3,i,T_i), N, Y_{i,2})$,
    LEAKXOR$(L^{\oplus}, M_{i,1}, Y_{i,1}, C_{i,1})$,
    LEAKXOR$(L^{\oplus}, M_{i,2}, Y_{i,2}, C_{i,2}))$
    $\mathbf{L} \xleftarrow{\cup} L_i$
  $C \leftarrow C_1 \,\|\, \cdots \,\|\, C_m$
  return $(\mathbf{L}, C)$

function LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K, T, X, Y)$
  return $[L^{\mathsf{in}}(K,T,X)]_p, [L^{\mathsf{out}}(K,T,Y)]_p$

function LEAKXOR$(L^{\oplus}, X, Y, Z)$
  return $[L^{\oplus}(X,Y)]_p, [L^{\mathsf{out}}(Y,Z)]_p$

with the IdealPRG in the black-box setting. Though, under leakage and absence of bad queries, we show that the advantage of a distinguisher $\mathbf{A}_{1,2}$ that tries to distinguish between Games $\mathcal{G}_1$ and $\mathcal{G}_2$ can be reduced to the advantage of an LUP-4 distinguisher $\mathbf{A}_u$ on an isolated call to the PRG by $\mathbf{A}_{1,2}$.

We define $F[\widetilde{\pi}]$ for one iteration of the LUP-4 game and an LUP-4 adversary $\mathbf{A}_u$ that runs an instance of $\mathbf{A}_{1,2}$. For each primitive query of $\mathbf{A}_{1,2}$, $\mathbf{A}_u$ simply forwards the query to its corresponding primitive oracle, collects the query together with the output in a transcript $\tau_p = \{(K^i, S^i, X^i, Y^i)\}$ for $i \in [q_p]$ and forwards the response to $\mathbf{A}_{1,2}$. For each construction query of $\mathbf{A}_{1,2}$, (N^j, M^j) with $(M_1^j,\ldots,M_{m^j}^j) \xleftarrow{2n} M^j$ for $j \in [q_c]$, $\mathbf{A}_u$ does the following:

1. $\mathbf{A}_u$ samples $k \twoheadleftarrow [m^j]$, $K_1^j \twoheadleftarrow \mathbb{F}_2^n$, $T_1^j \twoheadleftarrow \mathbb{F}_2^n$. Moreover, it initializes an empty list of leakages $\mathbf{L}$.

2. For $i = 1..k-1$, $\mathbf{A}_u$ queries its primitive oracle $\widetilde{\pi}$ with (K_i^j, i, T_i^j, N^j) to obtain $(L_i^j, K_{i+1}^j, T_{i+1}^j, Y_i^j)$, computes $C_i^j \leftarrow Y_i^j \oplus M_i^j$, and adds the leakage L_i^j to the list $\mathbf{L}^j$.
3. $\mathbf{A}_u$ queries its primitive oracle $\widetilde{\pi}$ with (K_k^j, k, T_k^j, N^j) to obtain $(L_i^j, K_{k+1}^j, T_{k+1}^j, Y_k^j)$. Then, $\mathbf{A}_u$ queries its $\mathsf{LUP\text{-}4}$ challenger with (K_k^j, k, T_k^j, N^j) to obtain $(L_k^j, K_{k+2}^j, T_{k+2}^j, Y_{k+1}^j)$ from the response, computes $C_{k+1}^j \leftarrow Y_{k+1}^j \oplus M_{k+1}^j$, and adds the leakage to the list $\mathbf{L}^j$.
4. For $i = k+2..m^j$, $\mathbf{A}_u$ queries primitive oracle with (K_i^j, i, T_i^j, N^j) to obtain $(L_i^j, K_{i+1}^j, T_{i+1}^j, Y_i^j)$ and computes $C_i^j \leftarrow Y_i^j \oplus M_i^j$, and adds the leakage to the list $\mathbf{L}^j$.
5. $\mathbf{A}_u$ returns $(\mathbf{L}^j, C^j)$ with $C^j = (C_1^j \parallel \ldots \parallel C_{m^j}^j)$ to $\mathbf{A}_{1,2}$.

At the end of the interaction, $\mathbf{A}_u$ outputs a set of guesses for the used tweakeys $\mathcal{K}$ that consists of all tweakeys in the primitive queries of $\mathbf{A}_{1,2}$, $\mathcal{K} = \{(K, S) : (K, S, X, Y) \in \tau_p\}$ for any X, $Y \in \mathbb{F}_2^n$.

The advantage of the distinguisher $\mathbf{A}_{1,2}$ is inherited by $\mathbf{A}_u$ who can measure its leakage of the $\mathsf{LUP\text{-}4}$ query p times. The difference consists of the σ indices over the guess of k. Hence, the advantage between the games is upper bounded by

$$\Delta\mathcal{G}_{1,2} \leq \sigma \cdot \mathbf{Adv}^{\mathsf{LUP\text{-}4}}_{F[\widetilde{\pi}]}(p, \sigma + q_c + q_p) + \frac{q_p \cdot 2(\sigma + 2q_c) + 2(\sigma + 2q_c)^2}{2^{2n}}\,.$$

Difference $\Delta\mathcal{G}_{2,3}$. Since the $\mathsf{IdealPRG}$ outputs a random keystream, in the black-box setting, its outputs would be indistinguishable from random. Though, under leakage, there is a remaining advantage that we can reduce to that of an $\mathsf{XOR\$}$ adversary. We follow the stepwise approach by Berti et al. of many hybrid adversaries. We define an adversary $\mathbf{A}_x$ that simulates a distinguisher $\mathbf{A}_{2,3}$ whose aim is to distinguish between Games $\mathcal{G}_2$ and $\mathcal{G}_3$. Whenever $\mathbf{A}_{2,3}$ asks a primitive query to $\widetilde{\pi}^{\pm}$, $\mathbf{A}_x$ answers it with its own primitive oracle $\widetilde{\pi}^{\pm}$, forwards the responds to $\mathbf{A}_{2,3}$, and stores the query and response into a transcript τ_p. For each construction query of $\mathbf{A}_{2,3}$, (N^j, M^j) with $(M_1^j, \ldots, M_{m^j}^j) \xleftarrow{2n} M^j$ for $j \in [q_c]$ and some fixed index $k \in [m^j]$, $\mathbf{A}_x$ does the following:

1. $\mathbf{A}_x$ samples $K_1 \twoheadleftarrow \mathbb{F}_2^n$, $T_1 \twoheadleftarrow \mathbb{F}_2^n$. Moreover, it initializes an empty list of leakages $\mathbf{L}$.
2. For $i = 1..k-1$, $\mathbf{A}_x$ samples unifomly random values K_{i+1}^j, T_{i+1}^j, Y_i^j, computes $C_i^j \leftarrow Y_i^j \oplus M_i^j$ and leakage L_i^j as $\mathsf{IdealPRG}$ and stores L_i^j into the list $\mathbf{L}^j$.
3. $\mathbf{A}_x$ samples K_k^j and T_k^j. For $c \in \{1, 2\}$, it builds $S_{k,c}^j = (c+1, k, T_k^j)$, submits it together with $M_{k,c}^j$ to its $\mathsf{XOR\$}$ challenger to obtain $(L_k^j, C_{k,c}^j)$. It computes the would-be leakage $L_{k,c}^j$. It builds $C_k^j = C_{k,1}^j \parallel C_{k,1}^j$ and $L_k^j = L_{k,1}^j \parallel L_{k,2}^j$ and adds it to the list $\mathbf{L}^j$.
4. For $i = k+1..m^j$, $\mathbf{A}_x$ proceeds as for $i = 1..k-1$.
5. $\mathbf{A}_x$ returns $(\mathbf{L}^j, C^j)$ with $C^j = (C_1^j \parallel \ldots \parallel C_m^j)$ to $\mathbf{A}_{2,3}$.

6. $\mathbf{A}_x$ forwards the guess of b' from $\mathbf{A}_{2,3}$ to its own challenger.

We see that the advantage of $\mathbf{A}_x$ with p leakage trails each is inherited from $\mathbf{A}_{2,3}$. We have to build σ such adversaries $\mathbf{A}_x$ over all queries and blocks that $\mathbf{A}_{2,3}$ queries to fully reduce it. Over all blocks, we obtain that

$$\Delta\mathcal{G}_{2,3} \leq \sigma \cdot \mathbf{Adv}^{\mathsf{XOR\$}}_{F[\tilde{\pi}]}(2\sigma + 2q_c + q_p)\,.$$

Our claim in Lemma 1 follows from summing up the individual bounds. □

E Blackbox Security of NHaT with Keyed Hashing

For the sake of completeness, we provide also a black-box security bound of NHaT in the following.

E.1 Black-box Model: MAC Security

A message authentication code (MAC) Π consists of two algorithms $\textsc{Auth} : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ and $\textsc{Verify} : \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{\bot, \top\}$ for authentication and verification, respectively. The canonical MAC defines the verification function with $\textsc{Auth}_K$ as

$$\textsc{Verify}_K(M,T) = \begin{cases} \top & \text{iff } \textsc{Auth}_K(M) = T \\ \bot & \text{otherwise.} \end{cases}.$$

Definition 6 (MAC Security). Let $\Pi = (\mathcal{T}, \mathcal{V})$ be a MAC and $K \twoheadleftarrow \mathcal{K}$. Then, the MAC advantage of an adversary $\mathbf{A}$ on F is defined as $\mathbf{Adv}^{\mathsf{MAC}}_{\mathcal{T}_K,\mathcal{V}_K}(\mathbf{A}) \stackrel{\text{def}}{=} \Delta_{\mathbf{A}}(\mathcal{T}_K, \mathcal{V}_K\,;\, \mathcal{T}_K, \bot)$, where $O_1 \not\rightarrow O_2$.

Note that this is equivalent that $\mathbf{A}$ forges, where to forge means that an adversary $\mathbf{A}$ finds a valid message-tag tuple (M,T) that it has not queried to the authentication oracle before. Valid means that $\textsc{Verify}_K(M,T) = \top$. The similar notions nMAC and sdMAC differ only in the domains of the MAC in the sense that the former takes a nonce input whereas the latter is stateless deterministic.

Lemma 7 (Lemma 3 in [15]). Let $\mathcal{S} = \{S_1, \ldots, S_r\}$, $\lambda_=$ be a list of permutation equalities and $\lambda_{\neq}$ be a list of permutation inequalities compatible with $\lambda_=$. Let $q \stackrel{\text{def}}{=} |\lambda_=|$ and $q' \stackrel{\text{def}}{=} |\lambda_{\neq}|$. Assume $q, q' < 2^n$. For $i = 1..r$, let q_i be the number of $(S,X,Y) \in \lambda_=$ such that $S_i = S$. Then, for $(\pi_s) \twoheadleftarrow \mathsf{Perm}(\pi)^s$

$$\Pr\left[(\pi_s) \in \mathsf{Comp}(\lambda)\right] \geq \frac{1}{\prod_{i=1}^{r} (2^n)_{q_i}} \left(1 - \frac{q'}{2^n - \max\{q_i, \ldots, q_r\}}\right).$$

E.2 nMAC Security of NHaT

Theorem 3 (nMAC Security of NHaT). Let $\widetilde{E} \in \mathsf{TBC}(\mathcal{K}, \mathbb{F}_2^n \times \mathcal{N}, \mathbb{F}_2^n)$. Let $H : \mathcal{K}_H \times \mathcal{M} \to \mathbb{F}_2^n$ and $H : \mathcal{K}'_H \times \mathcal{M} \to \mathcal{Y}$ be ϵ-AU. Further, let $K \twoheadleftarrow \mathcal{K}$, $K_H \twoheadleftarrow \mathcal{K}_H$, $K'_H \twoheadleftarrow \mathcal{K}'_H$ be independent keys. Let q_m and q_v be integers for the number of MAC and verification queries of $\mathbf{A}$ such that $q_m < 2^n$ and μ be the number of nonce-repeating queries. Then, $\mathbf{Adv}^{\mathsf{nMAC}}_{\mathsf{HaT}[\widetilde{E}_K,H]}(q_m, \mu, q_v)$ is at most

$$\mathbf{Adv}^{\mathsf{TPRP}}_{\widetilde{E}_K}(q_m + q_v) + 2(\mu - 1)q_m\epsilon^2 + \mu q_v\epsilon^2 + \frac{q_m q_v \epsilon^2}{2^n} + \frac{q_v}{2^n - \mu}\,.$$

One can see from Theorem 3 that a nonce-respecting adversary can only guess the correct authentication tag in verification queries, the number of which can be limited by a surrounding protocol. This bound is considerably better as the $q^2\epsilon^2$ security of HaT. If nonces repeat often, the security of NaT degrades to the birthday bound, whereas that of HaT and NHaT remain at n bit for optimal hash functions.

Proof of Theorem 3. The first step is to replace $\widetilde{E}_K$ with a permutation $\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}(\mathcal{T}_D, \mathbb{F}_2^n)$ that cannot be queried by $\mathbf{A}$. The advantage to distinguish between both settings is upper bounded by

$$\mathbf{Adv}^{\mathsf{TPRP}}_{\widetilde{E}_K}(q_m + q_v)\,.$$

Next, we follow the H-coefficient technique. Let $\tau = (\tau_m, \tau_v, K_H, K'_H)$ denote the transcript of $\mathbf{A}$ where

$$\tau_m = ((M_1, N_1, T_1), \ldots, (M_{q_m}, N_{q_m}, T_{q_m})),$$
$$\tau_v = ((M'_1, N'_1, T'_1), \ldots, (M'_{q_v}, N'_{q_v}, T'_{q_v}))$$

denote MAC and verification queries, respectively. Let Θ_{real} and Θ_{ideal} represent random variables for the transcripts in the real and ideal world, respectively. We define a variable bad that is set if any of the following bad events occurs:

- bad_1: There exist distinct MAC queries (M_i, N_i, T_i) and (M_j, N_j, T_j) such that $(X_i, Y_i, N_i), (X_j, Y_j, N_i)$.
- bad_2: There exist distinct MAC queries (M_i, N_i, T_i) and (M_j, N_j, T_j) such that $(T_i, Y_i, N_i), (T_j, Y_j, N_i)$.
- bad_3: There exist a MAC query (M_i, N_i, T_i) and a verification query (M_j, N_j, T_j) such that $(X_i, Y_i, N_i, T_i) = (X_j, Y_j, N_j, T_j)$.

We upper bound $\Pr[\mathsf{bad}] \leq \sum_{i=1}^{3} \Pr[\mathsf{bad}_i]$.

We define a transcript as bad if $\mathsf{bad} = \mathsf{true}$ and good otherwise. Then, the bound in Theorem 3 follows from the fundamental lemma of the H-coefficient technique and the application of Lemma 8 and 9. □

Lemma 8. For any integers q_m, q_v, μ, it holds that

$$\Pr[\mathsf{bad}] \leq 2(\mu - 1)q_m\epsilon^2 + \mu q_v\epsilon^2 + \frac{q_m q_v \epsilon^2}{2^n}\,.$$

Proof. In the following, we upper bound the probability of the individual bad events.

bad_1. In this case, it follows from the ϵ-almost universality and independence of H and H' that.

$$\Pr[X_i = X_j, Y_i = Y_j] \leq \epsilon^2 .$$

Since we need $N_i = N_j$, at least one of the queries must be nonce-repeating. Over all query pairs, we have $(\mu - 1) \cdot q_m$ pais. Therefore, it holds that

$$\Pr[\mathsf{bad}_1] \leq (\mu - 1) q_m \epsilon^2 .$$

bad_2. In this case, it follows from the ϵ-almost universality of H' and the uniform random choice of T_i and T_j in the ideal world that

$$\Pr[Y_i = Y_j, T_i = T_j] \leq \epsilon \cdot 2^{-n} .$$

Since we need $N_i = N_j$, at least one of the queries must be nonce-repeating. Over all query pairs, we have $(\mu - 1) \cdot q_m$ pais. Therefore, it holds that

$$\Pr[\mathsf{bad}_2] \leq (\mu - 1) q_m \cdot \epsilon \cdot 2^{-n} \leq (\mu - 1) q_m \cdot \epsilon^2 .$$

bad_3. In this case, we need a collision in both hash outputs. Since $(N_i, T_i) = (N_j, T_j)$, it must follow that $M_i \neq M_j$; otherwise, the adversary is trivial. From the ϵ-almost universality and independence of H and H', it follows again that

$$\Pr[X_i = X_j, Y_i = Y_j] \leq \epsilon^2 .$$

We have to consider two cases. Say the verification query comes before the MAC query. Then, the probability that both tags are identical in the ideal world is

$$\Pr[T_i = T_j] = \frac{1}{2^n} .$$

Then, the probability for (X_i, Y_i, T_i) is at most

$$\frac{q_m q_v \epsilon^2}{2^n} .$$

In the opposite case when the verification query follows the MAC query, $\mathbf{A}$ can simply choose $(N_j, T_j) = (N_i, T_i)$. However, it can address at most μ queries at once with a verification query. Hence, there are at most μq_v query combinations. Then, the probability that $(X_i, Y_i) = (X_j, Y_j)$ is at most

$$\mu q_v \epsilon^2 .$$

It follows that

$$\Pr[\mathsf{bad}_3] \leq \mu q_v \epsilon^2 + \frac{q_m q_v \epsilon^2}{2^n} . \quad \square$$

Lemma 9. For any good transcript τ, it holds that

$$\frac{\Pr[\Theta_{\text{real}} = \tau]}{\Pr[\Theta_{\text{ideal}} = \tau]} \leq 1 - \frac{q_v}{2^n - \mu} .$$

Proof. Let $\mathcal{T} = \{Y_i, \ldots, Y_{q_m}\}$ be the set of all tweaks in MAC queries and reorder them to eliminate duplicates as $\mathcal{T} = \{W_i, \ldots, W_r\}$ such that $r \leq q_m$ is the number of distinct tweaks used in MAC queries. Let ℓ_i denote the number of MAC queries (M_i, N_i, T_i) with nonce N_i and $H'_{K'_H}(M_i) = Y_i$. The probability of a good transcript τ in the ideal world is simply

$$\Pr\left[\Theta_{\text{ideal}} = \tau\right] = \frac{1}{|\mathcal{K}_H| \cdot |\mathcal{K}'_H| \cdot (2^n)^{q_m}}\,.$$

In the real world, a tweakable permutation $\widetilde{\pi}$ is compatible with τ if

$$\widetilde{\pi}^{Y_i,N_i}(X_i) = T_i, \text{ for all } i \in [q_m]\,,$$
$$\widetilde{\pi}^{Y_i,N_i}(X_i) \neq T_i, \text{ for all } i \in [q_v]\,.$$

Let $\mathsf{Comp}(\tau_m)$, $\mathsf{Comp}(\tau_v)$, and $\mathsf{Comp}(\tau)$ be the set of tweakable permutaitons compatible to τ_m, τ_v, and τ, respectively. Then, it holds that

$$\Pr\left[\Theta_{\text{real}} = \tau\right] = \frac{1}{|\mathcal{K}_H| \cdot |\mathcal{K}'_H|} \cdot \Pr\left[\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}(\mathcal{T}_D, \mathbb{F}_2^n) : \widetilde{\pi} \in \mathsf{Comp}(\tau)\right]\,.$$

We obtain that

$$\Pr\left[\widetilde{\pi} \in \mathsf{Comp}(\tau)\right] = \frac{1}{\prod_{i=1}^{r} (2^n)_{\ell_i}} \cdot \left(1 - \frac{q_v}{2^n - \mu}\right)\,.$$

We obtain that

$$\frac{\Pr\left[\Theta_{\text{real}} = \tau\right]}{\Pr\left[\Theta_{\text{ideal}} = \tau\right]} \geq \left(1 - \frac{q_v}{2^n - \mu}\right) \cdot \prod_{i=1}^{r} \frac{(2^n)^{\ell_i}}{(2^n)_{\ell_i}} \geq 1 - \frac{q_v}{2^n - \mu}\,. \quad \square$$

F Comparison of Alternative Encryption Modes

F.1 Requirements

Our aim for a more secure mode of operation was to obtain n-bit security under leakage in the context of use in an nAE scheme. We assume an ephemeral-key scheme with (1) n-bit $\mathsf{CPAmL2}$ security under nonce-respecting adversaries and (2) unpredictability of the iteration in $O(\sigma/2^{2n})$ to allow the use of a hybrid argument in the $\mathsf{CCAmL2}$ analysis. For this purpose, we employ a $2n$-bit tweakey and assume a TWEAKEY-based primitive that treats both n-bit tweakey words similarly as secrets.

Two desiderata seem to complement each other: inverse-freeness vs. XOR-freeness. In the black-box setting, keystream generators such as counter mode are preferable since the keystream can easily be added to the message, which equalize the operations for en- and decryption and spares the primitive's inverse. Under leakage, the XOR may leak information about the keystream.

Algorithm 7 Definition of alternative encryption schemes.

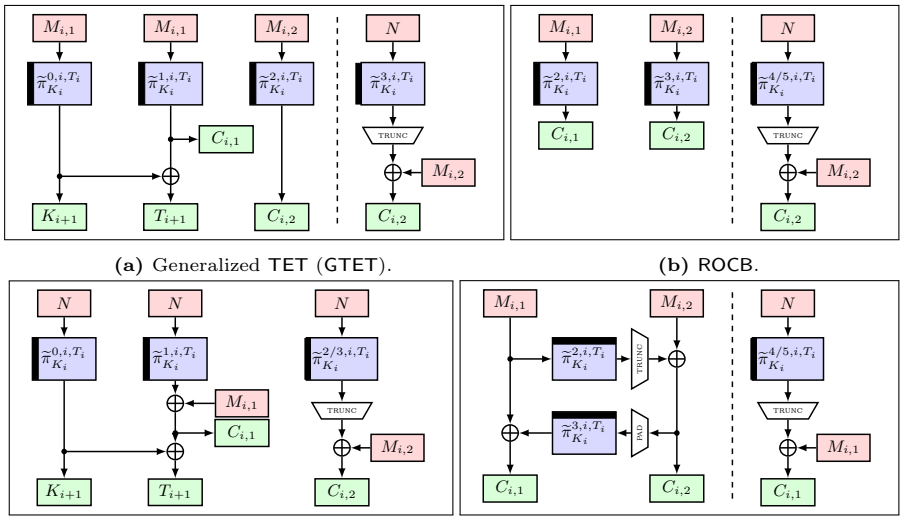
function $\mathrm{GTET}[\widetilde{\pi}, r]^N_{\mathrm{K}_0,T_0}(M)$
  $(M_1, \ldots, M_m) \xleftarrow{n \cdot r} M$
  for $i \leftarrow 1..m-1$ **do**
    $(M_{i,1}, \ldots, M_{i,r}) \xleftarrow{n} M_i$
    $K_{i+1} \leftarrow \widetilde{\pi}^{0,i,T_i}_{K_i}(M_{i,1})$
    for $j \leftarrow 1..r$ **do**
      $C_{i,j} \leftarrow \widetilde{\pi}^{j,i,T_i}_{K_i}(M_{i,j})$
    $T_{i+1} \leftarrow C_{i,j} \oplus K_{i+1}$
    $C_i \leftarrow (C_{i,1}, \ldots, C_{i,r})$
  $(M_{m,1}, \ldots, M_{m,r_m}) \xleftarrow{n} M_m$
  for $j \leftarrow 1..r_m - 1$ **do**
    $C_{m,j} \leftarrow \widetilde{\pi}^{j,m,T_m}_{K_m}(M_{m,j})$
  if $|M_{m,r_m}| = n$ **then**
    $C_{m,r_m} \leftarrow \widetilde{\pi}^{r_m,m,T_m}_{K_m}(M_{m,r_m})$
  else
    $S_{m,r_m} \leftarrow \widetilde{\pi}^{r+1,m,T_m}_{K_m}(N)$
    $S'_{m,r_m} \leftarrow \mathrm{TRUNC}_{|M_{m,r_m}|}(S_{m,r_m})$
    $C_{m,r_m} \leftarrow S'_{m,r_m} \oplus M_{m,r_m}$
  $C_m \leftarrow (C_{m,1}, \ldots, C_{m,r_m})$
  return $(C_1, \ldots, C_m)$

function $\mathrm{ROTR}[\widetilde{\pi}, r]^N_{\mathrm{K}_0,T_0}(M)$
  $(M_1, \ldots, M_m) \xleftarrow{n \cdot r} M$
  for $i \leftarrow 1..m$ **do**
    $(M_{i,1}, \ldots, M_{i,r_m}) \xleftarrow{n} M_i$
    $K_{i+1} \leftarrow \widetilde{\pi}^{0,i,T_i}_{K_i}(N)$
    $T_{i+1} \leftarrow \widetilde{\pi}^{1,i,T_i}_{K_i}(N)$
    for $j \leftarrow 1..r_m/2$ **do**
      $S_1 \leftarrow \widetilde{\pi}^{2j,i,T_i}_{K_i}(M_{i,2j-1})$
      $C_{i,2j} \leftarrow \mathrm{TRUNC}_{|M_{i,2j}|}(S_1) \oplus M_{i,2j}$
      $S_2 \leftarrow \widetilde{\pi}^{2j+1,i,T_i}_{K_i}(\mathrm{PAD}_n(C_{i,2j}))$
      $C_{i,2j} \leftarrow S_2 \oplus M_{i,2j-1}$
    if $r_m \bmod 2 = 1$ **then**
      $S_{i,r_m} \leftarrow \widetilde{\pi}^{r+2,i,T_i}_{K_i}(N)$
      $S'_{m,r_m} \leftarrow \mathrm{TRUNC}_{|M_{m,r_m}|}(S_{m,r_m})$
      $C_{i,r_m} \leftarrow S'_{i,r_m} \oplus M_{i,r_m}$
    $C_i \leftarrow (C_{i,1}, \ldots, C_{i,r_m})$
  return $(C_1, \ldots, C_m)$

function $\mathrm{GFBE}[\widetilde{\pi}, r]^N_{\mathrm{K}_0,T_0}(M)$
  $(M_1, \ldots, M_m) \xleftarrow{n \cdot r} M$
  for $i \leftarrow 1..m$ **do**
    $(M_{i,1}, \ldots, M_{i,r_m}) \xleftarrow{n} M_i$
    $K_{i+1} \leftarrow \widetilde{\pi}^{0,i,T_i}_{K_i}(N)$
    for $j \leftarrow 1..r_m$ **do**
      $C_{i,j} \leftarrow \widetilde{\pi}^{j,i,T_i}_{K_i}(N) \oplus M_{i,j}$
    $T_{i+1} \leftarrow C_{i,j} \oplus K_{i+1}$
    $D \leftarrow r_m$
    if $|M_{i,r_m}| < n$ **then** $D \leftarrow r+1$
    $S_{i,r_m} \leftarrow \widetilde{\pi}^{D,r_m,T_i}_{K_i}(N)$
    $S'_{i,r_m} \leftarrow \mathrm{TRUNC}_{|M_{m,r_m}|}(S_{i,r_m})$
    $C_{i,r_m} \leftarrow S'_{i,r_m} \oplus M_{i,r_m}$
    $C_i \leftarrow (C_{i,1}, \ldots, C_{i,r_m})$
  return $(C_1, \ldots, C_m)$

function $\mathrm{ROCB}[\widetilde{\pi}, r]^N_{\mathrm{K}_0,T_0}(M)$
  $(M_1, \ldots, M_m) \xleftarrow{n \cdot r} M$
  for $i \leftarrow 1..m$ **do**
    $(M_{i,1}, \ldots, M_{i,r_m}) \xleftarrow{n} M_i$
    $K_{i+1} \leftarrow \widetilde{\pi}^{0,i,T_i}_{K_i}(N)$
    $T_{i+1} \leftarrow \widetilde{\pi}^{1,i,T_i}_{K_i}(N)$
    for $j \leftarrow 1..r_m - 1$ **do**
      $C_{i,j} \leftarrow \widetilde{\pi}^{j+1,i,T_i}_{K_i}(M_{i,j})$
    if $|M_{i,r_m}| < n$ **then**
      $C_{i,r_m} \leftarrow \widetilde{\pi}^{r_m,i,T_i}_{K_i}(M_{i,r_m})$
    else
      $S_{i,r_m} \leftarrow \widetilde{\pi}^{r+1,i,T_i}_{K_i}(N)$
      $S'_{i,r_m} \leftarrow \mathrm{TRUNC}_{|M_{i,r_m}|}(S_{i,r_m})$
      $C_{i,r_m} \leftarrow S'_{i,r_m} \oplus M_{i,r_m}$
    $C_i \leftarrow C_{i,1}, \ldots, C_{i,r_m}$
  return $(C_1, \ldots, C_m)$

function $\mathrm{RCTR}[\widetilde{\pi}, r]^N_{\mathrm{K}_0,T_0}(M)$
  $(M_1, \ldots, M_m) \xleftarrow{n \cdot r} M$
  for $i \leftarrow 1..m$ **do**
    $(M_{i,1}, \ldots, M_{i,r_m}) \xleftarrow{n} M_i$
    $K_{i+1} \leftarrow \widetilde{\pi}^{0,i,T_i}_{K_i}(N)$
    $T_{i+1} \leftarrow \widetilde{\pi}^{1,i,T_i}_{K_i}(N)$
    for $j \leftarrow 1..r_m$ **do**
      $D \leftarrow j+1$
      if $|M_{i,j}| < n$ **then** $D \leftarrow r+1$
      $S_{i,j} \leftarrow \widetilde{\pi}^{D,i,T_i}_{K_i}(N)$
      $S'_{i,r_m} \leftarrow \mathrm{TRUNC}_{|M_{i,j}|}(S_{i,j})$
      $C_{i,j} \leftarrow S'_{i,j} \oplus M_{i,j}$
    $C_i \leftarrow (C_{i,1}, \ldots, C_{i,r_m})$
  return $(C_1, \ldots, C_m)$



(a) Generalized TET (GTET). **(b)** ROCB.

(c) Generalized FEMALE (GFBE). **(d)** ROTR.

Fig. 6: One iteration of alternative encryption modes for $r = 2$ message blocks. The right side shows the treatment of the final block if its size is smaller than bits. Since ROCB and ROTR derive the subsequent key and tweak as RCTR, only the message processing components are shown.

F.2 Considered Modes

In total, we considered five options to increase the security under leakage:

(1) Generalized TET [9] (GTET),
(2) Generalized FEMALE [29] (GFBE),
(3) Rekeying counter mode (RCTR),
(4) Rekeying OCB (ROCB), and
(5) Rekeying OTR (ROTR).

Their encryption definitions are given in Algorithm 7. Except for RCTR, they are illustrated in Figure 6 and compared in Table 4.

GTET and GFBE generalize the encryption schemes used in TET and FEMALE. The latter three modes represent ephemeral-key adaptations of the encryption in the well-known modes counter, OCB and OTR. We generalize the scheme definitions by a flexible number of primitive calls and adding a treatment of a final partial message block, where the nonce is encrypted and the truncated result is added to the message block, to their definitions. We note that FEMALE encrypts a message in two passes plus a hash over the ciphertext; GFBE corresponds to the feedback-based first level of encryption.

Comparison. GTET and GFBE possess the advantage that their rate is $r/(r+1)$ for r message blocks, i.e., they need only a single call more per iteration as a

Table 4: Comparison of potential encryption modes for a primitive with n-bit key and state. XORF = XOR-free, IF = inverse-free. Partial XOR-free means only XORs for partial final blocks. • = yes, ∘ = XORs only for partial final block, – = absent or trivial, r = blocks processed per iteration.

		Features		Security in bits		
Scheme	Rate	XORF	IF	CPA	CCA	CPAmL1
GTET	$r/(r+1)$	∘	–	n	$n/2$	n
GFBE	$r/(r+1)$	–	•	n	$n/2$	n
RCTR	$r/(r+2)$	–	•	n	–	n
ROCB	$r/(r+2)$	∘	–	n	–	n
ROTR	$r/(r+2)$	–	•	n	–	n

secret state, equivalent to the capacity in permutation-based schemes. Given the need for a $2n$-bit tweakey, they use at least one of the further random outputs for ciphertext generation and – hidden by adding the secret part – as input to the subsequent iteration. Note that chosen-ciphertext security is preserved when they are embedded into an nAE scheme that refuses to decrypt after the tag has been deemed invalid.

G Auxiliary Notions for GTET

Prior to the analysis of GTET, we have to tailor the unpredictability-under-leakage notion (LUP-2 in TEDT and LUP-4 in TEDT2 with RCTR) to an iteration of GTET.

G.1 Unpredictability for GTET

It takes a larger tweakey $K_0, T_0 \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ from the adversary, samples $K_1 \twoheadleftarrow \mathbb{F}_2^n$, $T_1 \twoheadleftarrow \mathbb{F}_2^n$, and uses it for p decryptions, as well as p calls of one iteration of the PRG that generates two n-bit outputs and K_2, T_1, $C_{1,1}$, …, $C_{1,r}$ for a given message M. $\mathbf{A}$ can query the encryption p times to collect a vector of in- and output leakages from all primitive calls except for the calls to $M_{0,1}$ and $M_{0,2}$, where it is not provided with input leakage. $\mathbf{A}$ outputs a set $\mathcal{K}'$ of q values K_1 and wins iff the correct tweakey is contained.

Definition 7 (GTET-r-LUP). Let $\widetilde{\pi} \in \mathsf{TBC}(\mathbb{F}_2^n, \mathcal{T}_D, \mathbb{F}_2^n)$. Let $\mathcal{L}^{\mathsf{in}}$ and $\mathcal{L}^{\mathsf{out}}$ be sets of leakage functions. Let $\mathbf{A}$ be an adversary that provides $K_0, T_0 \in \mathbb{F}_2^n$ to and plays the GTET-r-LUP experiment against $\widehat{\mathcal{E}}[\widetilde{\pi}]$, and outputs a set $\mathcal{K}' \subseteq (\mathbb{F}_2^n)^{\leq q}$. The GTET-$r$-LUP advantage of $\mathbf{A}$ is defined as

$$\mathbf{Adv}^{\mathsf{GTET}\text{-}r\text{-}\mathsf{LUP}}_{\widehat{\mathcal{E}}[\widetilde{\pi}]^{*,*,T_0}_{K_0},\mathcal{L}^{\mathsf{in}},\mathcal{L}^{\mathsf{out}}}(\mathbf{A}) \stackrel{\text{def}}{=} \Pr\left[K_1 \in \mathcal{K}'\right] .$$

We can already envision the birthday-bound consequences of the so-modified notion. The release of T_1 seems necessary to model the knowledge of $C_{i-1,1}$ in

Algorithm 8 GTET-r-LUP experiment.

```
11: procedure INITIALIZE(K_0, T_0)
12:     K_1 ←$ F_2^n
13:     T_1 ←$ F_2^n
14:     M_{0,1} ← (Ẽ^{0,0,T_0}_{K_0})^{-1}(K_1)
15:     M_{0,2} ← (Ẽ^{1,0,T_0}_{K_0})^{-1}(T_1)
21: function LEAK[Ẽ](Λ^in, Λ^out, K, T, X, Y)
22:     R^in, R^out ←$ R
23:     L^in ← Λ^in(K, T, X; R^in)
24:     L^out ← Λ^out(K, T, Y; R^out)
25:     return (L^in, L^out)
31: function FINALIZE(K')
32:     win ← |K'| ≤ q ∧
33:           (K_1, T_1) ∈ K'
34:     if win then return 1
35:     return 0
41: function Ê[Ẽ](M, Λ^in, Λ^out)
42:     (M_1, ..., M_r) ←$ M
43:     for i ← 1..p do
44:         R^out_{0,1}, R^out_{0,2} ←$ R
45:         K_1 ← Ẽ^{0,0,T_0}_{K_0}(M_{0,1})
46:         T_1 ← Ẽ^{1,0,T_0}_{K_0}(M_{0,2})
47:         K_2 ← Ẽ^{0,1,T_1}_{K_1}(M_1)
48:         L^out_{0,1} ← Λ^out_0(K_0, (0,0,T_0), K_1; R^out_{0,1})
49:         L^out_{0,2} ← Λ^out_0(K_0, (1,0,T_0), T_1; R^out_{0,2})
50:         L_0 ← L^out_{0,1}, L^out_{0,2}
51:         L^in_{1,1}, L^out_{1,1} ← LEAK[Ẽ](Λ^in, Λ^out, K_1, (0,1,T_1), M, K_2)
52:         L_{1,1} ← L^in_{1,1}, L^out_{1,1}
53:         for j ← 1..r do
54:             C_{1,j} ← Ẽ^{j,1,T_1}_{K_1}(M_j)
55:             L^in_{j,2}, L^out_{j,2} ← LEAK[Ẽ](Λ^in, Λ^out,
56:                 K_1, (j,1,T_1), M, C_{1,j})
57:             L_{1,j+1} ← L^in_{1,j+1}, L^out_{1,j+1}
58:     C_1 ← (C_{1,1}, ..., C_{1,r})
59:     return (K_2, C_1, [L_0]_p, [L_{1,1}]_p, [L_{1,2}]_p, ..., [L_{1,r+1}]_p)
```

GTET. It is not a fully exact representation since T_i would be unknown, but $C_{i-1,1} = T_i \oplus K_i$ would be known in GTET instead. Though, the knowledge of the output would allow the adversary to fully compute the decrypted block to $M_{0,2}$ in the GTET-r-LUP game.

As a consequence, the secret state consists of only n bits. Under leakage, the term of GTET-r-LUP security, given ρ chunks, whose goal is to recover an n-bit state is upper bounded at best by

$$\rho \cdot \mathbf{Adv}^{\mathsf{GTET}\text{-}r\text{-}\mathsf{LUP}}_{\mathsf{GTET}[\widetilde{\pi},r]}(\mathbf{A}) \leq \rho \cdot O\left(\frac{q_c + q_p}{c \cdot 2^n}\right).$$

While the term is an artifact in the black-box setting, it is tight under leakage. Thus, it seems hard to employ a variant of TET encryption for higher security under leakage. Though, the rate of TEDT could be improved by employing TET instead of the Bellare-Yee PRG for encryption. Nevertheless, we conduct the proof in detail to give a the bound for a higher-rate scheme.

We define $\mathbf{Adv}^{\mathsf{GTET}\text{-}r\text{-}\mathsf{LUP}}_{\widehat{\mathcal{E}}[\widetilde{\pi}],\mathcal{L}^{\mathsf{in}},\mathcal{L}^{\mathsf{out}}}(p,q)$ as the maximum of all GTET-r-LUP adversaries $\mathbf{A}$ on $\widehat{\mathcal{E}}[\widetilde{\pi}]$ that ask at most p queries and output a set of at most q guesses.

G.2 Indistinguishability of XOR with Two Unknowns

In theory, we can consider another formalism. GTET contains an XOR whose leakage may destroy privacy. In contast to RCTR, the XOR of a known output $T^j_{i+1}C^j_{i,1} \oplus K^j_{i+1}$ contains two unknowns. Thus, the leakage must leak information about both unknowns to allow an adversary to exploit it. We capture it for the sake of comprehension. The XOR\$2 game in Algorithm 9 is our real-or-random game, where the real world processes a message $M \in \mathbb{F}_2^n$ chosen by the adversary,

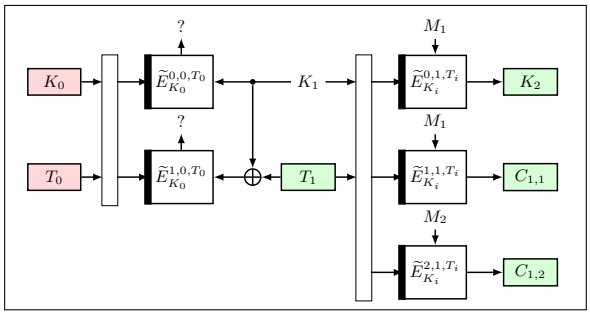


Fig. 7: The GTET-r-LUP setting for $r = 2$. White boxes only combine the tweakey.

Algorithm 9 XOR$2 experiment.

```
11: function INITIALIZE(K, T, M, Λ^out, Λ^⊕)
12:     Y ←$ F_2^n; b ←$ {0,1}
13:     M* ← M
14:     if b = 0 then
15:         M* ←$ F_2^n
16:     X ← (Ẽ_K^T)^{-1}(Y)
21: function FINALIZE(b')
22:     return b = b'
31: function Ê[Ẽ]_K(Λ^out, Λ^⊕)
32:     R^out, R^⊕ ←$ R
33:     Y ← Ẽ_K^T(X);  C ← Y ⊕ M*
34:     L^out ← Λ^out(K, T, Y; R^out)
35:     L^⊕ ← Λ^⊕(Y, C; R^⊕)
36:     for i ← 2..p do
37:         R^out, R^⊕ ←$ R
38:         Y ← Ẽ_K^T(X);  M* ← C ⊕ Y
39:         L^out ← Λ^out(K, T, Y; R^out)
40:         L^⊕ ← Λ^⊕(Y, M*; R^⊕)
41:     return ([L^out]_p, [L^⊕]_p)
```

and the ideal world samples and processes a message $M^* \xleftarrow{\$} \mathbb{F}_2^n$. Note that the result of the XOR is not returned to the adversary, but only the leakage.

Definition 8 (XOR$2). Let $\widetilde{\pi} \in \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$ and $K_1 \xleftarrow{\$} \mathcal{K}$. Let $\mathcal{L}^{\mathsf{out}}$, $\mathcal{L}_1^{\oplus}$, and $\mathcal{L}^{\oplus}$ be sets of leakage functions. Let $\mathbf{A}$ be a adversary that plays the XOR$2 experiment given in Algorithm 9 against $\widehat{\mathcal{E}}[\widetilde{\pi}]$. Then, the XOR$ advantage of $\mathbf{A}^b \Rightarrow b'$, interacting with world b and outputting b' is defined as

$$\mathbf{Adv}^{\mathsf{XOR\$2}}_{\widehat{\mathcal{E}}[\widetilde{\pi}]_K, \mathcal{L}^{\mathsf{out}}, \mathcal{L}^{\oplus}}(\mathbf{A}) \stackrel{\text{def}}{=} \left| \Pr\left[\mathbf{A}^1 \Rightarrow 1\right] - \Pr\left[\mathbf{A}^0 \Rightarrow 1\right] \right| .$$

We define $\mathbf{Adv}^{\mathsf{XOR\$2}}_{\widehat{\mathcal{E}}[\widetilde{\pi}]_K, \mathcal{L}^{\mathsf{out}}, \mathcal{L}^{\oplus}}(p, q)$ for the maximum advantage over all XOR$2 adversaries $\mathbf{A}$ on $\widehat{\mathcal{E}}[\widetilde{\pi}]_K$ that ask at most q queries under p measurements each.

H qCPA$mL2 Analysis of GTET$[\widetilde{\pi}, r]$

H.1 Reduction Analysis

Theorem 4. Let $\widetilde{\pi} \xleftarrow{\$} \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$. Let r and $n \geq 4$ be positive integers. Let $\mathbf{A}$ be an qCPA$mL2 adversary on $\mathit{\Pi}[\widetilde{\pi}]_K = \mathsf{TEDT2}[\widetilde{\pi}]_K$ that asks at most q_e encryption queries and q_d decryption queries of at most ρ chunks and at most $\sigma \leq 2^{n-3}$ blocks in total and $q_p \leq 2^{n-2}$ primitive queries. Let $F[\widetilde{\pi}, r]$ be an

Algorithm 10 Games $\mathcal{G}_1$ to $\mathcal{G}_3$ for the proof of $\mathsf{IdealEnc}[\widetilde{\pi}, r]$ replaces GTET [$\widetilde{\pi}$,r] in $\mathcal{G}_2$. Note that the primitive oracles $\widetilde{\boldsymbol{\pi}}^{\pm}$ and leakage functions belong to all games. The boxed statements belong only to Game $\mathcal{G}_3$ and $\mathcal{G}_4$.

procedure INITIALIZE(τ)
  $b \leftarrow \{0,1\}$; $K \twoheadleftarrow \mathcal{K}$; $\mathcal{Q} \leftarrow \emptyset$; bad $\leftarrow$ false

function REALPRG$[\widetilde{\pi}]^N(M)$
  $\mathbf{L} \leftarrow []$
  $\mathcal{Q} \xleftarrow{\cup} \{(K,(6,0,0)),(K,(7,0,0))\}$
  $K_1 \leftarrow \widetilde{\boldsymbol{\pi}}_K^{6,0,0}(N)$
  $T_1 \leftarrow \widetilde{\boldsymbol{\pi}}_K^{7,0,0}(N)$
  $(M_1,\ldots,M_m) \xleftarrow{2n} M$
  for $i \leftarrow 1..m$ **do**
    for $d \leftarrow 0..3$ **do** $\mathcal{Q} \xleftarrow{\cup} (K_i,(d,i,T_i))$
    $K_{i+1} \leftarrow \widetilde{\boldsymbol{\pi}}_{K_i}^{0,i,T_i}(N)$
    $T_{i+1} \leftarrow \widetilde{\boldsymbol{\pi}}_{K_i}^{1,i,T_i}(N)$
    $Y_i \leftarrow \widetilde{\boldsymbol{\pi}}_{K_i}^{2,T_i,i}(N) \,\|\, \widetilde{\boldsymbol{\pi}}_{K_i}^{3,T_i,i}(N)$
    $\boxed{M_i \twoheadleftarrow \mathbb{F}_2^{|M_i|}}$
    $C_i \leftarrow \text{TRUNC}_{|M_i|}(Y_i) \oplus M_i$
    $L_i \leftarrow ($
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (0,i,T_i), N, K_{i+1})$,
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (1,i,T_i), N, T_{i+1})$,
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (2,i,T_i), N, Y_{i,1})$,
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (3,i,T_i), N, Y_{i,2})$,
    LEAKXOR$(L^{\oplus}, M_{i,1}, Y_{i,1}, C_{i,1})$,
    LEAKXOR$(L^{\oplus}, M_{i,2}, Y_{i,2}, C_{i,2}))$
    $\mathbf{L} \xleftarrow{\cup} L_i$
  $C \leftarrow C_1 \,\|\, \cdots \,\|\, C_m$
  return $(\mathbf{L}, C)$

function $\widetilde{\boldsymbol{\pi}}_K^{d,i,T}(X)$
  if $(K,(d,i,T)) \in \mathcal{Q}$ **then** bad $\leftarrow$ true
  return $\widetilde{\pi}_K^{d,i,T}(X)$

function $(\widetilde{\boldsymbol{\pi}}_K^{d,i,T})^{-1}(Y)$
  if $(K,(d,i,T)) \in \mathcal{Q}$ **then** bad $\leftarrow$ true
  return $(\widetilde{\pi}_K^{d,i,T})^{-1}(Y)$

procedure INITIALIZE(τ)
  $b \leftarrow \{0,1\}$; $K \twoheadleftarrow \mathcal{K}$; $\mathcal{Q} \leftarrow \emptyset$; bad $\leftarrow$ false

function IDEALPRG$^N(M)$
  $\mathbf{L} \leftarrow []$
  $\mathcal{Q} \xleftarrow{\cup} \{(K,(6,0,0)),(K,(7,0,0))\}$
  $K_1 \twoheadleftarrow \mathbb{F}_2^n$
  $T_1 \twoheadleftarrow \mathbb{F}_2^n$
  $(M_1,\ldots,M_m) \xleftarrow{2n} M$
  for $i \leftarrow 1..m$ **do**
    for $d \leftarrow 0..3$ **do** $\mathcal{Q} \xleftarrow{\cup} (K_i,(d,i,T_i))$
    $K_{i+1} \twoheadleftarrow \mathbb{F}_2^n$
    $T_{i+1} \twoheadleftarrow \mathbb{F}_2^n$
    $Y_{i,1} \twoheadleftarrow \mathbb{F}_2^n$
    $Y_{i,2} \twoheadleftarrow \mathbb{F}_2^n$
    $\boxed{M_i \twoheadleftarrow \mathbb{F}_2^{|M_i|}}$
    $C_i \leftarrow \text{TRUNC}_{|M_i|}(Y_i) \oplus M_i$
    $L_i \leftarrow ($
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (0,i,T_i), N, K_{i+1})$,
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (1,i,T_i), N, T_{i+1})$,
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (2,i,T_i), N, Y_{i,1})$,
    LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K_i, (3,i,T_i), N, Y_{i,2})$,
    LEAKXOR$(L^{\oplus}, M_{i,1}, Y_{i,1}, C_{i,1})$,
    LEAKXOR$(L^{\oplus}, M_{i,2}, Y_{i,2}, C_{i,2}))$
    $\mathbf{L} \xleftarrow{\cup} L_i$
  $C \leftarrow C_1 \,\|\, \cdots \,\|\, C_m$
  return $(\mathbf{L}, C)$

function LEAK4UP$(L^{\mathsf{in}}, L^{\mathsf{out}}, K, T, X, Y)$
  return $[L^{\mathsf{in}}(K,T,X)]_p, [L^{\mathsf{out}}(K,T,Y)]_p$

function LEAKXOR$(L^{\oplus}, X, Y, Z)$
  return $[L^{\oplus}(X,Y)]_p, [L^{\mathsf{out}}(Y,Z)]_p$

iteration of $\mathsf{GTET}[\widetilde{\pi}, r]$. Let $\mathcal{L}_E$ and $\mathcal{L}_D$ be as defined at the top of Section 7. Then

$$\begin{aligned}\mathbf{Adv}^{\mathsf{qCPA\$mL2}}_{\Pi[\widetilde{\pi},r]_K}(\mathbf{A}) &\le \frac{2\sigma^2+(4+q_p)\sigma}{2^{2n}} + \Big(\frac{q}{2^n}\Big)^n + \frac{(n+1)\cdot q_p}{2^n} \\ &+ \rho\cdot\mathbf{Adv}^{\mathsf{GTET}\text{-}r\text{-}\mathsf{LUP}}_{F[\widetilde{\pi},r]}(\sigma+q_c+q_p) + q_c\cdot\mathbf{Adv}^{\mathsf{XOR\$}}_{F[\widetilde{\pi},r]}(2\sigma+2q_c+q_p) \\ &+ \rho\cdot\mathbf{Adv}^{\mathsf{XOR\$2}}_{F[\widetilde{\pi}]}(\rho+q_p)\,.\end{aligned}$$

Proof. Again, we employ a sequence of games $\mathcal{G}_1$ through $\mathcal{G}_4$. In that sequence, Game $\mathcal{G}_1$ represents the left-hand side of Equation (3),

$$\underset{\mathbf{A}}{\Delta}(\widehat{\mathcal{E}}_K, \widehat{\mathcal{E}}^{ch}_K, \widehat{\mathcal{D}}^{ch}_K, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D\,;\, \widehat{\mathcal{E}}_K, \widehat{\$}_{\mathcal{E}}, \widehat{\$}_{\mathcal{D}}, \widetilde{\pi}^{\pm}, \mathcal{L}_E, \mathcal{L}_D)\,,$$

with the restrictions $O_{2,N} \not> O_{1,N}$, $\{O_{1,N}, O_{2,N}\} \not> O_{2,N}$, and $O_2 \updownarrow O_3$. We will go stepwise from Game $\mathcal{G}_i$ to $\mathcal{G}_{i+1}$ for $i = 1..3$. For each step, we define an

adversary $\mathbf{A}_i$ whose goal is to distinguish between Game $\mathcal{G}_i$ to $\mathcal{G}_{i+1}$ and write $\Delta\mathcal{G}_{i,j}$ for the maximal advantage for all adversaries $\mathbf{A}_i$ to distinguish between Game $\mathcal{G}_i$ and $\mathcal{G}_j$, where all adversaries use equal query resources as $\mathbf{A}$.

We introduce an ideal encryption scheme, $\mathsf{IdealEnc}[\widetilde{\pi}, r]$ instead of $\mathsf{GTET}[\widetilde{\pi}, r]$ in the process and write $\mathsf{GTET}[\widetilde{\pi}, r]$ for the latter. We use a shorthand notation for the games, where we focus only on the nonce-based encryption scheme used in the oracles O_1, O_2, O_3 and indicate by M that real messages are encrypted, and by \$ that random messages are encrypted:

- $\Delta\mathcal{G}_{1,4} =^{\text{def}} \Delta_{\mathbf{A}}(\mathsf{GTET}[\widetilde{\pi}, r](M)\,;\,\mathsf{GTET}[\widetilde{\pi}, r](\$))$
- $\Delta\mathcal{G}_{1,2} =^{\text{def}} \Delta_{\mathbf{A}}(\mathsf{GTET}[\widetilde{\pi}, r](M)\,;\,\mathsf{IdealEnc}[\widetilde{\pi}, r](M))$
- $\Delta\mathcal{G}_{2,3} =^{\text{def}} \Delta_{\mathbf{A}}(\mathsf{IdealEnc}[\widetilde{\pi}, r](M)\,;\,\mathsf{IdealEnc}[\widetilde{\pi}, r](\$))$
- $\Delta\mathcal{G}_{3,4} =^{\text{def}} \Delta_{\mathbf{A}}(\mathsf{IdealEnc}[\widetilde{\pi}, r](\$)\,;\,\mathsf{GTET}[\widetilde{\pi}, r](\$))$

Again, the second and final step consider the same advantage that can be upper bounded by $\Delta\mathcal{G}_{1,2}$. Using the triangle inequality, we obtain

$$(3) \leq \Delta\mathcal{G}_{1,4} \leq \sum_{i=1}^{3} \Delta\mathcal{G}_{i,i+1}\,.$$

Prior, we consider an nE analysis of GTET of the game in the black-box setting. This result will then help bounding $\Delta\mathcal{G}_{1,2}$.

Difference $\Delta\mathcal{G}_{1,2}$. Next, we define an ideal encryption scheme as on the right-hand side of Algorithm 10. For each construction query (N^j, M^j), it

1. Samples all keys K_i^j as well as all would-be primitive outputs $C_{i,1}^j, \ldots, C_{i,r}^j$ uniformly and independently at random from $\mathbb{F}_2^n$,
2. Computes the leakages $L^{\oplus}$, L^{in}, and L^{out} with the help of its primitive oracle $\widetilde{\pi}$ and $\widetilde{\pi}^{-1}$, collects the would-be leakages into a vector $\mathbf{L}^j$, and
3. Outputs $(\mathbf{L}^j, C^j)$.

We will show that the advantage of a distinguisher $\mathbf{A}_{1,2}$ that tries to distinguish between Games $\mathcal{G}_1$ and $\mathcal{G}_2$ can be reduced to the advantage of an GTET-r-LUP distinguisher $\mathbf{A}_u$ on an isolated call to the $\mathsf{GTET}[\widetilde{\pi}, r]$ oracle by $\mathbf{A}_{1,2}$. For each primitive query of $\mathbf{A}_{1,2}$, $\mathbf{A}_u$ simply forwards the query to its corresponding primitive oracle, collects the query together with the output in a transcript $\tau_p = \{(K^i, S^i, X^i, Y^i)\}$ for $i \in [q_p]$ and forwards the response to $\mathbf{A}_{1,2}$. For each construction query of $\mathbf{A}_{1,2}$, (N^j, M^j) with $(M_1^j, \ldots, M_{m^j}^j) \xleftarrow{r \cdot n} M^j$ for $j \in [q_c]$, $\mathbf{A}_u$ does the following:

1. $\mathbf{A}_u$ samples $k \twoheadleftarrow [m^j]$, $K_1^j \twoheadleftarrow \mathbb{F}_2^n$, and $T_1^j \twoheadleftarrow \mathbb{F}_2^n$. Moreover, it initializes an empty list of leakages $\mathbf{L}$.
2. For $i = 1..k-1$, $\mathbf{A}_u$ queries its primitive oracle $\widetilde{\pi}$ with (K_i^j, i, T_i^j, N^j) to obtain $(L_i^j, K_{i+1}^j, T_{i+1}^j, C_i^j)$ and adds the leakage L_i^j to the list $\mathbf{L}^j$.
3. $\mathbf{A}_u$ queries its primitive oracle $\widetilde{\pi}$ with (K_k^j, k, T_k^j, N^j) to obtain $(L_i^j, K_{k+1}^j, T_{k+1}^j, Y_k^j)$. Then, $\mathbf{A}_u$ queries its GTET-r-LUP challenger with (K_k^j, k, T_k^j, N^j) to obtain $(L_k^j, K_{k+2}^j, T_{k+2}^j, C_{k+1}^j)$ from the response and adds the leakage to the list $\mathbf{L}^j$.

4. For $i = k+2..m^j$, $\mathbf{A}_u$ queries primitive oracle with (K_i^j, i, T_i^j, N^j) to obtain $(L_i^j, K_{i+1}^j, T_{i+1}^j, C_i^j)$ and adds the leakage to the list $\mathbf{L}^j$.
5. $\mathbf{A}_u$ returns $(\mathbf{L}^j, C^j)$ with $C^j = (C_1^j \parallel \ldots \parallel C_{m^j}^j)$ to $\mathbf{A}_{1,2}$.

At the end of the interaction, $\mathbf{A}_u$ outputs a set of guesses for the used tweakeys $\mathcal{K}$ that consists of all tweakeys in the primitive queries of $\mathbf{A}_{1,2}$, $\mathcal{K} = \{(K, S) : (K, S, X, Y) \in \tau_p\}$ for any X, $Y \in \mathbb{F}_2^n$.

We could exclude any bad queries such as tweakey collisions and upper bound their probability before. Since we assume then that no bad events occur among the guesses, then all tweakeys are fresh, and there is no difference between both games $\mathcal{G}_1$ with $\mathsf{GTET}[\widetilde{\pi}, r]$ and $\mathcal{G}_2$ with the ideal encryption scheme. The advantage of the distinguisher $\mathbf{A}_{1,2}$ is inherited by $\mathbf{A}_u$ who can measure its leakage of the LUP-4 query p times. The difference consists of the σ indices over the guess of k. Hence, the advantage between the games is upper bounded by

$$\Delta\mathcal{G}_{1,2} \leq \rho \cdot \mathbf{Adv}^{\mathsf{LUP\text{-}4}}_{F[\widetilde{\pi}]}(p, \sigma + q_c + q_p) + \mathbf{Adv}^{\mathsf{nE}}_{\mathsf{GTET}[\widetilde{\pi}, r]}(q, \rho, m, \sigma, q_p)\,.$$

Difference $\Delta\mathcal{G}_{2,3}$. Again, the output of random bits by IdealEnc would render it indistinguishable from random in the black-box setting. Though, under leakage, there is a remaining advantage that stems from the presence of XORs with plaintext material.

The XOR with a Partial Block. For GTET, those XORs are given only once per query in the computation of a keystream block that is truncated and XORed with a final partial plaintext block. Similar to the proof for RCTR, we can reduce the advantage to that of an XOR$ adversary, following the stepwise approach by Berti et al. of chunkwise hybrid adversaries. Though, we do not have to cover all queries, but at most a reduction of the final block.

For this purpose, we can define an adversary $\mathbf{A}_x$ that simulates a distinguisher $\mathbf{A}_{2,3}$ whose aim is to distinguish between Games $\mathcal{G}_2$ and $\mathcal{G}_3$. The simulation of $\mathbf{A}_x$ by $\mathbf{A}_{2,3}$ is highly similar to that before, but reduced to a single call. Whenever $\mathbf{A}_{2,3}$ asks a primitive query to $\widetilde{\pi}^{\pm}$, $\mathbf{A}_x$ answers it with its own primitive oracle $\widetilde{\pi}^{\pm}$, forwards the responds to $\mathbf{A}_{2,3}$, and stores the query and response into a transcript τ_p. For each construction query of $\mathbf{A}_{2,3}$, (N^j, M^j) with $(M_1^j, \ldots, M_{m^j}^j) \xleftarrow{r \cdot n} M^j$ for $j \in [q_c]$ and does the following.

1. $\mathbf{A}_x$ samples $K_1 \twoheadleftarrow \mathbb{F}_2^n$, $T_1 \twoheadleftarrow \mathbb{F}_2^n$. Moreover, it initializes an empty list of leakages $\mathbf{L}$.
2. For $i = 1..m^j - 1$, $\mathbf{A}_x$ samples unifomly random values K_{i+1}^j, T_{i+1}^j, C_i^j and leakage L_i^j as from IdealEnc and stores L_i^j into the list $\mathbf{L}^j$.
3. $\mathbf{A}_x$ samples K_k^j and T_k^j. Let $(M_{m,1}^j, \ldots, M_{m,c}^j) \xleftarrow{n} M_m^j$. $\mathbf{A}_x$ builds $S_{k,c}^j = (c+1, k, T_k^j)$, submits it together with $M_{k,c}^j$ to its XOR$ challenger to obtain $(L_k^j, C_{k,c}^j)$. It computes the would-be leakage $L_{k,c}^j$. It builds $C_k^j = C_{k,1}^j \parallel C_{k,1}^j$ and $L_k^j = L_{k,1}^j \parallel L_{k,2}^j$ and adds it to the list $\mathbf{L}^j$.
4. $\mathbf{A}_x$ returns $(\mathbf{L}^j, C^j)$ with $C^j = (C_1^j \parallel \ldots \parallel C_m^j)$ to $\mathbf{A}_{2,3}$.

5. $\mathbf{A}_x$ forwards the guess of b' from $\mathbf{A}_{2,3}$ to its own challenger.

Again, the advantage of $\mathbf{A}_x$ with p leakage trails each is inherited from $\mathbf{A}_{2,3}$. We have to build q_c such adversaries $\mathbf{A}_x$ over all construction queries that $\mathbf{A}_{2,3}$ queries. Hence, we obtain an advantage of

$$q_c \cdot \mathbf{Adv}^{\mathsf{XOR\$}}_{F[\widetilde{\pi}]}(q_c + q_p)\,.$$

The XOR of Keys and Tweaks. There is a similar XOR operation that we need to consider, though, it happens in each chunk: that of $T^j_{i+1} \leftarrow C^j_{i,1} \oplus K^j_{i+1}$. Here, two out of three values are unknown, which we modeled as the $\mathsf{XOR\$2}$ game. We can define a similar adversary that simulates an XOR and is called once per chunk. Again, this can be modeled as a stepwise reduction of hybrid games of an $\mathsf{XOR\$2}$ adversary $\mathbf{A}_{x2}$ that simulates $\mathbf{A}_{2,3}$, one per chunk of $\mathbf{A}_{2,3}$. As a result, we obtain an upper bound of the advantage of

$$\rho \cdot \mathbf{Adv}^{\mathsf{XOR\$2}}_{F[\widetilde{\pi}]}(\rho + q_p)\,.$$

Over both notions, we obtain the upper bound.

$$\Delta\mathcal{G}_{2,3} \leq q_c \cdot \mathbf{Adv}^{\mathsf{XOR\$}}_{F[\widetilde{\pi}]}(q_c + q_p) + \rho \cdot \mathbf{Adv}^{\mathsf{XOR\$2}}_{F[\widetilde{\pi}]}(\rho + q_p)\,.$$

Our claim in Lemma 4 follows from summing up the individual bounds. □

H.2 Proof of Lemma 10

Lemma 10. Let r and $n \geq 4$ be integers, $\widetilde{\pi} \twoheadleftarrow \mathsf{TBC}(\mathcal{K}, \mathcal{T}_D, \mathbb{F}_2^n)$, and $K \twoheadleftarrow \mathcal{K}$. Let $\mathbf{A}$ be an adversary on $\mathsf{GTET}[\widetilde{\pi}, r]_K$ that asks at most q_p primitive queries and q_e encryption queries and q_d decryption queries of at most m blocks corresponding to ρ chunks each and σ blocks in total. Then,

$$\mathbf{Adv}^{\mathsf{nE}}_{\mathsf{GTET}[\widetilde{\pi},r]_K}(\mathbf{A}) \leq \frac{2\sigma^2 + q_p\sigma}{2^{2n}} + \left(\frac{q}{2^n}\right)^n + \frac{(n+1)\cdot q_p}{2^n}\,.$$

Proof. Again, all queries by $\mathbf{A}$ will be stored together with their corresponding responses in a transcript $\tau = \{K, \tau_c, \tau_p\}$, where τ_c consists of exactly the construction queries of $\mathbf{A}$ and their corresponding responses, i.e., for encryption queries, it stores (N^i, M^i, C^i). We define m^i for the maximal number of chunks in M^i, where each chunk M^i_j consists of a sequence of $r \cdot n$-bits that are split into $M^i_{j,1}, \ldots, M^i_{j,r}$; the last chunk of M^i can consist of $\leq r \cdot n$ bits. τ_p represents the primitive queries of $\mathbf{A}$ to $\widetilde{\pi}^{\pm}$ and their associated responses, $(d^i, K^i, T^i, X^i, Y^i)$, where $Y^i \leftarrow \widetilde{\pi}^{T^i}_{K^i}(X^i)$ and $d^i = 1$ if it is a query in forward direction and $d^i = 0$ otherwise. We define a transcript τ as bad if any of the following events occurs:

- bad_1: There exist distinct $(i,j) \neq (i',j')$ with $i, i' \in [q_c]$ and $j \in [m^i]$, $j' \in [m^{i'}]$ such that $(K^i_j, T^i_j) = (K^{i'}_{j'}, T^{i'}_{j'})$.
- bad_2: There exist pairwise distinct $i_1, \ldots, i_n \in [q_c]$ and $j_1, \ldots, j_n$ such that $C^{i_1}_{j_1,1} = \cdots = C^{i_n}_{j_n,1}$ and $j_i < m^i$ for all $i \in \{i_1, \ldots, i_n\}$.

– bad_3: There exists $i \in [q_p]$ such that $K^i = K$.
– bad_4: There exists $i \in [q_c]$, $j \in [m^i]$ and $k \in [q_p]$ such that $(K^i_j, T^i_j) = (K^k, T^k)$.

We define BadT for the set of all bad transcripts and GoodT for the set of all good transcripts, i.e., transcripts that are not bad. Then, the proof follows from applying the Lemmas 2, 11, and 12.

Thus, it remains to upper bound the probability of bad events and to determine the interpolation ratio of good transcripts. We start with the former task.

Lemma 11. It holds that

$$\Pr[\Theta_{\text{ideal}} \in \text{BadT}] \leq \frac{2\sigma^2 + q_p\sigma}{2^{2n}} + \left(\frac{q}{2^n}\right)^n + \frac{(n+1)\cdot q_p}{2^n}\,.$$

Proof. In the following, we upper bound the probability of the individual bad events.

bad_1. In this event, a collision between two tweakeys in construction queries occurs. This means that

$$(K^i_j, T^i_j) = (K^{i'}_{j'}, T^{i'}_{j'})\,.$$

which is equivalent to

$$\begin{bmatrix} \widetilde{\pi}^{0,j-1,T^i_{j-1}}_{K^i_{j-1}}(M^i_j) = \widetilde{\pi}^{0,j'-1,T^{i'}_{j'-1}}_{K^{i'}_{j'-1}}(M^{i'}_{j'}) \\ \widetilde{\pi}^{1,j-1,T^i_{j-1}}_{K^i_{j-1}}(M^i_j) \oplus \widetilde{\pi}^{0,j-1,T^i_{j-1}}_{K^i_{j-1}}(M^i_j) = \widetilde{\pi}^{1,j'-1,T^{i'}_{j'-1}}_{K^{i'}_{j'-1}}(M^{i'}_{j'}) \oplus \widetilde{\pi}^{0,j'-1,T^{i'}_{j'-1}}_{K^{i'}_{j'-1}}(M^{i'}_{j'}) \end{bmatrix}.$$

Given the equality in the top row, we can simplify the lower row to

$$\begin{bmatrix} \widetilde{\pi}^{0,j-1,T^i_{j-1}}_{K^i_{j-1}}(M^i_j) = \widetilde{\pi}^{0,j'-1,T^{i'}_{j'-1}}_{K^{i'}_{j'-1}}(M^{i'}_{j'}) \\ \widetilde{\pi}^{1,j-1,T^i_{j-1}}_{K^i_{j-1}}(M^i_j) = \widetilde{\pi}^{1,j'-1,T^{i'}_{j'-1}}_{K^{i'}_{j'-1}}(M^{i'}_{j'}) \end{bmatrix}. \tag{5}$$

W.l.o.g., that this tweakey collision is the lexicographically first collision considering the indexing (i, i', j, j'). Otherwise, the present tweakey collision is a consequence of an earlier collision and we consider the lexicographically first such collision. Since it is the first tweakey collision, it must hold that

$$(K^i_{j-1}, T^i_{j-1}) \neq (K^{i'}_{j'-1}, T^{i'}_{j'-1})\,.$$

Therefore, all four calls to $\widetilde{\pi}$ consider different permutations and are pairwise independent from each other. Since it is the first collision, they do not collide with other tweakeys. Thus, the probability for each equality therein is 2^{-n}. It

follows that, over all σ blocks in construction queries, the probability for this event is at most

$$\Pr[\mathsf{bad}_1] \leq \frac{\binom{\sigma}{2}}{(2^n - 2\sigma)^2} \leq \frac{2\sigma^2}{2^{2n}} ,$$

given $\sigma \leq 2^{n-2}$. For the subsequent bad events, we assume that bad_1 did not occur, i.e., we condition on $\neg\mathsf{bad}_1$.

bad_2. In this case, an n-multicollision in values $C_{i,1}$, for $i \in [q]$, occurs. For this event, it has to hold that

$$\Pr\left[C^{i_1}_{j_1,1} = \cdots = C^{i_n}_{j_n,1}\right] = \Pr\left[\widetilde{\pi}^{1,j_1,T^{i_1}}_{K^{i_1}_{j_1}}(M^i_{j_1,1}) = \cdots = \widetilde{\pi}^{1,j_n,T^{i_n}}_{K^{i_n}_{j_n}}(M^i_{j_n,1})\right] .$$

Given that bad_1 has not occurred, all blocks come from pairwise independent permutations that have not been queried anywhere else. Therefore, the probability for each output is 2^{-n}. Over all q queries, we obtain ρ chunks, where the final chunk of each query is not considered since the tweakey it generates will not used for any subsequent tweakey input.

$$\Pr[\mathsf{bad}_2|\neg\mathsf{bad}_1] \leq \binom{\rho - q}{n} \cdot \left(\frac{1}{2^n}\right)^{n-1} \leq \frac{(\rho - q)^n}{n! \cdot 2^{n(n-1)}} \leq \left(\frac{(\rho - q)}{2^n}\right)^n$$

using $n! \geq 2^n$ for $n \geq 4$.

bad_3. For this event, a key collision between a TGF query and a primitive query occurs for $i \in [q_p]$ such that $K^i = K$. Since the adversary must hit the correct secret key K under the assumption that it does not leak, the probability is at most

$$\Pr[\mathsf{bad}_3|\neg\mathsf{bad}_1 \wedge \neg\mathsf{bad}_2] \leq \frac{q_p}{2^n} .$$

bad_4. Finally, a tweakey collision between a construction block and a primitive query can occur. It must hold that

$$\Pr\begin{bmatrix} K^i_j = K^k \\ K^i_j \oplus C^i_{j,1} = T^k \end{bmatrix} = \Pr\begin{bmatrix} \widetilde{\pi}^{0,T^i_{j-1},j}_{K^i_{j-1}}(M^i_{j-1}) = K^k \\ \widetilde{\pi}^{0,j,T^i_{j-1}}_{K^i_{j-1}}(M^i_{j-1}) \oplus \widetilde{\pi}^{1,j,T^i_{j-1}}_{K^i_{j-1}}(M^i_{j-1}) = T^k \end{bmatrix} .$$

Conditioning on $\neg\mathsf{bad}_1$ means that all tweakeys are distinct.

We can distinguish between two cases:

Case 1: The construction query occured before the primitive query. Further conditioning on $\neg\mathsf{bad}_2$, we know that each value $C^i_{j,1}$, for all $i \in [q_c]$ and $j \in [m^i]$, occurs at most n times in the transcript. Thus, $\mathbf{A}$ can choose K^k for a primitive query and can select T^k to cover at most n values $C^1_{j,1}$ at a time. Over all queries, the probability in this case is at most

$$\frac{n \cdot q_p}{2^n} .$$

Case 2: The construction query occured after the primitive query. Since bad_1 did not occur, in this case, the tweakey was not used before and the output from the construction query was sampled uniformly at random from 2^n values. Thus, the probability to hit any primitive query is 2^{-2n}, and over all queries

$$\frac{q_p \cdot \sigma}{2^{2n}}$$

Over both cases, we obtain

$$\Pr\left[\mathsf{bad}_4 | \neg\mathsf{bad}_1 \wedge \neg\mathsf{bad}_2\right] \leq \frac{n \cdot q_p}{2^n} + \frac{q_p \cdot \sigma}{2^{2n}} \,.$$

The bound in Lemma 11 follows then from

$$\Pr\left[\mathsf{bad}\right] \leq \Pr\left[\mathsf{bad}_1\right] + \Pr\left[\mathsf{bad}_2 | \neg\mathsf{bad}_1\right] + \sum_{i=3}^{4} \Pr\left[\mathsf{bad}_i | \neg\mathsf{bad}_1 \wedge \neg\mathsf{bad}_2\right] \,.$$

Lemma 12. For any good transcript τ, it holds that

$$\frac{\Pr[\Theta_{\text{real}} = \tau]}{\Pr[\Theta_{\text{ideal}} = \tau]} = 1 \,.$$

Proof. Let $\mathcal{T} = \{S^1, \ldots, S^{q_p}\}$ denote the set of all tweaks in primitive queries and let us reorder them to eliminate duplicates as $\mathcal{T} = \{W^1, \ldots, W^r\}$ such that $r \leq q_p$ is the number of distinct tweaks. The probability of choosing the initial key is $\Pr[K] = 2^{-n}$.

Let further b denote the number of bits from partial final blocks in construction queries, and σ' the number of full blocks over all queries.

In the ideal world, all output bits of any block $C^i_{j,k}$ are sampled independently uniformly at random. Given a randomly chosen tweakable block cipher $\widetilde{\pi} \twoheadleftarrow \mathsf{TBC}(\mathbb{F}_2^n, \mathcal{T}_D, \mathbb{F}_2^n)$ for primitive queries, the probability of its primitive outputs is

$$\Pr\left[\Theta_{\text{ideal}} = \tau\right] = \frac{1}{|\mathcal{K}|} \cdot \frac{1}{2^b} \cdot \frac{1}{2^{\sigma' \cdot n}} \cdot \prod_{i=1}^{r} \frac{1}{(2^n)_{\ell_i}} \,.$$

Considering the real world, we observe that a good transcript lacks collisions of tweakeys in construction queries. Moreover, the nonce requirement ensures that no partial blocks can collide. The probability that the primitive is compatible to the transcript is again

$$\Pr[\widetilde{\pi} \in \mathsf{Comp}(\tau)] = \prod_{i=1}^{r} \frac{1}{(2^n)_{\ell_i}} \,.$$

Hence,

$$\Pr\left[\Theta_{\text{real}} = \tau\right] = \frac{1}{|\mathcal{K}|} \cdot \frac{1}{2^b} \cdot \frac{1}{2^{\sigma' \cdot n}} \cdot \prod_{i=1}^{r} \frac{1}{(2^n)_{\ell_i}} \,.$$

The claim in Lemma 12 follows.

I Relation to Indifferentiability

The indifferentiability framework by Maurer et al. [41] is a simulation-based security model which tries to capture what a construction would need to be as good as an ideal object. Thus, it captures also yet unknown attacks. Indifferentiability has been applied to authenticated encryption schemes by Barbosa and Farshim [3]. Though, indifferentiability in general and their work in particular showed three limitations.

First, their work outlined that nonce-misuse resistance or robustness were necessary (though not always sufficient) for schemes to be indifferentiable from the ideal AE scheme, which is a conflict in the understanding of what leakage resilience could provide by the school by Standaert et al.

Second, the applicability of indifferentiability to show leakage resilience is not well-understood yet. The original indifferentiability framework captures only security notions that can be evaluated as single-stage games. Ristenpart et al. [51] had identified leakage resilience as a multi-stage game. Only the work by Barbosa and Farshim [3] showed that certain multi-stage games can be rewritten as single-stage games, including the setting of leakage resilience. However, a formal treatment has not been conducted yet, and is an interesting open task.

Third, due to technical details, security bounds from indifferentiability proofs are often inferior by magnitudes compared to bounds in the ideal-primitive or standard models. Since our focus lied on quantitative bounds under common assumptions, this work had opted for the established models in this work. Nevertheless, proofs in the indifferentiability setting could be valuable for increasing the security of schemes under leakage.