IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2023, No. 4, pp. 330–364. DOI:10.46586/tosc.v2023.i4.330-364

On Large Tweaks in Tweakable Even-Mansour with Linear Tweak and Key Mixing

Benoît Cogliati[1], Jordan Ethan[2], Ashwin Jha[2] and Soumya Kanti Saha[3]

[1] Thales DIS France SAS, Meudon, France
benoit.cogliati@gmail.com
[2] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
{jordan.ethan,ashwin.jha}@cispa.de
[3] Indian Institute of Science, Bengaluru, India
soumya.riju97@gmail.com

Abstract. In this paper, we provide the first analysis of the Iterated Tweakable Even-Mansour cipher with linear tweak and key (or tweakey) mixing, henceforth referred as TEML, for an arbitrary tweak(ey) size kn for all $k \geq 1$, and arbitrary number of rounds $r \geq 2$. Note that TEML captures the high-level design paradigm of most of the existing tweakable block ciphers (TBCs), including SKINNY, Deoxys, TweGIFT, TweAES etc. from a provable security point of view. At ASIACRYPT 2015, Cogliati and Seurin initiated the study of TEML by showing that 4-round TEML with a $2n$-bit uniform at random key, and n-bit tweak is secure up to $2^{2n/3}$ queries. In this work, we extend this line of research in two directions. First, we propose a necessary and sufficient class of linear tweakey schedules to absorb mn-bit tweak(ey) material in a minimal number of rounds, for all $m \geq 1$. Second, we give a rigorous provable security treatment for r-round TEML, for all $r \geq 2$. In particular, we first show that the $2r$-round TEML with a $(2r+1)n$-bit key, αn-bit tweak, and a special class of tweakey schedule is IND-CCA secure up to $O(2^{\frac{r-\alpha}{r}n})$ queries. Our proof crucially relies on the use of the *coupling* technique to upper-bound the statistical distance of the outputs of TEML cipher from the uniform distribution. Our main technical contribution is a novel approach for computing the probability of failure in coupling, which could be of independent interest for deriving tighter bounds in coupling-based security proofs. Next, we shift our focus to the chosen-key setting, and show that $(r+3)$-round TEML, with rn bits of tweakey material and a special class of tweakey schedule, offers some form of resistance to chosen-key attacks. We prove this by showing that $r+3$ rounds of TEML are both necessary and sufficient for *sequential indifferentiability*. As a consequence of our results, we provide a sound provable security footing for the TWEAKEY framework, a high level design rationale of popular TBC.

Keywords: TEM · indifferentiability · indistinguishability · coupling

1 Introduction

Tweakable Block Ciphers (or TBCs in short) are symmetric-key cryptographic primitives that, in addition to the usual secret key of a standard block cipher, take an additional public indexing input, called *tweak*. In a seminal work [LRW11], Liskov et al. formalized the concept of a TBC, $\widetilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$, as a family of permutations on the plaintext/ciphertext space $\mathcal{M}$, and indexed by two parameters: the secret key $K \in \mathcal{K}$ and the public tweak $T \in \mathcal{T}$.

Owing to their versatility, tweakable block ciphers have a broad range of applicability, most notably in authenticated encryption schemes [LRW11, Rog04, PS16], and message

Received: 2023-09-01 Accepted: 2023-11-01 Published: 2023-12-08

authentication codes [Nai15, IMPS17, CLS17, GLN19, CLL22]. Apart from these, TBCs have also been employed to achieve other symmetric-key security goals, e.g., [Min09, RZ11, JN18, BLN18].

Historically, TBCs were mostly constructed on top of a block cipher. Indeed, there is a plethora of block cipher-based TBC constructions, such as LRW1 and LRW2 by Liskov et al. [LRW11], Rogaway's XEX [Rog04] and its future refinements [CS08, Min06, GJMN16], $\tilde{\mathsf{F}}[1]$ and $\tilde{\mathsf{F}}[2]$ by Mennink [Men15a, Men15b], and XHX by Jha et al. [JLM+17] etc. Several other constructions [LST12, LS13b, BGGS20, Men18, JN20, LL18] employed a cascade of previously mentioned constructions to obtain a higher security guarantee.

All the aforementioned constructions are based on block ciphers. However, there also exist constructions that are based on public random permutation. In [CLS15], Cogliati, Lampe and Seurin introduced the Tweakable Even-Mansour (TEM) construction and its cascaded variant. They showed that, as long as the round subkeys are derived from the tweak using keyed hash functions, the two round construction is secure up to roughly $2^{2n/3}$ queries. They also proved that the $2r$-round construction is secure up to $2^{rn/(r+1)}$ queries. Later, Cogliati and Seurin [CS15a] proved that the 4-round tweakable Even-Mansour with a linear tweak and key (tweakey) mixing, also called TEML, is secure up to $2^{2n/3}$ queries. Dutta [Dut20] proved a similar result with a smaller number of independent permutations.

TWEAKEY and the Advent of Dedicated Design Strategies: In [JNP14], Jean et al. introduced a new dedicated design strategy for TBC designs. This completely revolutionized the design landscape of concrete TBCs. Most notably, they proposed the Superposition TWEAKEY (STK)[1] construction which is derived from the key-alternating cipher. Specifically, they proposed using p words of unified tweak and key (that they dubbed a tweakey), from which round tweakeys are extracted and xored between application of AES-like rounds. Between rounds, the nibbles of the p tweakey words are shuffled, and then each c-bit cell of the j-th bit tweakey word is multiplied by a constant α_j in $\mathsf{GF}(2^c)$. Round tweakeys are then simply computed by XORing all the p-words of the tweakey. Critically, the tweakey schedule is a linear function of the key and the tweak. This framework has been the basis of most recent TBCs such as Deoxys-BC [JNP14], Joltik-BC [JNP14], Kiasu-BC [JNP14], and Skinny [BJK+16]. The success of the TWEAKEY framework also motivated new design frameworks for more specialized usage, such as the Elastic-Tweak framework [CDJ+21] for TBCs with small tweak size like TweAES and TweGIFT.

While the rationale behind the TWEAKEY framework has been extensively tested through the cryptanalysis of various proposals, it has seen little theoretical analysis. Notably, it is well-known that the high-level design of STK largely follows the Tweakable Even-Mansour construction. However, the only work that provides an asymptotically tight bound for TEM requires the use of almost-universal hash functions. A typical tweak and key mixing of this type would be the multiplication of the key and the tweak in $\mathsf{GF}(2^n)$, where n denotes the block size. This deviates significantly from the STK construction and presents obvious performance drawbacks. The only works dealing with linear tweak and key mixing currently only consider 4-round constructions. In addition, while STK allows for large-tweak values in design, there is no suitable theoretical model for analyzing TEML with tweak size larger than n bits. Given the recent push [NI19, BGIM19, CJPS22] from the community towards leveraging large tweak size, it is necessary to come up with a sound theoretical mechanism for it. The main goal of the paper is to study the TEML construction for arbitrary number of rounds and possibly large tweak size. This allows us to give a theoretically sound argument in favor of the resistance of STK-based TBCs against generic attacks in the indistinguishability setup, where the goal is to show indistinguishability

[1] STK corresponds to a class of tweakey schedules.

from a secret tweakable random permutation assuming a uniform at random and secret key.

Chosen-Key Model and Sequential Indifferentiability: In practice, however, (tweakable) block ciphers are often employed in the known-key and chosen-key [KR07, BKN09] attack models, where the adversary either knows the random key or, in an even more stronger setting, can instantiate the block cipher with its choice of key at each invocation. Knudsen and Rijmen [KR07] suggested *correlation intractability* [CGH98] notion due to Canetti et al. as a possible theoretical formalization to capture the KKA and CKA models. In [MPS12], Mandal et al. introduced the notion of *sequential indifferentiability* which directly implies and, since then, is the go to method to prove collision intractability. Note that, sequential indifferentiability is a weaker variant of the classical indifferentiability notion by Maurer et al. [MRH04], where the adversary is compelled to make all the simulator (res. internal primitive) queries before it can make any ideal cipher (res. construction) queries.

The indifferentiability of Even-Mansour ciphers has been an active area of research with a series of papers proving both indifferentiability [ABD+13, LS13a, DSST17, GL16] and its sequential counterpart [CS15b, CS16, XDG22]. Interestingly though all these works, either consider a trivial key schedule, where the round keys are the same as the master key, and hence the master key size is same as the permutation input size, say n, or employ independent random oracles to derive the round keys from the master key. This makes it difficult to employ these results directly to TWEAKEY based block ciphers where the tweakey size is usually bigger than the block size and the tweakey schedule is, in general, linear. So, it would be interesting to see how these changes affect the security in the indifferentiability setting. In this paper, we take the first step by studying the sequential indifferentiability of TWEAKEY block ciphers with arbitrary key size and a special class of linear key scheduling.

1.1 Our Contributions

Our contributions are twofold.

I. In section 3, using the well-known coupling technique, we show that a $2r$-round Tweakable Even-Mansour cipher with αn-bit tweak, and weak α-bijective tweak schedule (see Definition 3.2), is secure up to $2^{\frac{r-\alpha}{r}n}$ chosen plaintext and chosen ciphertext queries, under the assumption that the round permutations are independent public random permutation and round keys are chosen independently and uniformly at random.

   Our proof extends the proof strategy used in [LS14, CLS15], and introduces, what we call, the *activity patterns* — a succinct string representation of the coupling failure event (see section 4). Basic coupling proofs proceed iteratively by upper-bounding, for each round, the probability to get various collision events using the randomness of the round key. This simple approach can fail if collision events can span several rounds. Looking ahead briefly, a collision between two inputs of the j-th round permutation for the TEML construction with n-bit tweak gives rise to an equation of the form $x_i^j \oplus t_i = x_{i'}^j \oplus t_{i'}$, where the key has been eliminated. This forces us to consider the inputs to the previous round permutation, which may both be involved in another collision, and so on (possibly) down to the first round. In order to solve this issue, we introduce the idea of activity patterns: instead of considering each round individually, we consider the succession of collision events throughout the full evaluation of the construction, and sum over all possible choices. Our main contribution is to give a fine-grained analysis of the r-TEML construction.

Table 1.1: Comparison of sequential indifferentiablity results on TEML. The column **Complex.** indicates the simulator query/time complexity.

Rounds	Primitives	Key Size	Complex.	Bounds	Ref.
4	4	n	$\mathcal{O}(q^2)$	$\mathcal{O}\left(\frac{q^4}{N}\right)$	[CS15b]
4	2	n	$\mathcal{O}(q^2)$	$\mathcal{O}\left(\frac{q^4}{N}\right)$	[XDG22]
$r+3$	$r+3$	rn	$\mathcal{O}\left(q^{r+1}\right)$	$\mathcal{O}\left(\frac{q^{2r+2}}{N}\right)$	Sec. 6

II. In section 6 we prove the sequential indifferentiability of (Tweakable) Even-Mansour cipher with rn-bit key (or tweakey) and any weak r-bijective key schedule. Specifically, we show an attack on $r+2$ rounds and prove the security for $r+3$ rounds, thereby establishing the necessary and sufficient number of rounds for security in sequential indifferentiability setting. Note that our result directly implies the security of $(r+3)$-round (Tweakable) Even-Mansour against chosen key attacks.

   Note that, both [XDG22] and this paper build over [CS15b]. In fact, our result can be seen as a generalization of [XDG22], for larger key (key size $\geq rn$-bit for $r \geq 1$) and typical linear TWEAKEY schedules. Table 1.1 gives a comparison of the three results.

A Note on Our Choice of Proof Technique: We employ the coupling technique [Ald83] to establish the IND-CCA bound. One can argue that the H-coefficients technique [Pat08] might have the potential to derive very tight security bound, as has been demonstrated [CS14] in the case of key alternating ciphers. However, we note that, in the tight security analysis of key alternating ciphers for an arbitrary number of rounds, the main technical step is actually a combinatorial result (see [CS14, Lemma 1]) that gives a very sharp lower bound on the number of permutations that can realize a given transcript. Indeed, all the existing tight security analyses of key alternating cipher, be it [CS14] or a subsequent work by Hoang and Tessaro [HT16], employ this key result. Its proof utilizes two crucial observations. Firstly, the secret round keys or masks (which are independent of the queries) can be simply subsumed within the permutation calls. Second, and somewhat more importantly, there are no internal input (corresponding to the internal permutation calls) collisions between any two distinct queries. These two facts together help in deriving a conditional lower bound for the current query based on the lower bound for the previous queries. Unfortunately, in the case of TEM, unless the tweak is a constant (equivalent to a key alternating cipher), these two observations no longer apply. First, the secret round masks are tweak-dependent, and thus, depend on adversarial queries. Second, for two distinct tweaks, there can be internal input collisions. As a result, the previous combinatorial result is not applicable directly, and as of now, it seems hard to extend when there are multiple tweaks in play, even for a very small tweak space. Indeed, coming up with a similar result for even AXU hash-based tweakey schedule, let alone the linear tweakey schedule, seems technically challenging. On the other hand, there is no existing analysis for TEML with arbitrary number of rounds and arbitrary tweak sizes. It is our firm belief that such analyses (even with a loose security bound) could shed some light on the provable security of the high level design strategy of the popular TWEAKEY framework. Given the apparent need for such an excursion, the technical challenges in reusing/extending the existing results on key alternating ciphers, and inspired by the pragmatic approach from [LS14, CLS15], we employ the coupling technique to derive probably a non-tight yet meaningful security bound for arbitrary number of rounds and arbitrary tweak size. We also note that, apart from giving some security guarantee for arbitrary number of rounds, the coupling-based analysis is also useful in getting a good indication on what could be the tight security bound. This could serve as a motivation

and a plausible target bound for future endeavors in this direction.

2 Preliminaries

General Notations. Let $\mathbb{N}$ denote the set of all positive integers, and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. For $i \leq j \in \mathbb{N}_0$, $[\![i, j]\!]$ denotes the set $\{i, \ldots, j\}$. For $n \geq k \in \mathbb{N}_0$, the falling factorial is defined as $(n)_k := n(n-1)...(n-k+1)$. For a finite set $\mathcal{X}$, we write $x \leftarrow_{\$} \mathcal{X}$ to denote the uniform at random draw of x from $\mathcal{X}$.

Alphabet and String. An alphabet Γ is a finite non-empty set of symbols. For any alphabet Γ, Γ^* denotes the set of all strings over Γ, including the empty string. For any string $x \in \Gamma^*$, $|x| = r$, denotes the length of x, where $x = x_1 \cdots x_r$ and $x_i \in \Gamma$ for all $i \in [\![1, r]\!]$. For any two strings, $x, y \in \Gamma^*$, the *concatenation* of x and y is denoted by xy. We say that string x is a *prefix* of string y, denoted $x \subseteq y$, if a string z exists such that $xz = y$ and that x is a *proper prefix* of y, denoted $x \subset y$ if $x \neq y$. For $s \in \mathbb{N}$, x^s denotes the string $x \cdots x$ of length s for some symbol $x \in \Gamma$, and Γ^s denotes the set of all strings over Γ of length s.

(Tweakable) Block Cipher. A *block cipher* with key space $\{0,1\}^\kappa$ and message space $\{0,1\}^n$ is a mapping $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ such that for any $k \in \{0,1\}^\kappa$, $m \mapsto E(k, m)$ is a permutation of $\{0,1\}^n$. For $n \in \mathbb{N}$, $\mathsf{BC}(\kappa, n)$ denotes the set of all block ciphers with key space $\{0,1\}^\kappa$ and message space $\{0,1\}^n$, and $\mathsf{P}(n)$ denotes the set of all permutations of $\{0,1\}^n$.

A *tweakable block cipher* with key space $\{0,1\}^\kappa$, tweak space $\{0,1\}^\tau$ and message space $\{0,1\}^n$ is a mapping $\widetilde{E} : \{0,1\}^\kappa \times \{0,1\}^\tau \times \{0,1\}^n \to \{0,1\}^n$ such that for any $k \in \{0,1\}^\kappa$ and any tweak $t \in \{0,1\}^\tau$, $m \mapsto E(k, t, m)$ is a permutation of $\{0,1\}^n$. $\widetilde{\mathsf{BC}}(\kappa, \tau, n)$ denotes the set of all tweakable block ciphers with key space $\{0,1\}^\kappa$, tweak space $\{0,1\}^\tau$ and message space $\{0,1\}^n$. Similarly, we denote by $\widetilde{\mathsf{P}}(\tau, n)$ the set of all tweakable permutations with tweak space $\{0,1\}^\tau$ and message space $\{0,1\}^n$, i.e. the set of all functions $P : \{0,1\}^\tau \times \{0,1\}^n \longrightarrow \{0,1\}^n$ such that, for all $t \in \{0,1\}^\tau$, $P(t, \cdot)$ is a permutation of $\{0,1\}^n$.

Throughout, we fix κ, τ, and n as the key size, tweak size and block size, respectively. We also write $N := 2^n$. In addition, we often use the term *tweakey* to refer to key and tweak input in a combined manner.

Oracle and Adversary. An oracle $\mathcal{O}$ is simply an interface to some function. An adversary, $\mathcal{A}$ is an interactive Turing machine (or an algorithm) that interacts with a given set of oracles in a black box fashion and returns a bit output at the end of the interaction. For an oracle $\mathcal{O}$, $\mathcal{A}^{\mathcal{O}}$ denotes the output of $\mathcal{A}$ after its interaction with $\mathcal{O}$. $\mathcal{A}^{\mathcal{O}^{\pm}}$ denotes that $\mathcal{A}$ has bidirectional access to $\mathcal{O}$, i.e., oracle $\mathcal{O}$ and its inverse. In this paper, we assume that the adversary is non-trivial, i.e., it never makes a duplicate query, and it never makes a query for which the response is already known due to some previous query.

2.1 IND-CCA Security Under the Random Permutation Model

Let $\widetilde{E} : \{0,1\}^\kappa \times \{0,1\}^\tau \times \{0,1\}^n \to \{0,1\}^n$ be a tweakable block cipher that is constructed over a tuple of independent and uniform random permutations $\mathbf{P} = (P_1, \ldots, P_r)$, denoted by $\widetilde{E}^{\mathbf{P}}$, where each $P_i \leftarrow_{\$} \mathsf{P}(n)$.
The IND-CCA game describes the goal of an adversary $\mathcal{A}$ to distinguish between two pairs of oracles using adaptive bidirectional queries:

- the *real world* oracle, where $\mathcal{A}$ can make adaptive bidirectional queries to $(\widetilde{E}_k^{\mathbf{P}}, \mathbf{P})$; and

- the *ideal world* oracle $(\widetilde{\Pi}, \mathbf{P})$, where $\widetilde{\Pi} \leftarrow\!\!\$\, \widetilde{\mathsf{P}}(\tau, n)$ independent from $\mathbf{P}$.

After issuing all its queries and obtaining the corresponding responses, $\mathcal{A}$ outputs a single bit of response.

Definition 2.1. Let $q_c, q_p \in \mathbb{N}$, and $\varepsilon \in [0,1]$ be some security parameters. A tweakable block cipher $\widetilde{E}^{\mathbf{P}}$ is said to be (q_c, q_p, ε)-IND-CCA secure if

$$\mathbf{Adv}^{\mathsf{ind\text{-}cca}}_{\widetilde{E}^{\mathbf{P}}}(q_c, q_p) := \max_{\mathcal{A} \in \mathbb{A}(q_c, q_p)} \left| \Pr\left[\mathcal{A}^{\widetilde{E}^{\mathbf{P}}_k, \mathbf{P}} = 1\right] - \Pr\left[\mathcal{A}^{\widetilde{\Pi}, \mathbf{P}} = 1\right] \right| \leq \varepsilon, \tag{1}$$

where $\mathbb{A}(q_c, q_p)$ denotes the set of all adversaries that make at most q_c queries to the left oracle, and at most q_p queries to $\mathbf{P}$, and the probabilities are taken over the uniformly random and independent choices of k, $\mathbf{P}$, and $\widetilde{\Pi}$.

2.2 Sequential Indifferentiability and Chosen-Key Security

Let $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher that is constructed over a tuple of independent and uniform random permutations $\mathbf{P} = (P_1, \ldots, P_r)$, denoted by $E^{\mathbf{P}}$, where each $P_i \leftarrow\!\!\$\, \mathsf{P}(n)$.
The indifferentiability game describes the goal of an adversary $\mathcal{A}$ to distinguish between two pairs of oracles using adaptive bidirectional queries:

- the *real world* oracle, $(E^{\mathbf{P}}_k, \mathbf{P})$, where $E^{\mathbf{P}}_k$ and $\mathbf{P}$ are sometimes referred as the left and right oracle, respectively; and
- the *ideal world* oracle, $(\widetilde{\Pi}, \mathsf{Sim})$, where $\widetilde{\Pi} \leftarrow\!\!\$\, \widetilde{\mathsf{P}}(\tau, n)$ (the left oracle), and Sim (the right oracle) is an oracle Turing machine, referred as the simulator, with bidirectional oracle access to $\widetilde{\Pi}$.

Let $\mathcal{A}$ be an adversary accessing a pair of oracles that we denote generically as $(\mathcal{C}, \mathcal{P})$. $\mathcal{A}$ is said to be *sequential* if after its first query to its left oracle $\mathcal{C}$, it does not query its right oracle $\mathcal{P}$. Hence, such an adversary works in two phases: first it queries only $\mathcal{P}$, and then only $\mathcal{C}$. We define the total oracle query cost of $\mathcal{A}$ as the total number of queries received by the right oracle (from $\mathcal{A}$ or $\mathcal{C}$) when $\mathcal{A}$ interacts with $(\mathcal{C}^{\mathcal{P}}, \mathcal{P})$. In particular, if $\mathcal{C}$ makes c queries to $\mathcal{P}$ to answer any query it receives, and if $\mathcal{A}$ makes q_c queries to its left oracle and q_p queries to its right oracle, then the total oracle query cost is at most $q_p + cq_c$.

Definition 2.2. Let $q, \sigma, \mathtt{T} \in \mathbb{N}$ and $0 < \varepsilon \in \mathbb{R}$ be some security parameters. A block cipher $E^{\mathbf{P}}$ is said to be $(q, \sigma, \mathtt{T}, \varepsilon)$-sequential indifferentiable from an ideal cipher if there exists an oracle simulator Sim such that

$$\mathbf{Adv}^{\mathsf{seq\text{-}indiff}}(q, \sigma, \mathtt{T}) := \max_{\mathcal{A} \in \mathbb{A}(q)} \left| \Pr\left[\mathcal{A}^{\widetilde{\Pi}, \mathsf{Sim}^{\widetilde{\Pi}}} = 1\right] - \Pr\left[\mathcal{A}^{E^{\mathbf{P}}, \mathbf{P}} = 1\right] \right| \leq \varepsilon, \tag{2}$$

where $\mathbb{A}(q)$ denote the set of all adversaries that make at most q queries. Here, Sim makes at most σ oracle queries, and runs in time at most $\mathtt{T}$.

Sequential Indifferentiability to Chosen-Key Security. While it is well-known [CGH98] that a rigorous definition of chosen-key security is impossible in the standard model, the idealized models help us avoid classical impossibility results. This is done using the notion of evasive relations.

Definition 2.3. An m-ary relation $\mathcal{R}$ is said to be (q, ε)-evasive with respect to an ideal cipher $\widetilde{\Pi}$ if for any oracle Turing machine $\mathcal{M}$ making at most q oracle queries, one has

$$\Pr\left[\widetilde{\Pi} \leftarrow\!\!\$\, \widetilde{\mathsf{P}}(\kappa, n), (\alpha_i)_{i \in [\![1,m]\!]} \leftarrow \mathcal{M}^{\widetilde{\Pi}} : (\alpha_i, \widetilde{\Pi}(\alpha_i))_{i \in [\![1,m]\!]} \in \mathcal{R}\right] \leq \varepsilon.$$

Informally, a relation is evasive if it is hard, for any algorithm with oracle access to an ideal cipher, to output an m-tuple of inputs $(\alpha_i)_{i\in[\![1,m]\!]}$ such that $(\alpha_i, \widetilde{\Pi}(\alpha_i))_{i\in[\![1,m]\!]}$ satisfies the relation.

A similar notion can be defined through correlation intractability, when we consider a block cipher $E^{\mathbf{P}}$ constructed over a tuple of random permutations $\mathbf{P}$.

Definition 2.4. Let $E^{\mathbf{P}}$ be a block cipher construction over a tuple of independent and uniform random permutations $\mathbf{P}$, and let $\mathcal{R}$ be an m-ary relation. $E^{\mathbf{P}}$ is said to be (q,ε)-correlation intractable with respect to $\mathcal{R}$ if, for any Turing machine $\mathcal{M}$ with oracle access to $\mathbf{P}$ making at most q oracle queries, one has

$$\Pr\left[\mathbf{P} \leftarrow_{\$} (\mathsf{P}(n))^r, (\alpha_i)_{i\in[\![1,m]\!]} \leftarrow \mathcal{M}^{\mathbf{P}} : (\alpha_i, E^{\mathbf{P}}(\alpha_i))_{i\in[\![1,m]\!]} \in \mathcal{R}\right] \leq \varepsilon.$$

We will deem a block cipher construction $E^{\mathbf{P}}$ resistant to chosen-key attacks if, for every relation $\mathcal{R}$ that is (q,ε)-evasive with respect to an ideal cipher, $E^{\mathbf{P}}$ is (q',ε')-correlation intractable with respect to $\mathcal{R}$, with $q' \approx q$ and $\varepsilon' \approx \varepsilon$. The link between correlation intractability and sequential indifferentiability comes from the following result based on [MPS12, Theorem 3].

Theorem 2.1. *[CS15b, Theorem 4] Let $E^{\mathbf{P}}$ be a block cipher constructed over a tuple of independent and uniform random permutations $\mathbf{P}$ such that $E^{\mathbf{P}}$ makes at most c queries to $\mathbf{P}$ on any input. Assume that $E^{\mathbf{P}}$ is $(q+cm,\sigma,\mathtt{T},\varepsilon_{\mathcal{SI}})$-sequential indifferentiable from an ideal cipher. Then, for any m-ary relation $\mathcal{R}$, if $\mathcal{R}$ is $(\sigma+m,\varepsilon_{\mathcal{R}})$-evasive with respect to an ideal cipher, then $E^{\mathbf{P}}$ is $(q,\varepsilon_{\mathcal{SI}}+\varepsilon_{\mathcal{R}})$-correlation intractable with respect to $\mathcal{R}$.*

Theorem 2.1 clearly implies that proving the sequential indifferentiability of $E^{\mathbf{P}}$ is sufficient to justify some form of resistance to chosen-key attacks.

2.3 Statistical Distance and the Coupling Technique

Let Ω be a finite event space and two probability distributions μ and ν are defined on Ω. The *statistical distance* (or total variation) between μ and ν, denoted by $\|\mu-\nu\|$, is defined as:

$$\|\mu-\nu\| := \frac{1}{2}\sum_{x\in\Omega}|\mu(x)-\nu(x)|\,.$$

The following formulations can be easily verified to be equivalent:

$$\|\mu-\nu\| = \max_{S\subset\Omega}\{\mu(S)-\nu(S)\} = \max_{S\subset\Omega}\{\nu(S)-\mu(S)\}$$

It is easy to verify that the statistical distance satisfies the symmetry and *triangle inequality*. Moreover, it is always lying between zero and one. It is one if and only if the support[2] of the probability distributions are disjoint and zero if and only if the distributions are the same.

A *coupling* of μ and ν is a distribution λ on $\Omega\times\Omega$ such that for all $x\in\Omega$, $\sum_{y\in\Omega}\lambda(x,y) = \mu(x)$ and for all $y\in\Omega$, $\sum_{x\in\Omega}\lambda(x,y) = \nu(y)$. In other words, λ is a joint distribution whose marginal distributions are resp. μ and ν. The following lemma is the main technical ingredient of the well-known coupling technique [Ald83]. A proof of this lemma is available in [LPS12], and restated here for completeness.

Lemma 2.1 (Coupling Lemma)**.** *Let μ and ν be probability distributions on a finite event space Ω. Let λ be a coupling of μ and ν, and let $(X,Y)\sim\lambda$ (i.e. (X,Y) is a random variable sampled according to distribution λ). Then $\|\mu-\nu\| \leq \Pr[X\neq Y]$.*

[2]The set of all elements having positive probability.

Proof. Let λ be the coupling of μ and ν, and $(X, Y) \sim \lambda$. By definition, we have that for any $z \in \Omega$, $\lambda(z, z) \leq \min\{\mu(z), \nu(z)\}$. Moreover, $\Pr[X = Y] = \sum_{z\in\Omega} \lambda(z, z)$. Hence we have:

$$\Pr[X = Y] \leq \sum_{z\in\Omega} \min\{\mu(z), \nu(z)\}.$$

Therefore:

$$\begin{aligned}
\Pr[X \neq Y] &\geq 1 - \sum_{z\in\Omega} \min\{\mu(z), \nu(z)\} \\
&= \sum_{z\in\Omega} (\mu(z) - \min\{\mu(z) - \nu(z)\}) \\
&= \sum_{\substack{z\in\Omega \\ \mu(z)\geq\nu(z)}} (\mu(z) - \nu(z)) \\
&= \max_{S\subset\Omega} \{\mu(S) - \nu(S)\} \\
&= \|\mu - \nu\|.
\end{aligned}$$

□

3 TEML: TEM with Linear Tweak-Key Mixing

Throughout, we fix $r \in \mathbb{N}$ as the number of rounds. In addition, we set $\eta = \alpha n$, $\kappa = \beta n$, and define $\theta := \alpha + \beta$.

Iterated Tweakable Even-Mansour. The r-round Tweakable Even-Mansour cipher is built on a tuple of r permutations $\mathbf{P} = (P_1, \ldots, P_r)$ of $\{0,1\}^n$ and a tuple of $(r+1)$ functions $\gamma = (\gamma_0, \ldots, \gamma_r)$ from $\{0,1\}^{\theta n} \to \{0,1\}^n$. It takes as input an θn-bit tweakey (k, t) and an n-bit block x, and outputs

$$\mathsf{TEM}^{\gamma,\mathbf{P}}_{k,t}(x) = P_r\left(P_{r-1}\left(\cdots P_1\left(x \oplus \gamma_0(k,t)\right)\cdots\right) \oplus \gamma_{r-1}(k,t)\right) \oplus \gamma_r(k,t), \tag{3}$$

The tuple γ is referred to as the tweakey schedule of the construction. When clear from the context, we will sometimes drop γ and $\mathbf{P}$ from the notation.

3.1 The TEML Construction

In [CS15a], Cogliati and Seurin provided the first result on Tweakable Even-Mansour with linear tweak and key mixing, henceforth referred as TEML. They proved beyond-the-birthday bound security for a 4-round TEM with $2n$-bit key and n-bit tweak., i.e., $\alpha = 1$ and $\beta = 2$.

In this paper, our goal is to generalize TEML for all $r \geq 1$ and $\alpha \geq 1$, i.e., we also consider tweaks larger than n bits. This is particularly the case for several TBCs based on the TWEAKEY framework. For instance, Skinny-128-384 can be used with 128-bit block and key size and 256-bit tweak size.

When γ is linear, then there exists a tuple of linear functions $\lambda = (\lambda_0, \ldots, \lambda_r)$ and $\delta = (\delta_0, \ldots, \delta_r)$, such that for all $i \in [\![0, r]\!]$

$$\gamma_i(k, t) = \lambda_i(k) \oplus \delta_i(t) \tag{4}$$

In other words, we can always view the key and tweak scheduling as separate linear functions, whenever the tweakey schedule is linear. We refer to λ and δ as the key and tweak schedule corresponding the tweakey schedule γ, respectively.

Ideally, one would want a minimal increase in the number of rounds on account of a larger tweak, to obtain similar security bounds as in the case of $\alpha = 1$. From (3), it is clear that an r-round TEM construction uses $r + 1$ round tweakeys. So, r must be at least

$\alpha - 1$, otherwise, it is easy to see that the adversary can choose two distinct tweaks t and t', such that $\delta_i(t) = \delta_i(t')$ for all $i \in [\![0, r]\!]$, resulting in a simple collision distinguisher. The case where $\alpha - 1 \leq r \leq \alpha$ does not fare well either. Specifically, the adversary can always choose distinct tweaks and block input pairs (t, x) and (t', x') such that $t \neq t'$, $\delta_0(t) \oplus x = \delta_0(t') \oplus x'$, and $\delta_i(t) = \delta_i(t')$ for all $i \in [\![1, r-1]\!]$ (since $r - 1 \leq \alpha$). Clearly, the XOR of the outputs corresponding to (t, x) and (t', x') equals $\delta_r(t) \oplus \delta_r(t')$.

The above discussion clearly shows that $r = \alpha + 1$ rounds are necessary to securely absorb an η-bit tweak using a linear tweak schedule, where $\eta = \alpha n$ denotes the tweak size in bits. However, just having $r > \alpha$ rounds is not sufficient for security. Indeed, one can come up with some pathological linear tweak(ey) schedule that makes the resulting construction completely insecure. For instance, assume $\alpha = 2$, and let $\delta_i(t_1, t_2) = t_1$ for all $i \in [\![0, r-1]\!]$ and $\delta_r(t_1, t_2) = t_2$. This tweak schedule is obviously insecure irrespective of the number of rounds. So, some care has to be taken while deciding on a tweak(ey) schedule.

In fact, similar concerns were raised in the core discussion behind the rationale of the STK construction in [JNP14]. Indeed, their main observation requires a one-to-one relation between the input tweakey (k, t) and any θ-subset of the $(r + 1)$ round tweakeys. Formally, we introduce the following definitions.

Definition 3.1 (Strong s-bijectivity)**.** Let $s \in \mathbb{N}$. A strong s-bijective schedule $\gamma := (\gamma_0, \ldots, \gamma_r)$ is a tuple of $r \geq s$ linear functions $\gamma_i : \{0,1\}^{sn} \to \{0,1\}^n$ such that for any s-subtuple, $\gamma' = (\gamma_{i_1}, \ldots, \gamma_{i_s})$ of γ, the mapping

$$(k, t) \mapsto (\gamma_{i_1}(k, t), \ldots, \gamma_{i_s}(k, t))$$

is a bijection.

Definition 3.2 (Weak s-bijectivity)**.** Let $s \in \mathbb{N}$. A weak s-bijective schedule $\gamma := (\gamma_0, \ldots, \gamma_r)$ is a tuple of $r \geq s$ linear functions $\gamma_i : \{0,1\}^{sn} \to \{0,1\}^n$ such that for any contiguous s-subtuple, $\gamma' = (\gamma_i, \ldots, \gamma_{i+s-1})$ of γ, the mapping

$$(k, t) \mapsto (\gamma_i(k, t), \ldots, \gamma_{i+s-1}(k, t))$$

is a bijection.

It is obvious to see that strong s-bijectivity implies weak s-bijectivity. However, the converse may not be true. By definition, a strong s-bijective schedule cannot collide on more than $(s - 1)$ round tweakeys for any two distinct tweaks. On the contrary, a weak s-bijective schedule only requires at least one distinct round tweakey for every consecutive s rounds. In the following results, we show that weak s-bijectivity of the public[3] part of the tweak(ey) schedule is sufficient for desired security with minimal number of rounds. In particular, we will not employ the strong bijectivity property in this paper.

3.2 IND-CCA Security of TEML

In the indistinguishability framework, the underlying key is secret. Additionally, it is quite common to consider independent and uniform at random keys at each round. We will also employ this assumption.

More specifically, we assume that the key is an $(r + 1)$ tuple $\mathbf{k} = (k_0, \ldots, k_r)$, where $k_i \xleftarrow{\$} \{0,1\}^n$, and k_i is independent of k_j, for all $i \neq j \in [\![0, r]\!]$. In addition, we take $\lambda_i(\mathbf{k}) = k_i$, i.e., we ignore the key schedule λ, and simply XOR the i-th component of $\mathbf{k}$ as the i-th round key. The following result establishes the IND-CCA security of r-TEML for any $r \geq 2$.

[3] In the indistinguishability setting, this is the tweak part of the tweakey, whereas in the indifferentiability setting the entire tweakey is controlled by the adversary.

Theorem 3.1 (IND-CCA Security)**.** *Let $r \geq \alpha + 1$ be an even integer and $r' = r/2$. Let $q_c, q_p, q_{\max}$ be positive integers such that $q_{\max} = \max\{q_c, q_p\}$ and $q_c + q_p < N/2$. Then, for any weak α-bijective tweak schedule δ, we have*

$$\mathbf{Adv}^{\mathsf{ind\text{-}cca}}_{r\text{-}\mathsf{TEML}^{\delta,\mathbf{P}}}(q_c, q_p) \leq \sqrt{2^{4+3r'} q_c \left(\frac{2q_{\max}}{N}\right)^{\left\lceil \frac{r'}{\alpha} \right\rceil - 1}}.$$

For odd $r \geq 3$, we have:

$$\mathbf{Adv}^{\mathsf{ind\text{-}cca}}_{r\text{-}\mathsf{TEML}^{\delta,\mathbf{P}}}(q_c, q_p) \leq \mathbf{Adv}^{\mathsf{ind\text{-}cca}}_{(r-1)\text{-}\mathsf{TEML}^{\delta,\mathbf{P}}}(q_c, q_p).$$

More concretely, r-TEML achieves IND-CCA security up to $O(N^{\frac{r-2\alpha}{r}})$ queries. Note that, we can use the bound of Theorem 3.1 for both r and $r-1$ (if r is odd). Hence, in the following, we always assume that r is even.

3.3 Sequential Indifferentiability of TEML

In the sequential indifferentiability setting, we are concerned with resistance against chosen-key attacks. In this case, since the adversary will always be allowed to choose its own keys, there will be functionally no difference between the tweak and key bits. In other words, the full tweakey is public and controlled by the adversary. Consequently, we will need weak bijectivity property for the entire tweakey input. In the following results we take the tweakey size to be rn bits.

We provide two results in this direction. We start off with a simple attack (see Lemma 3.1) on $(r+2)$-TEML with a r-bijective tweakey schedule δ. This clearly establishes that $r+3$ rounds are necessary for security.

Lemma 3.1 (Seq. Indiff. Attack on $r+2$ Rounds)**.** *For any efficient simulator Sim making at most σ oracle queries to the ideal cipher Π, there exists a sequential distinguisher $\mathcal{D}$ with at most $2r+6$ total query cost such that:*

$$\left|\Pr\left[\mathcal{D}^{\Pi,\mathsf{Sim}^{\Pi}} = 1\right] - \Pr\left[\mathcal{D}^{\mathit{TEML}^{\delta,\mathbf{P}},\mathbf{P}} = 1\right]\right| \geq 1 - \frac{1}{N-1} - \frac{q'^4}{2N},$$

where $q' = 2r + \sigma + 6$ is the total calls to Π from $\mathcal{D}$ and Sim combined.

The proof of this lemma mostly follows the strategy used in a similar attack on Even-Mansour cipher [CS15a]. For completeness, we provide the proof in section 5.

Next, in Theorem 3.2, we show that $r+3$ rounds are also sufficient for sequential indifferentiability.

Theorem 3.2 (Sequential Indifferentiability)**.** *Let $q, \sigma, t \in \mathbb{N}, \varepsilon \in [0,1]$. Suppose $q^{r+1} \leq N/4$. Then, the $(r+3)$-round TEML construction with a weak r-bijective tweakey schedule γ is $(q, \sigma, \mathtt{T}, \varepsilon)$-sequentially indifferentiable from an ideal cipher, where $\sigma = q^{r+1}$, $\mathtt{T} = \mathcal{O}(q^{r+1})$, and*

$$\mathbf{Adv}^{\mathsf{seq\text{-}indiff}}_{r\text{-}\mathsf{TEML}^{\gamma,\mathbf{P}}}(q, \sigma, \mathtt{T}) \leq \varepsilon = \frac{\left((r+5)^2 + 32\right) q^{2r+2}}{N}.$$

4 Proof of IND-CCA Security of TEML

Fix a computationally unbounded and deterministic adversary $\mathcal{A}$ that maximizes the advantage. Let $\mathcal{T} = \{0,1\}^{\eta}$. Given a tuple $\mathbf{t} = (t_1, \ldots, t_{q_c}) \in \mathcal{T}^{q_c}$, we will write

$\Omega_{\mathbf{t}} \subset (\{0,1\}^n)^{q_c}$ to denote the set of all possible inputs $\mathbf{x} = (x_1, \ldots, x_{q_c}) \in (\{0,1\}^n)^{q_c}$ such that all pairs (t_i, x_i) are pairwise distinct, i.e.,

$$\Omega_{\mathbf{t}} := \{\mathbf{x} := (x_1, \ldots, x_{q_c}) \in (\{0,1\}^n)^{q_c} : \forall i \neq j, (x_i, t_i) \neq (x_j, t_j)\}.$$

Query Transcript. The interaction of $\mathcal{A}$ with its oracles can be summarized in a *query transcript* $(\mathcal{Q}_C, \mathcal{Q}_{P_1}, \ldots, \mathcal{Q}_{P_r})$ of the attack. Here $\mathcal{Q}_C$ records the queries to the construction oracle which contains all triples $(t, x, y) \in \mathcal{T} \times \{0,1\}^n \times \{0,1\}^n$ such that $\mathcal{A}$ either made the direct query (t, x) to the construction oracle and received answer y, or made the inverse query (t, y) and received answer x. Similarly, for each $i \in [\![1, r]\!]$, $\mathcal{Q}_{P_i}$ contains the queries to the round permutation P_i in the form of pairs $(u, v) \in \{0,1\}^n \times \{0,1\}^n$ such that $\mathcal{A}$ either made the direct query u to permutation P_i and received answer v, or made the inverse query v and received answer u. Note that the queries are recorded in a directionless and unordered fashion, but by our assumption that $\mathcal{A}$ is deterministic, there is a one-to-one mapping between this representation and the raw transcript of the interaction of $\mathcal{A}$ with oracles (see [CS14][CLS15] for more details). Also, note that by our assumption $\mathcal{A}$ never makes pointless queries. So, each query to the construction oracle results in a distinct triple in $\mathcal{Q}_C$, and each query to P_i results in a distinct pair in $\mathcal{Q}_{P_i}$, so that $|\mathcal{Q}_C| = q_c$ and $|\mathcal{Q}_{P_i}| = q_p$ for each $i \in [\![1, r]\!]$ since we assume that $\mathcal{A}$ always makes the maximal number of allowed queries to each oracle. Let m denote the number of distinct tweaks appearing in $\mathcal{Q}_C$, and q_i the number of queries for the i-th tweak, $1 \leq i \leq m$, using an arbitrary ordering of tweaks. Note that m may depend on the answer received from the oracles, yet we have $\sum_{i=1}^{m} q_i = q_c$.

Let $\tau' = (\mathcal{Q}_C, \mathcal{Q}_{P_1}, \ldots, \mathcal{Q}_{P_r})$ be the resulting transcript. We say that τ' is *attainable* (with respect to some fixed adversary $\mathcal{A}$) if the probability to realize τ' in an interaction of $\mathcal{A}$ with $(\widetilde{\Pi}, \mathbf{P})$ (the ideal world) is non-zero. Let Θ denote the set of all attainable transcripts. We denote by μ_{re} (resp. μ_{id}), the probability distribution of the transcript induced in the real world (resp. the ideal world). Note that these two probability distributions depend on the adversary. By a slight abuse of notations, we reuse the same notations to denote the random variables distributed according to these distributions.

Given a permutation queries transcript $\mathcal{Q}$ and a permutation P, $P \vdash \mathcal{Q}$ (referred as P *extends* $\mathcal{Q}$) denotes the event $P(u) = v$ for all $(u, v) \in \mathcal{Q}$. By extension, given a tuple of permutation queries transcript $\mathcal{Q}_{\mathbf{P}} = (\mathcal{Q}_{P_1}, \ldots, \mathcal{Q}_{P_r})$ and a tuple of permutations $\mathbf{P} = (P_1, \ldots, P_r)$, $\mathbf{P} \vdash \mathcal{Q}_{\mathbf{P}}$ (referred as $\mathbf{P}$ *extends* $\mathcal{Q}_{\mathbf{P}}$) denotes the event $\wedge_{i=1}^{r}(P_i \vdash \mathcal{Q}_{P_i})$. Note that for a permutation transcript of size q_p, we have:

$$\Pr\left[P \leftarrow_{\$} \mathsf{P}(n) : P \vdash \mathcal{Q}\right] = \frac{1}{(N)_{q_p}}.$$

Also, it follows from the above fact that

$$\Pr\left[\mathbf{P} \leftarrow_{\$} (\mathsf{P}(n))^r : \mathbf{P} \vdash \mathcal{Q}_{\mathbf{P}}\right] = \frac{1}{\left((N)_{q_p}\right)^r},$$

as the permutations $\mathbf{P} = (P_1, \ldots, P_{r'})$ are uniformly random and independent.

Similarly, given a tweakable permutation transcript $\widetilde{\mathcal{Q}}$ and a tweakable permutation $\widetilde{P}$, we say that $\widetilde{P} \vdash \widetilde{\mathcal{Q}}$, if $\widetilde{P}(t, x) = y$ for all $(t, x, y) \in \widetilde{\mathcal{Q}}$. For a tweakable permutation transcript $\widetilde{\mathcal{Q}}$ with m distinct tweaks and q_i queries corresponding to the i-th tweak, we have:

$$\Pr\left[\widetilde{P} \leftarrow_{\$} \widetilde{\mathsf{P}}(\eta, n) : \widetilde{P} \vdash \widetilde{\mathcal{Q}}\right] = \prod_{i=1}^{m} \frac{1}{(N)_{q_i}}.$$

So, the probability of getting any attainable transcript $\tau' = (\mathcal{Q}_C, \mathcal{Q}_{\mathbf{P}})$ in the ideal world is

$$\Pr\left[\mu_{\mathsf{id}} = \tau'\right] = \left(\frac{1}{(N)_{q_p}}\right)^r \times \prod_{i=1}^{m} \frac{1}{(N)_{q_i}}.$$

In the real world, the probability to obtain τ' is

$$\Pr\left[\mu_{\mathsf{re}} = \tau'\right] = \left(\frac{1}{(N)_{q_p}}\right)^r \times p(\tau'),$$

where $p(\tau') := \Pr\left[\mathbf{P} \leftarrow_{\$} (\mathsf{P}(n))^r : \mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}} \vdash \mathcal{Q}_C \mid \mathbf{P} \vdash \mathcal{Q}_{\mathbf{P}}\right]$.

Proof Overview. Let us fix an IND-CCA-distinguisher $\mathcal{A}$ against the r-TEML construction. We start by recalling the H-coefficient technique [Pat08].

Lemma 4.1. *[Pat08, CS14] Let $\Theta = \Theta_{\mathsf{good}} \sqcup \Theta_{\mathsf{bad}}$ be a partition of the set of attainable transcripts. Assume that there exists $\varepsilon \geq 0$ such that, for any $\tau' \in \Theta_{\mathsf{good}}$, one has*

$$\frac{\Pr\left[\mu_{\mathsf{re}} = \tau'\right]}{\Pr\left[\mu_{\mathsf{id}} = \tau'\right]} \geq 1 - \varepsilon.$$

Then

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{r'\text{-}\mathsf{TEML}}(\mathcal{A}) \leq \Pr\left[\mu_{\mathsf{id}} \in \Theta_{\mathsf{bad}}\right] + \varepsilon.$$

Our goal is to apply Lemma 4.1 with $\Theta_{\mathsf{bad}} = \emptyset$. In order to do so, we have to lower bound the ratio of the probabilities of observing any given attainable transcript in both the worlds. We start by dividing the r-TEML construction into two r'-TEML constructions as follows. For any $\mathbf{k} = (k_0, \ldots, k_r) \in \{0,1\}^{(r+1)n}$, and tweak schedule $\delta = (\delta_0, \ldots, \delta_r)$, any permutation tuple $\mathbf{P} = (P_1, \ldots, P_r)$, any $t \in \{0,1\}^\eta$, and any $x \in \{0,1\}^n$, one has

$$r\text{-}\mathsf{TEML}^{\delta,\mathbf{P}}_{\mathbf{k}}(t,x) = \left(r'\text{-}\mathsf{TEML}^{\delta^2\mathbf{P}_2}_{\mathbf{k}_2}\right)^{-1}\left(t, r'\text{-}\mathsf{TEML}^{\delta^1,\mathbf{P}_1}_{\mathbf{k}_1}(t,x) \oplus \delta_{r'}(t)\right),$$

where

$$\begin{aligned}
&\mathbf{P}_1 = (P_1, \ldots, P_{r'}), \mathbf{P}_2 = (P_r, \ldots, P_{r'+1}),\\
&\mathbf{k}_1 = (k_0, \ldots, k_{r'-1}, k_{r'} \oplus k'), \mathbf{k}_2 = (k_r, \ldots, k_{r'+1}, k'),\\
&\delta^1 = (\delta_0, \ldots, \delta_{r'}), \delta^2 = (\delta_r, \ldots, \delta_{r'+1}),
\end{aligned}$$

for any $k' \in \{0,1\}^n$. Hence, the r-TEML construction with uniformly random keys and round permutations can be seen as the composition (up to a shift) of two independent instances of the r'-TEML construction, also with uniformly random keys and round permutations.

The crucial point of our proof will be to upper bound the statistical distance between the distribution of the outputs of r'-TEML *conditioned on partial information on the permutations* (namely $P_i \vdash \mathcal{Q}_{P_i}$ for $i = 1, \ldots, r'$) and the uniform distribution on $\Omega_{\mathbf{t}}$.

Definition 4.1. Fix $\mathbf{t} = (t_1, \ldots, t_{q_c})$ and $\mathbf{x} = (x_1, \ldots, x_{q_c}) \in \Omega_{\mathbf{t}}$. We denote $\mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_{\mathbf{P}}}$ the distribution of the tuple

$$r'\text{-}\mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}}(\mathbf{t},\mathbf{x}) := \left(r'\text{-}\mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}}(t_1,x_1), \ldots, r'\text{-}\mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}}(t_{q_c},x_{q_c})\right)$$

conditioned on the event $\mathbf{P} \vdash \mathcal{Q}_{\mathbf{P}}$ (i.e. when the key $\mathbf{k} = (k_0, \ldots, k_{r'})$ is uniformly random and the permutation $\mathbf{P} = (P_1, \ldots, P'_r)$ are uniformly random among permutation satisfying $\wedge_{i=1}^{r'}(P_i \vdash \mathcal{Q}_{P_i})$). We denote $\mu^*_{\mathbf{t}}$ the uniform distribution on $\Omega_{\mathbf{t}}$.

The following lemma, establishing an appropriate upper bound for $\|\mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_{\mathbf{P}}} - \mu^*_{\mathbf{t}}\|$ is the main technical contribution of this paper.

Lemma 4.2. *Let $q_c, q_p \in \mathbb{N}$ such that $q_c + q_p < N/2$, and $q_{\max} = \max\{q_c, q_p\}$. Fix any attainable permutation transcript $\mathcal{Q}_{\mathbf{P}}$ and any $\mathbf{t} \in \mathcal{T}^{q_c}$, $\mathbf{x} \in \Omega_{\mathbf{t}}$. Then, we have*

$$\|\mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_P} - \mu^*_{\mathbf{t}}\| \leq 8^{r'} q_c \left(\frac{2q_{\max}}{N}\right)^{\left\lceil \frac{r'}{\alpha} \right\rceil - 1}.$$

Proof. Fix any attainable permutation queries transcript $\mathcal{Q}_{\mathbf{P}} = (\mathcal{Q}_{P_1}, \ldots, \mathcal{Q}_{P_{r'}})$ and $\mathbf{t} = (t_1, \ldots, t_{q_c}) \in \mathcal{T}^{q_c}$, $\mathbf{x} = (x_1, \ldots, x_{q_c}) \in \Omega_{\mathbf{t}}$. Our main task is to upper bound $\|\mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_{\mathbf{P}}} - \mu^*_{\mathbf{t}}\|$.

We can split the computation of $\|\mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_{\mathbf{P}}} - \mu^*_{\mathbf{t}}\|$ into q_c simpler computations. The idea is to construct a distribution ν_l for every $l \in [\![0, q_c]\!]$ such that ν_l is the distribution of the outputs of a random instance of r'-$\mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}}$ queried with (t_i, x_i) for $i \in [\![1, l]\!]$ and the last $q_c - l$ queries keep the same tweak t_i as in adversarial queries, but their block inputs z_i are chosen uniformly at random among the values that were not queried. More precisely, for each $l \in [\![0, q_c]\!]$, let $\mathbf{z} = (z_1, \ldots, z_{q_c})$ be a tuple of queries such that:

$$\begin{cases} z_i = x_i, \forall i \in [\![1, l]\!], \\ z_i \leftarrow_{\$} \{0,1\}^n \setminus \{z_j | t_j = t_i, j < i\}, \forall i > l. \end{cases}$$

This means that the first l queries are the adversary's queries and the remaining z_i are chosen uniformly at random among all the possible values (all queries have to be pairwise distinct). Denote ν_l the distribution of r'-$\mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}}(\mathbf{t}, \mathbf{z})$, conditioned on $\mathbf{P} \vdash \mathcal{Q}_{\mathbf{P}}$. Hence we have:

$$\|\mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_{\mathbf{P}}} - \mu^*_{\mathbf{t}}\| = \|\nu_{q_c} - \nu_0\| \leq \sum_{l=0}^{q_c - 1} \|\nu_{l+1} - \nu_l\|. \tag{5}$$

Note that for $l = q_c$, $z_i = x_i$, $\forall i \in [\![1, q_c]\!]$ and hence r'-$\mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}}(\mathbf{t}, \mathbf{z}) = r'$-$\mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}}(\mathbf{t}, \mathbf{x})$ leads to $\nu_{q_c} = \mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_{\mathbf{P}}}$. It is easy to see that ν_0 is identical to $\mu^*_{\mathbf{t}}$. In Lemma 4.3, we upper bound the total variation distance $\|\nu_{l+1} - \nu_l\|$. The proof of this lemma is deferred to section 4.1.

Lemma 4.3 (Hybrids-Distance)**.** *Let $q_c, q_p \in \mathbb{N}$ such that $q_c + q_p < N/2$ and $q_{\max} = \max\{q_c, q_p\}$. For any $l \in [\![0, q_c - 1]\!]$, we have*

$$\|\nu_{l+1} - \nu_l\| \leq 8^{r'} \left(\frac{2q_{\max}}{N}\right)^{\left\lceil \frac{r'}{\alpha} \right\rceil - 1}.$$

The proof of Lemma 4.2 follows from (5) and Lemma 4.3. □

Concluding the proof of Theorem 3.1. The final result in Theorem 3.1 can be obtained by relying on the following composition lemma, whose proof is identical to [CLS15, Lemma 11].

Lemma 4.4. *Let r be an even integer and $r' = r/2$. Let $q_c, q_p \in \mathbb{N}$ and $q_{\max} = \max\{q_c, q_p\}$. Assume that there exists an ε such that, for any attainable queries transcript $\mathcal{Q}_{\mathbf{P}}$ and every $\mathbf{t} \in \mathcal{T}^{q_c}, \mathbf{x} \in \Omega_{\mathbf{t}}$, we have*

$$\|\mu_{\mathbf{t},\mathbf{x},\mathcal{Q}_{\mathbf{P}}} - \mu^*_{\mathbf{t}}\| \leq \varepsilon.$$

Then, for any attainable transcript queries τ', one has

$$\Pr[\mu_{\mathsf{re}} = \tau'] \geq (1 - 4\sqrt{\varepsilon}) \Pr[\mu_{\mathsf{id}} = \tau'], \tag{6}$$

The proof of Theorem 3.1 follows from Lemmata 4.1, 4.2 and 4.4.

4.1 Proof of Hybrids-Distance Lemma

To prove Lemma 4.3, it remains to upper bound the total variation distance between ν_{l+1} and ν_l, for each $l \in [\![0, q_c - 1]\!]$. In this section, since we are considering a single instance of r'-TEML, we will drop the number of rounds r' and simply denote it by TEML in order to lighten the notation.

Note that we only have to consider the first $l+1$ elements of the two tuples of outputs since for both distributions, the i-th input for $i > l+1$ is sampled at random. In other words,

$$\|\nu_{l+1} - \nu_l\| = \left\|\nu'_{l+1} - \nu'_l\right\|, \tag{7}$$

where ν'_{l+1} and ν'_l are the respective distributions of the $l+1$ first outputs of the cipher as defined in section 3. We will construct a suitable coupling of the two distributions, ν'_{l+1} and ν'_l, and apply the coupling lemma (see Lemma 2.1) to bound the distance.

Coupling of ν'_{l+1} and ν'_l. To define the coupling of ν'_{l+1} and ν'_l, we consider the tweakable Even-Mansour cipher $\mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}}$, where $\mathbf{P}$ satisfies $\mathbf{P} \vdash \mathcal{Q}_{\mathbf{P}}$. Namely, where the key $\mathbf{k} = (k_1, \ldots, k_{r'})$ is uniformly random and the permutations $\mathbf{P} = (P_1, \ldots, P_{r'})$ are uniformly random among permutation satisfying $\wedge_{i=1}^{r'}(P_i \vdash \mathcal{Q}_{P_i})$). It receives inputs $\mathbf{x}' = (x_1, \ldots, x_{l+1})$ and $\mathbf{t}' = (t_1, \ldots, t_{l+1})$, so that $\mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}}(\mathbf{t}', \mathbf{x}')$ is distributed according to ν'_{l+1}.

We will now construct a second tweakable Even-Mansour cipher $\mathsf{TEML}^{\mathbf{P}'}_{\mathbf{k}'}$ with inputs $\mathbf{z}' = (z_1, \ldots, z_{l+1})$ and $\mathbf{t}' = (t_1, \ldots, t_{l+1})$, satisfying the following properties:

- Property I: $\mathbf{P}' = (P'_1, \ldots, P'_{r'})$ are uniformly random among permutation tuples satisfying $\mathbf{P}' \vdash \mathcal{Q}_{\mathbf{P}}$ and $\mathbf{k}'$ is uniformly random.
- Property II: $z_i = x_i$ for every $i \in [\![1, l]\!]$, and z_{l+1} is uniformly random in $\{0,1\}^n \setminus \{x_j | t_j = t_{l+1}, j < l+1\}$;
- Property III: for each $i \in [\![1, l+1]\!]$, if the outputs of the j-th round permutation in the computations of $\mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}}(t_i, x_i)$ and $\mathsf{TEML}^{\mathbf{P}'}_{\mathbf{k}'}(t_i, z_i)$ are equal, then this also holds for all subsequent inner permutations.

Note that, the same tweaks are used for both ciphers. So, Property I and II will ensure that $\mathsf{TEML}^{\mathbf{P}'}_{\mathbf{k}'}(\mathbf{t}', \mathbf{z}')$ is distributed according to ν'_l.

For $i \in [\![1, r']\!]$, we denote:

$$U_i = \{u_i | (u_i, v_i) \in \mathcal{Q}_{P_i}\},$$
$$V_i = \{v_i | (u_i, v_i) \in \mathcal{Q}_{P_i}\}.$$

For $i \in [\![1, l+1]\!]$ and $j \in [\![1, r']\!]$, we also define x_i^j (resp. y_i^j) as the output (rep. input) of the j-th round permutation, P_j when computing $\mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}}(t_i, x_i)$, and similarly z_i^j (resp. w_i^j) as the output (rep. input) of the j-th round permutation, P'_j when computing $\mathsf{TEML}^{\mathbf{P}'}_{\mathbf{k}'}(t_i, z_i)$, i.e.,

$$\begin{cases} x_i^0 = x_i \\ z_i^0 = z_i \\ y_i^j = x_i^{j-1} \oplus k_j \oplus \delta_j(\mathbf{t}_i) \\ w_i^j = z_i^{j-1} \oplus k'_j \oplus \delta_j(\mathbf{t}_i) \\ x_i^j = P_j(y_i^j) \\ z_i^j = P'_j(w_i^j). \end{cases} \tag{8}$$

We refer to $\tau = ((x_i^j, y_i^j, z_i^j, w_i^j, \mathbf{k}, \mathbf{t}, U_j) : \ i \in [\![1, l+1]\!], \ j \in [\![1, r']\!], \ \mathbf{k} = (k_1, \ldots, k_{r'}), \ \mathbf{t} = (t_1, \ldots, t_{l+1})))$, as an extension of the transcript $(\mathcal{Q}_{\boldsymbol{C}}, \mathcal{Q}_{\mathbf{P}})$, and call it the *view* of τ. Note

that for a view, we must have

$$\left(x_i^j = x_{i'}^j\right) \iff \left(y_i^j = y_{i'}^j\right), \left(z_i^j = z_{i'}^j\right) \iff \left(w_i^j = w_{i'}^j\right).$$

In order to apply the coupling lemma, we have to find how to correlate $(\mathbf{P}, \mathbf{k})$ and $(\mathbf{P'}, \mathbf{k'})$ so that the outputs, $\left(x_1^{r'}, \ldots, x_{l+1}^{r'}\right)$ and $\left(z_1^{r'}, \ldots, z_{l+1}^{r'}\right)$, are equal with high probability. We choose $(\mathbf{P}, \mathbf{k})$ uniformly at random and we construct $(\mathbf{P'}, \mathbf{k'})$ as a function of $(\mathbf{P}, \mathbf{k})$, i.e., $(\mathbf{P'}, \mathbf{k'})$ will not be independent from $(\mathbf{P}, \mathbf{k})$. The only requirement is that both $(\mathbf{P}, \mathbf{k})$ and $(\mathbf{P'}, \mathbf{k'})$ have the correct marginal distributions. We have to pay attention that the distribution of $(\mathbf{P'}, \mathbf{k'})$ remains uniform in order for $\left(z_1^{r'}, \ldots, z_{l+1}^{r'}\right)$ to be distributed according to ν_l'.
We now describe how $(\mathbf{P'}, \mathbf{k'})$ is constructed using $(\mathbf{P}, \mathbf{k})$. First, choose the key $\mathbf{k'} = \mathbf{k} = (k_1, \ldots, k_r)$. Next, we want both tuples $\mathbf{P}$ and $\mathbf{P'}$ to agree on the permutation queries, i.e., for any $i \in [\![1, r']\!]$ and $(y, x) \in \mathcal{Q}_{P_i}$, we want $P_i'(y) = x$. Moreover, in order to obtain Property III, we will want that for every $(i, j) \in [\![1, l]\!] \times [\![1, r']\!]$, $z_i^j = x_i^j$ and $w_i^j = y_i^j$.

For the $(l+1)$-th query, we will try to make the outputs of the two corresponding permutations equal, at some round j, as long as it does not interfere with the previous rules, i.e., Property I-III. If it succeeds, by Property III, the outputs of all the subsequent round permutations must be equal. Formally, we describe the following sampling.

Coupling the first l queries: For every $i \in [\![1, l]\!]$, the i-th queries x_i^0 and z_i^0 are equal by definition. Considering the system (8), we set $P_j'\left(w_i^j\right) = P_j'\left(y_i^j\right) = P_j\left(y_i^j\right)$ for every $i \in [\![1, l]\!]$ and $j \in [\![1, r']\!]$. This implies that the first l outputs $\left(x_1^{r'}, \ldots, x_l^{r'}\right)$ and $\left(z_1^{r'}, \ldots, z_l^{r'}\right)$ are equal.

Coupling the $(l+1)$-th query: For every $j \in [\![1, r']\!]$ we define the coupling for the $(l+1)$-th query as follows:

Rule (1) : If $w_{l+1}^j \in U_j$ or there exists $i \in [\![1, l]\!]$ such that $w_{l+1}^j = w_i^j = y_i^j$, then $z_{l+1}^j = P_j'\left(w_{l+1}^j\right)$ is already determined; unless we have coupled z_{l+1}^{j-1} and x_{l+1}^{j-1} in a previous round, we cannot couple z_{l+1}^j and x_{l+1}^j at this round.
Rule (2) : else, if $w_{l+1}^j \notin U_j$ and $w_{l+1}^j \neq w_i^j$ for all $i \in [\![1, l]\!]$, then;

(a) If $y_{l+1}^j \in U_j$ or there exists $i \in [\![1, l]\!]$ such that $y_{l+1}^j = y_i^j$, then we choose $z_{l+1}^j = P_j'\left(w_{l+1}^j\right)$ uniformly at random in $\{0,1\}^n \setminus (V_j \cup \{P_j'\left(w_i^j\right), i \leq l\})$, and, so we cannot couple z_{l+1}^j and x_{l+1}^j at this round.
(b) else, we define $P_j'\left(w_{l+1}^j\right) = P_j\left(y_{l+1}^j\right)$. This implies that $z_{l+1}^j = x_{l+1}^j$.

Note that, for the first l construction queries we define $\mathbf{P'}$ to be exactly same as $\mathbf{P}$ and for the $(l+1)$-th query we have defined $\mathbf{P'}$ by the above rules. Hence, using the fact that the keys and the tweaks are the same for both the ciphers, we can conclude that Property III is satisfied. So, once $z_{l+1}^j = x_{l+1}^j$, we must have $z_{l+1}^{j'} = x_{l+1}^{j'}$ for any subsequent round $j' \geq j$. In particular, for $j' = r'$, $z_{l+1}^{r'} = x_{l+1}^{r'}$. So, the coupling succeeds.

Uniformly random $(\mathbf{P'}, \mathbf{k'})$. Since $\mathbf{k'} = \mathbf{k}$ and $\mathbf{k}$ is uniformly random, $\mathbf{k'}$ is also uniformly random. During the coupling of the first l queries , we set $P_j'\left(w_i^j\right) = P_j\left(y_i^j\right)$ for every $i \in [\![1, l]\!]$, $j \in [\![1, r']\!]$ and $P_j\left(y_i^j\right)$ is uniformly random among possible values, thus $P_j'\left(w_i^j\right)$ is uniformly random among possible values.

Rule (1) says that if there is a collision with a previous input of P'_j, we cannot choose the value of $P'_j(w^j_{l+1})$ so this does not change anything to the distribution of P'_j.
When the conditions of Rule (2)(a) are met, we have:

– for some $i \in [\![1, l]\!]$:

$$\begin{cases} P_j\left(y^j_{l+1}\right) = P_j\left(y^j_i\right) = P'_j\left(w^j_i\right) \\ w^j_{l+1} \neq w^j_i, \end{cases}$$

– or for some $(u_j, v_j) \in \mathcal{Q}_{P_j}$:

$$\begin{cases} P_j\left(y^j_{l+1}\right) = P_j\left(u_j\right) = P'_j\left(u_j\right) \\ w^j_{l+1} \neq u_j. \end{cases}$$

The two cases imply that $P'_j\left(w^j_{l+1}\right)$ is chosen uniformly random among possible values to keep P'_j uniformly distributed and distinct from $P'_j\left(w^j_i\right)$.
Finally, in Rule (2)(b), both $P_j(y^j_{l+1})$ and $P'_j\left(w^j_{l+1}\right)$ are set to a uniformly at random chosen value excluding V_j.
In conclusion, the permutations P'_j are uniformly random and independent as desired, whence $\left(z^{r'}_1, \ldots, z^{r'}_{l+1}\right)$ is distributed according to ν'_l.

This justifies Property I. Hence, the joint distribution,

$$\left(\mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}}\left(\mathbf{t}', \mathbf{x}'\right), \mathsf{TEML}^{\mathbf{P}'}_{\mathbf{k}'}\left(\mathbf{t}', \mathbf{z}'\right)\right),$$

is created in such a way that the marginal distribution $\mathsf{TEML}^{\mathbf{P}}_{\mathbf{k}}(\mathbf{t}', \mathbf{x}')$ and $\mathsf{TEML}^{\mathbf{P}'}_{\mathbf{k}'}(\mathbf{t}', \mathbf{z}')$ are ν'_{l+1} and ν'_l, respectively.
We can now apply Lemma 2.1 to obtain:

$$\left\|\nu'_{l+1} - \nu'_l\right\| \leq \mathsf{Pr}\left[\left(z^{r'}_1, \ldots, z^{r'}_{l+1}\right) \neq \left(x^{r'}_1, \ldots, x^{r'}_{l+1}\right)\right]. \tag{9}$$

Probability of Failure in Coupling. From (9), it remains to upper bound the probability that the coupling fails, i.e., to upper bound

$$\mathsf{Pr}\left[\left(z^{r'}_1, \ldots, z^{r'}_{l+1}\right) \neq \left(x^{r'}_1, \ldots, x^{r'}_{l+1}\right)\right].$$

Let `fail` be the event $\left(z^{r'}_1, \ldots, z^{r'}_{l+1}\right) \neq \left(x^{r'}_1, \ldots, x^{r'}_{l+1}\right)$, and notice that since $z^{r'}_i = x^{r'}_i$ for all $i \in [\![1, l]\!]$, we have:

$$\mathsf{Pr}\left[\left(z^{r'}_1, \ldots, z^{r'}_{l+1}\right) \neq \left(x^{r'}_1, \ldots, x^{r'}_{l+1}\right)\right] \leq \mathsf{Pr}\left[\left(z^{r'}_{l+1}\right) \neq \left(x^{r'}_{l+1}\right)\right]. \tag{10}$$

Hence, we are left with the task to upper bound the probability that $x^{r'}_{l+1} \neq z^{r'}_{l+1}$. In earlier works [LPS12, LS13b, CLS15], this is done by analyzing each round independently. However, in our case, this approach seems loose. Instead of bounding the probability locally at each round, we consider the global event of failure for all the rounds at once. We briefly discuss the motivation behind this change.

Consider the following collision events on round $j \in [\![1, r']\!]$,

$$\mathcal{F}^j_0 = (y^j_{l+1} \in U_j), \mathcal{F}^j_2 = (w^j_{l+1} \in U_j),$$

$$\mathcal{F}_1^j = \left(\exists i \in [\![1, r']\!] : y_{l+1}^j = y_i^j\right), \mathcal{F}_3^j = \left(\exists i \in [\![1, r']\!] : w_{l+1}^j = w_i^j\right).$$

Then, it is easy to see that, if $x_{l+1}^{r'} \neq z_{l+1}^{r'}$ then each round must have incurred a collision event. More precisely,

$$\texttt{fail} \subseteq \bigcap_{j=1}^{r'} \left(\mathcal{F}_0^j \cup \mathcal{F}_1^j \cup \mathcal{F}_1^j \cup \mathcal{F}_3^j\right) := Col(\tau)$$

From now on we say τ leads to a coupling failure if and only if $Col(\tau)$ occurs.

Fix a round $j \in [\![1, r']\!]$ and suppose $\mathcal{F}_2^j \cup \mathcal{F}_3^j$ occurs, then without loss of generality, there exists $i \in [\![1, l]\!]$, such that $y_{l+1}^j = y_i^j$. In the proof of r-TEM$[\mathcal{H}]$ [CLS15] and CLRW [LS13b], this gives an equation of the form

$$\mathcal{H}_{k_j}(t_{l+1}) \oplus \mathcal{H}_{k_j}(t_i) = x_{l+1}^{j-1} \oplus x_i^{j-1},$$

where $\mathcal{H}$ is an AXU-hash function. Since, $t_{l+1} \neq t_i$, the event can be easily bounded by the AXU[4] property of H.

However, in our case the same event gives rise to the following equation:

$$x_{l+1}^{j-1} \oplus \delta_j(\mathbf{t}_{l+1}) \oplus k_j = x_i^{j-1} \oplus \delta_j(\mathbf{t}_i) \oplus k_j. \tag{11}$$

First notice that the key k_j cancels out. As a result, we no longer have the AXU property. Next, if $\delta_j(\mathbf{t}_{l+1} \oplus \mathbf{t}_i) = 0$ then we must have $x_{l+1}^{j-1} = x_i^{j-1}$, i.e., the current collision is implied by a similar collision in the previous round. However, there can be at most $\alpha - 1$ consecutive such collisions, otherwise this would violate the α-bijectivity property of δ. Now, assume that $\delta_j(\mathbf{t}_{l+1} \oplus \mathbf{t}_i) \neq 0$. We consider the randomness of the permutation P_{j-1}. If at least one of the events $y_{l+1}^{j-1} \notin U_{j-1}$ or $y_i^{j-1} \notin U_{j-1}$ holds, then we can simply use the randomness of P_{j-1}, since this value is not known to the adversary. When $y_{l+1}^{j-1}, y_i^{j-1} \in U_{j-1}$, these outputs are already revealed to the adversary, whence we cannot use the randomness of P_{j-1} on these inputs. However, $y_{l+1}^{j-1}, y_i^{j-1} \in U_{j-1}$ is still an event over the randomness of the round key k_{j-1}. Therefore, it holds with probability at most $q_p N^{-1}$. Hence, the predicate $\texttt{YY}_j$ holds with probability at most $(q_p \cdot q_c) N^{-1}$. Looking ahead momentarily, this is far from a desirable upper bound.

Interestingly, we can actually extend this same argument to previous rounds until we reach a round $j' < j$, where $y_{l+1}^{j'} \notin U_{j'}$ or $y_i^{j'} \notin U_{j'}$, at which point we can terminate the argument. Considering such an extension might actually be useful in getting a better bound for $\mathcal{F}_2^j$ (equivalently $\mathcal{F}_3^j$). Note that the argument above creates a chain structure for calculating the probability that $\mathcal{F}_2^j$ holds (equivalently $\mathcal{F}_3^j$), it starts at round j with (11) and stops once a source of randomness has been found at round $j' < j$. The following definition gives a concrete formulation of this idea.

Definition 4.2. For symbols $(C, c) \in \{(Y, y), (W, w)\}$, and indices $i \in [\![1, l]\!]$, $j \in [\![2, r']\!]$, $p \in [\![0, j-1]\!]$, we say a (i, j, p)-chain, denoted $C(i, j, p)$, occurs in the view τ if the following conditions occur:

1. $c_{l+1}^j = c_i^j$;
2. $\delta_j(t_{l+1} \oplus t_i) \neq 0$;
3. $\forall j' \in [\![j-p, j-1]\!]$, $c_{l+1}^{j'}, c_i^{j'} \in U_{j'}$; and
4. if $j - p - 1 > 0$, $\left|\left\{c_{l+1}^{j-p-1}, c_i^{j-p-1}\right\} \cap U_{j-p-1}\right| < 2$.

If $p = 0$ we refer to the special case where the third condition does not occur, we call it an empty chain. Otherwise, if $p = j - 1$, then we call $C(i, j, p)$ a complete chain, and a partial chain in any other case.

[4] For some $x \neq x'$ and Δ, $\Pr[k \xleftarrow{\$} \mathcal{K} : \mathcal{H}_k(x) \oplus \mathcal{H}_k(x') = \Delta]$ is negligible.

For a symbol $\mathcal{C} \in \{\mathcal{Y}, \mathcal{W}\}$ and query $i \in [\![1, l]\!]$ we denote by $\mathcal{C}(j, p)$ the set of all chains $C(i, j, p)$ where $p \in [\![0, j-1]\!]$ is called the size of the chain.

Activity Pattern. More importantly, it is clear from the preceding discussion that we may have to consider a joint event on some consecutive rounds, as the earlier approach of bounding failure probability locally at each round will be tedious and loose. In our new approach, we associate to a view τ, a string $\mathtt{S}(\tau) = s_1 \dots s_{r'}$ over the alphabet $\Gamma = \{\top, \bot, 0, 1, 2, 3, 4, 5, 6, 7\}$ representing the failure at every round. In this context, the symbol $\top$ will correspond to an empty symbol while the symbol $\bot$ corresponds to a failing event we give away (as we cannot bound it's probability).

Description of $\mathtt{S}(\tau)$. We give a description of the mapping $\tau \mapsto \mathtt{S}(\tau)$:

- We start with a string $\mathtt{S}(\tau) = (\top)^{r'}$ (our representation of an empty string);
- For any round $j \in [\![1, r']\!]$, we assign a symbol to s_j the following way,
  - If $\mathcal{F}_0^j$ (collision with a permutation query) occurs we assign $s_j \leftarrow 0$. Otherwise, if $\mathcal{F}_2^j$ occurs we assign $s_j \leftarrow 4$. The randomness can be drawn from the key k_j;
  - Else if $\mathcal{F}_1^j \cup \mathcal{F}_3^j$ (collision between internal variables) occurs but $\delta_j(t_{l+1} \oplus t_i) = 0$ then we assign $s_j \leftarrow \bot$, as this represents an implied collision;
  - Else, if $\mathcal{Y}(j, 0)$ is non empty (there is an empty chain) then we assign $s_j \leftarrow 1$, otherwise if $\mathcal{W}(j, 0)$ is non empty then we assign $s_j \leftarrow 5$, in this case the probability of the collision between the variables can be bounded with the randomness of the previous round;
- Once the first loop is done we start searching for chains. At this point any chain will be of size at least one. For any $j \in [\![2, r']\!]$ we assign the following symbols,
  - Let p be maximal such that $\mathcal{Y}(j, p)$ is non empty, if $p > 0$, we assign $s_j \leftarrow 3$ and $s_{j'} \leftarrow 2$ for every $j' \in [\![j-p, j-1]\!]$;
  - Otherwise, let p' be maximal such that $\mathcal{W}(j, p)$ is non empty, if $p' > 0$, we assign $s_j \leftarrow 7$ and $s_{j'} \leftarrow 6$ for every $j' \in [\![j-p', j-1]\!]$;

We call $\mathtt{S}(\tau)$ the *activity pattern*, or simply the pattern corresponding to τ. In a sense, $\mathtt{S}(\tau)$ gives a necessary local view of the activity at each round during the computation of $\mathsf{TEML}_{\mathbf{k}}^{\mathbf{P}}(t', x')$ and $\mathsf{TEML}_{\mathbf{k}'}^{\mathbf{P}'}(t', z')$. It is easy to see that the following set captures the various patterns we can produce, i.e., let

$$\mathcal{P} = \left\{ S_1 \cdots S_d \ : \ \forall i \in [\![1, d]\!], (S_i \in \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3) \wedge \sum_{i=1}^{d} |S_i| = r' \right\},$$

where $\mathcal{S}_1 = \cup_{i=1}^{r'} \{0, 1, 4, 5\}^i$, $\mathcal{S}_2 = \{2^i 3, 6^i 7 : i \in [\![1, r'-1]\!]\}$, and $\mathcal{S}_3 = \{\bot^i \ : \ i \in [\![1, \alpha-1]\!]\}$. The following lemma gives a complete the characterization of $\mathtt{S}(\tau)$.

Lemma 4.5. *For any view τ, if $Col(\tau)$ occurred then $\mathtt{S}(\tau) \in \mathcal{P}$. Moreover, $\mathtt{S}(\tau)$ consists of at least $\lceil r'/\alpha \rceil$ non $\bot$ symbols.*

Proof. Let $\mathtt{S}(\tau) = s_1 \dots s_{r'}$ where for every $j \in [\![1, r']\!]$, $s_j \in \Gamma$. The first part of the lemma is easy to see by the construction of $\mathtt{S}(\tau)$, the definition of $\mathcal{P}$, and α-bijectivity of δ. As for the second part, note that, $s_1 \neq \bot$ (by definition). Divide the remaining $r'-1$ symbols in contiguous substrings of length α, except the last which could be of length less than α. Then, using the α-bijectivity of δ, we have at least $1 + \lfloor \frac{r'-1}{\alpha} \rfloor \geq \lceil r'/\alpha \rceil$ non-$\bot$ symbols in $\mathtt{S}(\tau)$. □

We will be interested in the probability that the view τ produces a pattern $S \in \mathcal{P}$, i.e, $\Pr[\mathtt{S}(\tau) = S]$, which essentially covers the global failure event. For any string $S \in \mathcal{P}$, fix a representation $S = S_1 \dots S_d$. Then, we can write $\mathtt{S}(\tau) := E_1(\tau) \cdots E_d(\tau)$ such that $|E_i(\tau)| = |S_i|$. Let d' be the number of strings such that $S_i \notin \perp^{<\alpha}$, and let $m_1 < m_2 < \ldots < m_{d'}$ be the indices corresponding to these sets.

Let $\mathtt{F}_0$ be an event that is always true, and for all $j \in [\![1, d'-1]\!]$, let $\mathtt{F}_j$ denote the event $(E_{m_1}(\tau) = S_{m_1}, \dots, E_{m_j}(\tau) = S_{m_j})$. We are interested in the following conditional probability

$$\Pr_{\tau}[E_{m_i}(\tau) = S_{m_i} \mid \mathtt{F}_{i-1}] \tag{12}$$

for all $i \in [\![1, d']\!]$. Note that, the above conditional event is well-defined, and non-trivial. This can be argued as follows: for distinct $i, i' \in [\![1, d']\!]$, $E_{m_i}(\tau)$ and $E_{m_{i'}}(\tau)$ involve different rounds. Further, at round $j \in [\![1, r']\!]$, we either use the randomness of the key k_j or that of the permutation P_{j-1}, which means that no two rounds share the same source of randomness.

In Lemma 4.6, we upper bound the conditional probability (12) depending on the type of string S_{m_i} for all $i \in [\![1, d']\!]$.

Lemma 4.6. *Suppose $|S_{m_i}| = s$. Then*

1. *for $i = 1$, we have*

$$\Pr_{\tau}[E_{m_i}(\tau) = S_{m_i} \mid \mathtt{F}_{i-1}] \leq \left(\frac{2q_{\max}}{N}\right)^{s-1}.$$

2. *for $i > 1$, we have*

$$\Pr_{\tau}[E_{m_i}(\tau) = S_{m_i} \mid \mathtt{F}_{i-1}] \leq \left(\frac{2q_{\max}}{N}\right)^{s}.$$

Proof. We prove the result in two cases:

Case A. $S_{m_i} \in \mathcal{S}_1$: Suppose $\Pi_{m_i}(\tau) = e_j \cdots e_{j+s-1}$ for some consecutive rounds $j, \dots, j+s-1$ such that $j \in [\![1, r'-s+1]\!]$. First, assume that $i > 1$, i.e., $\Pi_{m_i}(\tau)$ is not a prefix of $\mathtt{S}(\tau)$. Let $\mathtt{E}_0 = \mathtt{F}_{i-1}$, and for all $s' \in [\![1, s-1]\!]$, let $\mathtt{E}_{s'}$ denote the event $(e_j = f_j, \dots, e_{j+s'-1} = f_{j+s'-1}, \mathtt{F}_{i-1})$. Our goal is to compute

$$\Pr[e_{j+s'} = f_{j+s'} \mid \mathtt{E}_{s'}], \quad \forall s' \in [\![0, s-2]\!]. \tag{13}$$

Now, we may have the following two cases, depending on the value of f_j:

Case I: $f_{j+s'} \in \{0, 4\}$. In this case, since $k_{j+s'}$ is uniform and independent of $\{k_1, \dots, k_{j+s'-1}\}$, we have

$$\Pr[(e_{j+s'} = f_{j+s'}) \,|\mathtt{E}_{s'}] \leq \frac{q_p}{N}. \tag{14}$$

Case II: $f_{j+s'} \in \{1, 5\}$. Without loss of generality, assume $f_{j+s'} = 1$. Then there exists $i' \in [\![1, l]\!]$ such that $y_{l+1}^{ij} = y_{i'}^{ij}$. This gives rise to the following equation,

$$P_{j+s'-1}\left(y_{l+1}^{j+s'-1}\right) \oplus P_{j+s'-1}\left(y_{i'}^{j+s'-1}\right) = \delta_{j+s'}(t_{l+1} \oplus t_{i'}) \neq 0 \tag{Eq$_{i'}$}$$

Further, since $\mathcal{F}_0^j \cup \mathcal{F}_2^j = \emptyset$, at least one of $y_{l+1}^{j+s'-1}$ or $y_{i'}^{j+s'-1}$ is fresh (does not belong to $U_{j+s'-1}$), whence we have,

$$\Pr[(e_{j+s'} = f_{j+s'}) \,|\mathtt{E}_{s'}] \leq \sum_{i' \leq l} \Pr[(\text{Eq}_{i'})] \leq \sum_{i' \leq l} \frac{1}{N - q_c - q_p} \leq \frac{2q_c}{N}. \tag{15}$$

From (13), (14) and (15), we have,

$$\Pr\left[e_{j+s'} = f_{j+s'} | \mathtt{E}_{s'}\right] \leq \frac{2q_{\max}}{N}. \tag{16}$$

Using chain rule, for $i > 1$, we have,

$$\Pr_{\tau}\left[\Pi_{m_i}(\tau) = S_{m_i} \mid \mathtt{F}_{i-1}\right] = \prod_{s'=1}^{s} \Pr\left[e_{j+s'} = f_{j+s'} \mid \mathtt{E}_{s'}\right] \leq \left(\frac{2q_{\max}}{N}\right)^{s}. \tag{17}$$

This proves the second part of the lemma for Case A. Now, if $i = 1$, then $\Pi_i(\tau)$ is a prefix of $\mathsf{S}(\tau)$, i.e., $j = 1$, then we can view it as $\Pi_{m_i}(\tau) := e_1 \| E'(\tau)$. Note that, the adversary can easily choose inputs $(x_1, \ldots, x_{l+1})$ such that

$$y^1_{l+1} = x^0_{l+1} \oplus \delta_1(t_{l+1}) \oplus k_1 = x^0_i \oplus \delta_1(t_i) \oplus k_1 = y^1_i.$$

Therefore, $\Pr\left[e_{i_1} = f_1\right] \leq 1$. Now, we have

$$\begin{aligned}\Pr_{\tau}\left[\Pi_{i_1}(\tau) = S_{i_1} \mid \mathtt{F}_0\right] &= \Pr\left[e_1 = f_1\right] \times \Pr\left[E'(\tau) = f_2 \cdots f_s \mid \mathtt{E}_1\right] \\ &\leq \left(\frac{2q_{\max}}{N}\right)^{s-1},\end{aligned}$$

where the last inequality follows from (17). This completes Case A.

Case B. $S_{m_i} \in \mathcal{S}_2$: Assume without loss of generality, $S_{m_i} = 2^{s-1}3$ (the proof for the other type of chain is identical). Suppose $E_{m_i}(\tau) = e_j \cdots e_{j+s-1}$ for some consecutive rounds $j, \ldots, j+s-1$ such that $1 \leq i_1 < \ldots < i_s \leq r$. Our goal is to compute

$$\Pr\left[e_{i_1} = 2, \ldots, e_{i_{s-1}} = 2, e_{i_s} = 3 \mid \mathtt{F}_{i-1}\right]. \tag{18}$$

Since $S_i = (2^{s-1}3)$, there exists $i' \in [\![1, l]\!]$ such that a $Y(i', j, p)$ chain for $p = s-1 \geq 1$ yields it. Hence, the following conditions hold,

$$y^{i_s}_{l+1} = y^{i_s}_{i'}, \delta_j(t_{l+1} \oplus t_{i'}) \neq 0 \tag{$C^1_{i'}$}$$
$$\forall m \in [\![1, s-1]\!], y^{i_m}_{l+1} \in U_{i_m}, \tag{C^2}$$
$$\forall m \in [\![1, s-1]\!], y^{i_m}_{i'} \in U_{i_m}, \tag{$C^3_{i'}$}$$

and, assuming $i_1 - 1 \geq 1$, one of the following conditions hold,

$$y^{i_1-1}_{l+1} \notin U_{i_1-1}, \tag{C^4}$$
$$y^{i_1-1}_{i'} \notin U_{i_1-1}. \tag{$C^5_{i'}$}$$

Using these conditions we conclude that,

$$\begin{aligned}\Pr\left[e_{i_1} = 2, \ldots, e_{i_{s-1}} = 2, e_{i_s} = 3 \mid \mathtt{F}_{i-1}\right] &\leq \sum_{i' \leq l} \Pr\left[\left((C^1_{i'}) \wedge (C^2) \wedge (C^3_{i'})\right) \wedge \left((C^4) \vee (C^5_{i'})\right)\right] \\ &\leq \sum_{i' \leq l} \Bigg(\Pr\left[(C^1_{i'}) \wedge (C^2) \wedge (C^3_{i'}) \wedge (C^4)\right] \\ &\qquad + \Pr\left[(C^1_{i'}) \wedge (C^2) \wedge (C^3_{i'}) \wedge (C^5_{i'})\right] \Bigg) \\ &\leq \sum_{i' \leq l} \Bigg(\Pr\left[(C^1_{i'}) \wedge (C^2) \wedge (C^3_{i'}) \mid (C^4)\right]\end{aligned}$$

$$+ \Pr\left[(C^1_{i'}) \wedge (C^2) \wedge (C^3_{i'}) \mid (C^5_{i'})\right]\Big). \tag{19}$$

Fix some query $i \in [\![1, l]\!]$ and consider the event,

$$\mathcal{E} = \left((C^1_{i'}) \wedge (C^2) \wedge (C^3_{i'}) \mid (C^5_{i'})\right)$$

Without loss of generality assume that $(C^5_{i'})$ occurs. To analyze this event we will need to split the conditions (C^2) and $(C^3_{i'})$ into sub-events. Our strategy will be to first bound the events $y^{i_m}_{l+1} \in U_{i_m}$ for any $m \in [\![1, s-1]\!]$, conditioned on $y^{i_s}_{l+1} = y^{i_s}_{i'}$. Assuming those events we can upper bound the probability that $y^{i_m}_{i'} \in U_{i_m}$ for any $m \in [\![1, s-1]\!]$ (starting from $m = s-1$ down to 1). More precisely, for any $j \in [\![1, s-1]\!]$, we define the events

$$\mathtt{E}^j_{l+1} : \bigwedge_{m=1}^{j} \left(y^{i_m}_{l+1} \in U_{i_m}\right), \qquad \mathtt{E}^j_{i'} : \bigwedge_{m=j}^{s-1} \left(y^{i_m}_{i'} \in U_{i_m}\right).$$

Additionally, we let $\mathtt{E}^0_{l+1}, \mathtt{E}^s_{i'}$ be always true. Then, using the chain rule, one has

$$\begin{aligned}
\Pr[\mathcal{E}] = &\prod_{m=1}^{s-1} \Pr\left[y^{i_m}_{l+1} \in U_{i_m} \mid \mathtt{E}^{m-1}_{l+1} \wedge (C^5_{i'})\right] \\
&\times \Pr\left[(C^1_{i'}) \mid \mathtt{E}^{s-1}_{l+1} \wedge (C^5_{i'})\right] \\
&\times \prod_{m=s-1}^{1} \Pr\left[y^{i_m}_{i'} \in U_{i_m} \mid \mathtt{E}^{s-1}_{l+1} \wedge (C^1_{i'}) \wedge \mathtt{E}^{m+1}_{i'} \wedge (C^5_{i'})\right] \\
\leq &\left(\frac{q_p}{N}\right)^{s-1} \times \prod_{m=s-1}^{1} \Pr\left[y^{i_m}_{i'} \in U_{i_m} \mid \mathtt{E}^{s-1}_{l+1} \wedge (C^1_{i'}) \wedge \mathtt{E}^{m+1}_{i'} \wedge (C^5_{i'})\right]. \qquad (20)
\end{aligned}$$

where the last inequality holds since all the round keys are uniform and independent, and $\Pr\left[(C^1_{i'}) \mid \mathtt{E}^{s-1}_{l+1} \wedge (C^5_{i'})\right] \leq 1$.

First assume that $E_i(\tau)$ is not a prefix of $\mathtt{S}(\tau)$, then $2 \leq i_1 < \ldots < i_s \leq r$. In this case, using (20), we will give a better upper bound on

$$\prod_{m=s-1}^{1} \Pr\left[y^{i_m}_{i'} \in U_{i_m} \mid \mathtt{E}^{s-1}_{l+1} \wedge (C^1_{i'}) \wedge \mathtt{E}^{m+1}_{i'} \wedge (C^5_{i'})\right] \leq \Pr\left[y^{i_1}_{i'} \in U_{i_1} | F_0\right], \tag{21}$$

where $F_0 = \left(\mathtt{E}^{s-1}_{l+1} \wedge (C^1_{i'}) \wedge \mathtt{E}^2_{i'} \wedge (C^5_{i'})\right)$. Now to bound the last probability, we claim that conditioned on F_0, there exists at most one $u_{i_1} \in U_{i_1}$ such that $y^{i_1}_{i'} = u_{i_1}$. To prove that we proceed by reverse recursion on $1 \leq m \leq s-1$. First for $m = s-1$, note that since F_0 occurs this implies the variable $x^{i_s-1}_{l+1}$ is fixed, since the round keys involved in the event $\mathtt{E}^{s-1}_{l+1}$ are fixed and as a consequence $x^{i_s-1}_{i'}$ is fixed to,

$$x^{i_s-1}_{i'} = x^{i_s-1}_{l+1} \oplus \delta_{i_s-1}(\mathbf{t}_{l+1} \oplus \mathbf{t}_{i'}) \neq x^{i_s-1}_{l+1}.$$

Since the adversary never repeats a primitive query, this gives at most one choice of $v_{i_s-1} \in V_{i_s-1}$, such that $x^{i_s-1}_{i'} = v_{i_s-1}$. In other words, there is at most one $(u_{i_s-1}, v_{i_s-1}) \in \mathcal{Q}_{P_{i_s-1}}$ such that $y^{i_s-1}_{i'} = u_{i_s-1}$. Applying the same argumentation, it is easy to show that there is at most one $(u_{i_m}, v_{i_m}) \in \mathcal{Q}_{P_{i_m}}$, such that $y^{i_m}_{i'} = u_{i_m}$ for any $1 \leq m < s-1$, which proves our claim. Now, $y^{i_1}_{i'} = u_{i_1}$ can be rewritten

as $x_{i'}^{i_1-1} = u_{i_1} \oplus k_{i_1} \oplus \delta_{i_1-1}(\mathbf{t}_{i'})$. Since, $y_{i'}^{i_1-1} \notin U_{i_1-1}$ and $x_{i'}^{i_1-1} = P_{i_1-1}(y_{i'}^{i_1-1})$ then by using the randomness of the permutation P_{i_1-1}, the value $y_{i'}^{i_1-1}$ is chosen uniformly at random from a set of size at least $N - q_c - q_p$. In conclusion, we have the following upper bound,

$$\Pr\left[y_{i'}^{i_1} \in U_{i_1} \mid F_0\right] \leq \frac{1}{N - q_c - q_p} \leq \frac{2}{N}. \tag{22}$$

Using (19), (20), (21), and (22), we have:

$$\Pr\left[e_{i_1} = 2, \ldots, e_{i_{s-1}} = 2, e_{i_s} = 3 \mid \mathtt{F}_{i-1}\right] \leq 4\left(\frac{q_{\max}}{N}\right)^s \leq \left(\frac{2q_{\max}}{N}\right)^s.$$

This proves the second part of the lemma for Case B.

Now, assume that $E_i(\tau)$ is a prefix of $\mathtt{S}(\tau)$. In this case, we have

$$\begin{aligned}\Pr\left[e_{i_1} = 2, \ldots, e_{i_{s-1}} = 2, e_{i_s} = 3 \mid \mathtt{F}_{i-1}\right] &\leq \Pr\left[\forall m \in [\![1, s-1]\!], y_{l+1}^{i_m} \in U_{i_m}\right] \\ &\leq \left(\frac{2q_p}{2^n}\right)^{s-1} \leq \left(\frac{2q_{\max}}{2^n}\right)^{s-1},\end{aligned}$$

where the second inequality follows from the independence of key tuple. This proves the first part of the lemma for Case B, whence the proof is complete. □

Lemma 4.7. *Let q_c, q_p be positive integers, $q_{\max} = \max\{q_c, q_p\}$, and $q_c + q_p < N/2$. For any pattern $S \in \mathcal{P}$, we have*

$$\Pr_{\tau}\left[\mathtt{S}(\tau) = S\right] \leq \left(\frac{2q_{\max}}{N}\right)^{\left\lceil \frac{r'}{\alpha} \right\rceil - 1}.$$

Proof. By repeated application of Lemma 4.6, we have

$$\begin{aligned}\Pr_{\tau}\left[\mathtt{S}(\tau) = S\right] &\leq \prod_{i=1}^{k'} \Pr_{\tau}\left[E_{m_i}(\tau) = S_{m_i} \mid \mathtt{E}_{i-1}\right] \\ &\leq \left(\frac{2q_{\max}}{N}\right)^{s_{m_1}-1} \times \prod_{i=2}^{k'} \left(\frac{2q_{\max}}{N}\right)^{s_{m_i}} \\ &\leq \left(\frac{2q_{\max}}{N}\right)^{\sum_{i=1}^{k'} s_{m_i} - 1} \\ &\leq \left(\frac{2q_{\max}}{N}\right)^{\left\lceil \frac{r'}{\alpha} \right\rceil - 1}\end{aligned}$$

where $\sum_{i=1}^{k'} s_{m_i} \geq \left\lceil \frac{r'}{\alpha} \right\rceil$ comes from Lemma 4.5. This completes the proof. □

Now, we return to our main problem, i.e., (10). We have,

$$\begin{aligned}\Pr[\mathtt{fail}] &\leq \sum_{\tau}(Col(\tau)) \qquad (23)\\ &\leq \sum_{S \in \mathcal{P}} \Pr[Col(\tau) \wedge \mathtt{S}(\tau) = S] \\ &\leq \sum_{S \in \mathcal{P}} \Pr[\mathtt{S}(\tau) = S] \\ &\leq 8^{r'} q_c \left(\frac{2q_{\max}}{N}\right)^{\lceil r'/\alpha \rceil - 1} \qquad (24)\end{aligned}$$

The Hybrids-Distance lemma follows from (10) and (24).

5 Proof of Lemma 3.1

Consider the distinguisher (or differentiator) $\mathcal{D}$ described in Algorithm 5.1 interacting with $(\widetilde{\Pi}, \mathbf{P})$.

We claim that $\mathcal{D}$ outputs 1 with overwhelming probability when interacting with $(\mathsf{TEML}^{\mathbf{P}}, \mathbf{P})$. As a first step, notice that K, K', K'', K''' are not pairwise distinct in only two scenarios, either $x_2 = x_2'$ or $x_2 \oplus x_2' = \gamma_1(K \oplus K')$. Note that the first scenario is impossible as if $x_2 = x_2'$ it implies that $y_2 = y_2'$, which in turn implies that $x_3 = x_3'$. Continuing this process we must have $y_{r+1} = y_{r+1'}$, hence the two round keys k_{r+1} and $k_{r+1'}$ must be equal in contradiction to our assumption. In the second scenario, note that the probability the equation $x_2 \oplus x_2' = \gamma_1(K \oplus K') \neq 0$ holds, where $x_2 = P_2^{-1}(y_2), x_2' = P_2^{-1}(y_2')$, is satisfied with probability at most $1/(N-1)$, since P_2 is a random permutation. Moreover, note that since the schedule is linear, $\gamma_1(K + K') \oplus k_1'' \oplus k_1''' = 0$. Therefore, since the schedule is also r-bijective we get that $K \oplus K' \oplus K'' \oplus K''' = 0$.

Next, we show that conditioned on $\mathcal{D}$ not outputting 0 in line 15, it always outputs 1. Consider the computational paths of inputs (K, x), (K', x'), (K'', x''), (K''', x''') and note that they are well defined from our last two observations. Following both computational paths of (K, x) and (K', x') inside the EM cipher, it is easy to see that the input to P_{r+2} in both paths is x_{r+2}. Similarly, in both computational paths of (K'', X'') and (K''', x''') the input to the permutation P_{r+2} is $x_{r+2}' = y_{r+1} \oplus k_{r+1}' = y_{r+1}' \oplus k_{r+1}$. It implies that,

$$\begin{aligned} y \oplus y' &= (y_{r+2} \oplus \gamma_{r+2}(K)) \oplus (y_{r+2}' \oplus \gamma_{r+2}(K')), \\ y'' \oplus y''' &= (y_{r+2}' \oplus \gamma_{r+2}(K'')) \oplus (y_{r+2} \oplus \gamma_{r+2}(K''')). \end{aligned} \tag{25}$$

where $y_{r+2} = P_{r+2}(x_{r+2})$ and $y_{r+2}' = P_{r+2}(x_{i+2}')$. Finally, one has,

$$y \oplus y' \oplus y'' \oplus y''' = \gamma_{r+2}(K \oplus K' \oplus K'' \oplus K''') = \gamma_{r+2}(0) = 0,$$

where the last equation comes from combining the equations in (25), the definition of the master keys in line 10 and the fact that γ_{r+2} is linear. Hence, we get the following upper bound,

$$\Pr\left(\mathcal{D}^{\mathsf{TEML}^{\mathbf{P}}, \mathbf{P}} = 1\right) \geq 1 - \frac{1}{N-1}.$$

Consider now what happens when $\mathcal{D}$ interacts with $(\widetilde{\Pi}, \mathsf{Sim}^{\widetilde{\Pi}})$ for some efficient simulator Sim which makes at most σ queries when $\mathcal{D}$ makes at most $2r+6$ queries. Denote by $\{0,1\}^n$ the Turing machine which runs both $\mathcal{D}$ and Sim together, which make at most $q' = 2r + 6$ queries to $\widetilde{\Pi}$. Whenever $\mathcal{D}$ outputs 1, we see that $\{0,1\}^n$ has successfully found four inputs $(K, x), (K', x'), (K'', x''), (K''', x''') \in \{0,1\}^{rn} \times \{0,1\}^n$ such that K, K', K'', K''' are pairwise distinct and satisfy the following system of equations:

$$\begin{aligned} K \oplus K' \oplus K'' \oplus K''' &= 0, \\ x \oplus x' \oplus x'' \oplus x''' &= 0, \\ y \oplus y' \oplus y'' \oplus y''' &= 0. \end{aligned}$$

where $y = \widetilde{\Pi}(K, x), y' = \widetilde{\Pi}(K', x'), y'' = \widetilde{\Pi}(K'', x''), y''' = \widetilde{\Pi}(K''', x''')$. Note that the first two equations occur with probability 1 according to the definition of the distinguisher. Consider the q' queries of $\{0,1\}^n$ to $\widetilde{\Pi}$ sequentially, and denote by $\mathtt{BAD}$ the event where such values can be found among the q' queries. For any $i \in rng1q'$, let $\mathtt{BAD}_i$ be the event where such values can be found among the first i queries. Hence, by the union bound,

$$\Pr(\mathtt{BAD}) \leq \sum_{i=1}^{q'} \Pr\left(\mathtt{BAD}_i | \overline{\mathtt{BAD}_{i-1}}\right).$$

Algorithm 5.1 The sequential differentiator, $\mathcal{D}^{(\widetilde{\Pi},\mathbf{P})}$.

Choose $x_{r+2}, k_{r+1}, k'_{r+1} \leftarrow_{\$} \{0,1\}^n$ at random such that $k_{r+1} \neq k'_{r+1}$.
Compute $y_{r+1} \leftarrow x_{r+2} \oplus k_{r+1}, y'_{r+1} \leftarrow x_{r+2} \oplus k'_{r+1}$.
Query $x_{r+1} \leftarrow P_{r+1}^{-1}(y_{r+1}), x'_{r+1} \leftarrow P_{r+1}^{-1}(y'_{r+1})$.
for $i' \in [\![2, r]\!]$ **do**
    Let $i = (r-2) - i'$.
    Choose $k_i \leftarrow_{\$} \{0,1\}^n$ at random.
    Compute $y_i \leftarrow k_i \oplus x_{i+1}, y'_i \leftarrow k_i \oplus x'_{i+1}$.
    Query $x_i \leftarrow P_i^{-1}(y_i), x'_i \leftarrow P_i^{-1}(y'_i)$.
end for
Let $I = [\![2, r+1]\!]$, compute the master keys:

$$K \leftarrow \gamma_I^{-1}(k_2, \ldots, k_{r+1}), K' \leftarrow \gamma_I^{-1}(k_2, \ldots, k'_{r+1}).$$

Compute $y_1 \leftarrow x_2 \oplus \gamma_1(K), y'_1 \leftarrow x'_2 \oplus \gamma_1(K')$.
Query $x_1 \leftarrow P_1^{-1}(y_1), x'_1 \leftarrow P_1^{-1}(y'_1)$.
Compute the round keys $k''_1 \leftarrow y_1 \oplus x'_2, k'''_1 \leftarrow y'_1 \oplus x_2$.
Let $I' = [\![1, r]\!]$, compute the master keys:

$$K'' \leftarrow \gamma_{I'}^{-1}(k''_1, k_2, \ldots, k_r), K''' \leftarrow \gamma_{I'}^{-1}(k'''_1, k_2, \ldots, k_r).$$

if K, K', K'', K'' are not pairwise distinct **then** return 0 **else** continue
Compute the inputs:

$$x \leftarrow x_1 \oplus \gamma_0(K), x' \leftarrow x'_1 \oplus \gamma_0(K'),$$
$$x'' \leftarrow x_1 \oplus \gamma_0(K''), x''' \leftarrow x'_1 \oplus \gamma_0(K''').$$

Query $y \leftarrow \widetilde{\Pi}(K, x), y' \leftarrow \widetilde{\Pi}(K', x'), y'' \leftarrow \widetilde{\Pi}(K'', x''), y''' \leftarrow \widetilde{\Pi}(k''', x''')$.
if $y \oplus y' \oplus y'' \oplus y''' = 0$ **then** return 1 **else** return 0

Let $i \in [\![1, q']\!]$ and consider the i-th encryption query (a similar argument can made about decryption), $y_i = \widetilde{\Pi}(K_i, x_i)$. Assume without loss of generality that K_i is distinct from the previous keys used up to now (otherwise it does not help to find pairwise distinct keys in the system above). Hence, y_i is chosen uniformly at random from the set $\{0,1\}^n$. Therefore, $\mathtt{BAD}_i$ occurs only if y_i takes some value from a set of size at most $\binom{i-1}{3} \leq i^3$. In conclusion,

$$\Pr[\mathtt{BAD}] \leq \sum_{i=1}^{q'} \Pr\left(\mathtt{BAD}_i | \overline{\mathtt{BAD}_{i-1}}\right) \leq \sum_{i=1}^{q'} \frac{i^3}{N} \leq \frac{q'^4}{2N}$$

Therefore, we conclude that,

$$\Pr\left(\mathcal{D}^{\widetilde{\Pi},\mathsf{Sim}^{\widetilde{\Pi}}} = 1\right) \leq \Pr[\mathtt{BAD}] \leq \frac{q'^4}{2N}.$$

6 Proof of Sequential Indifferentiability of TEML

Notations. In this section, we have access to a weak r-bijective tweakey schedule γ. For any contiguous r-tuple, $I = [\![i, i+r-1]\!]$ we denote by $\gamma_I^{-1} : (\{0,1\}^n)^r \to \{0,1\}^{rn}$ the inverse mapping of the bijection

$$k \mapsto (\gamma_i(k), \ldots, \gamma_{i+r-1}(k)).$$

6.1 The Simulator

We start with an informal description of our simulator (a formal pseudo-code description is given in Algorithm 6.1). The simulator offers an interface $\mathsf{Query}(i, \delta, w)$ to the distinguisher for querying the internal permutations, where $i \in [\![1, r+3]\!]$ indicates the index of the permutation, and $\delta \in \{+,-\}$ the direction of the query (direct or inverse). For each $i \in [\![1, r+3]\!]$ the simulator maintains (internally) a table Π_i mapping entries $(\delta, w) \in \{+,-\} \times \{0,1\}^n$ to a value $w' \in \{0,1\}^n$, initially undefined for all entries (defined by the symbol $\perp$). We denote by Π_i^+, respectively Π_i^-, the time dependent sets of strings $w \in \{0,1\}^n$ such that Π_i^+, respectively Π_i^-, is defined (not a $\perp$ symbol). When the simulator receives a query (i, δ, w), it looks in table Π_i to see whether the corresponding answer $\Pi_i(\delta, w)$ is already defined. When this is the case, it outputs the answer and waits for the next query. Otherwise, it randomly draws an answer $w' \in \{0,1\}^n$ and defines $\Pi_i(\delta, w) := w'$, as well as the opposite direction table entry, $\Pi_i(\overline{\delta}, w') := w$, where $\overline{\delta}$ defined as $-$ if δ is $+$, and $+$ otherwise. In order to handily describe how the answer w', we make the randomness used by the simulator explicit through a tuple of random permutations $\mathbf{P} = (P_1, \ldots, P_{r+3})$.

After this random choice of the answer w', and before returning it to the distinguisher, the simulator takes additional steps to ensure consistency with the ideal cipher by running a chain completion mechanism. For that we define the set of all intermediate value in the middle layer of the cipher. Formally, let A_r be the set of all r tuples, $\boldsymbol{a} = ((u_i, v_i) : i \in [\![3, r+1]\!])$, where for each $i \in [\![3, r+1]\!]$, we have that $\Pi_i^+(u_i) = v_i$ and $\Pi_i^-(v_i) = u_i$. Then, if the distinguisher called $\mathsf{Query}(i, \delta, w)$ for $i = 2$ or $i = r+2$, the simulator completes all newly created "chains" $(v_2, \boldsymbol{a}, u_{r+2})$ where $v_2 \in \Pi_2^-$, $u_{r+2} \in \Pi_{r+2}^+$ and $\boldsymbol{a} \in A_r$, by executing a procedure $\mathsf{CompleteChain}(v_2, \boldsymbol{a}, u_{r+}, \ell)$ where ℓ indicates at which endpoint the chain will be "adapted".

For example, assume that the distinguisher called $\mathsf{Query}(2, +, u_2)$ and that the answer randomly chosen by the simulator was u_2 (or backwards where the random value is u_2). Then for each $\boldsymbol{a} \in A_r$ and endpoints $u_{r+2} \in \Pi_{r+2}^-$, the simulator computes the corresponding round keys $k_i = v_i \oplus u_{i+1}$ for $i \in [\![2, r+1]\!]$, and defines the master key $K = \gamma_{[\![2,r+1]\!]}(k_2, \ldots, k_{r+1})$. The simulator then can adapt at round 1. Indeed, we can

Algorithm 6.1 Formal Description of the Simulator, $\mathsf{Sim}(\mathbf{P})$

Variables:
  tables $(\Pi_i : i \in [\![1, r+3]\!])$ initially empty.

procedure QUERY(i, δ, w)
  if $(\delta, w) \notin \Pi_i$ **then**
    $w' = P_i(\delta, w)$
    $\Pi_i(\delta, w) := w'$
    $\Pi_i(\bar{\delta}, w') := w$ ▷ may overwrite an entry
    // complete $(v_2, \boldsymbol{a}, u_{r+2})$ chains if exists
    if $i = 2$ **then**
      if $\delta = +$ **then** $v_i := w'$ **else** $v_i := w$
      for all $\boldsymbol{a} \in A_r, u_{r+3} \in \Pi^+_{r+3}$ **do**
        COMPLETECHAIN$(v_2, \boldsymbol{a}, u_{r+2}, 1)$
      end for
    else if $i = r + 2$ **then**
      if $\delta = +$ **then** $u_i := w$ **else** $u_i := w'$
      for all $\boldsymbol{a} \in A_r, v_2 \in \Pi^-_2$ **do**
        COMPLETECHAIN$(v_2, \boldsymbol{a}, u_{r+2}, r+3)$
      end for
    end if
  end if
  return $\Pi_i(\delta, w)$
end procedure

procedure FORCEVAL(u_i, v_i, i)
  $\Pi_i(+, u_i) := v_i$ ▷ may overwrite an entry
  $\Pi_i(-, v_i) := u_i$ ▷ may overwrite an entry
end procedure

procedure COMPLETECHAIN$(v_2, \boldsymbol{a}, u_{r+2}, \ell)$
  for $i \in [\![2, r+1]\!]$ **do**
    $k_i := v_i \oplus u_{i+1}$
  end for
  $K = \gamma_I^{-1}(k_2, \ldots, k_{r+1})$, $I = [\![2, r+1]\!]$
  case $\ell = 1$:
    // evaluate chain backwards up to v_1
    $u_2 := \Pi_2(-, v_2)$
    $v_1 := u_2 \oplus \gamma_2(K)$
    // evaluate chain forwards up to u_1
    $v_{r+2} := \Pi_{r+2}(+, u_{r+2})$
    $u_{r+3} = v_{r+2} \oplus \gamma_{r+2}(K)$
    $v_{r+3} = \text{QUERY}(r+3, +, u_{r+3})$
    $x := \widetilde{\Pi}(-, K, v_{r+3} \oplus \gamma_{r+3}(K))$
    $u_1 := x \oplus \gamma_0(K)$
    // adapt the chain
    FORCEVAL$(u_1, v_1, 1)$
  case $\ell = r + 3$:
    // evaluate chain forwards up to u_{r+3}
    $v_{r+2} = \Pi_{r+2}(+, u_{r+2})$
    $u_{r+3} = v_{r+2} \oplus \gamma_{r+2}(K)$
    // evaluate chain backwards up to v_{r+3}
    $u_2 := \Pi_2(-, v_2)$
    $v_1 := u_2 \oplus \gamma_1(K)$
    $u_1 := \text{QUERY}(1, -, v_1)$
    $y = \widetilde{\Pi}(+, K, u_1 \oplus \gamma_0(K))$
    $v_{r+3} = y \oplus \gamma_{r+3}(K)$
    // adapt the chain
    FORCEVAL$(u_{r+3}, v_{r+3}, r+3)$
end procedure

compute the value $v_1 = k_2 \oplus \Pi_2(-, v_2)$. Moreover, looking at the other endpoint of the cipher we can retrieve u_1 by applying the following steps:

$$\begin{aligned}
&v_{r+2} = \Pi_{r+2}(+, u_{r+2}), u_{r+3} = v_{r+2} \oplus \gamma_{r+2}(K),\\
&v_{r+3} = \Pi_{r+3}(+, u_{r+3}), x = \widetilde{\Pi}(-, K, v_{r+3} \oplus \gamma_{r+3}(K)),\\
&u_1 = x \oplus \gamma_0(K),
\end{aligned}$$

where v_{r+3} is drawn at random if it is not in Π^+_{r+3}. Now we can force the pair of input/output (u_1, v_1) to the table Π_1, in order to ensure consistency of the simulated TEML construction with $\widetilde{\Pi}$. For the case of Query $(r+3, \cdot, \cdot)$ the behavior of the simulator will be symmetrical, namely adaptation of the chain takes place in Π_{r+3} instead.

In order to prove this claim, we will first define a simulator Sim, then prove that it runs in polynomial time and makes a polynomial number of queries, and finally prove that the two systems $\Sigma_1 = (\widetilde{\Pi}, \mathsf{Sim}^E)$ and $\Sigma_3 = (\mathsf{TEML}^{\mathbf{P}}, \mathbf{P})$ are sequentially indistinguishable, using an intermediate system Σ_2.

Lemma 6.1. *Consider the execution of the simulator* $\mathsf{Sim}^{\widetilde{\Pi}}$ *which makes q queries in total. Then:*

1. *the size of $\Pi_2, \ldots, \Pi_{r+2}$ is at most q,*
2. *the size of Π_1, Π_{r+3} is that most $q^{r+1} + q$.*
3. *the simulator executes* CompleteChain *at most q^{r+1} times and makes at most q^{r+1} queries to E,*
4. *the total runtime of the simulator is $\mathcal{O}(q^{r+1})$.*

Proof. Notice that for $i \in [\![2, r+2]\!]$, the table Π_i can only increase in a call to the procedure Query(i, δ, w). Therefore the size of Π_i is bounded by the number of the distinguisher's queries q. CompleteChain is called once for at most every tuple of permutation queries, $((u_i, \Pi_i(+, u_i)) : i \in [\![2, r+2]\!])$, hence at most q^{r+1} in total. Since CompleteChain makes at most one query to $\widetilde{\Pi}$, the simulator cannot make more than q^{r+1} queries to $\widetilde{\Pi}$. Note that tables Π_1 and Π_{r+3} are only increased by one for the calls to Query$(1, \delta, w)$ or Query$(5, \delta, w)$, which happens only once in the procedure CompleteChain , therefore the size of those tables is bounded by $q^{r+1} + q$. In conclusion, since CompleteChain runs in constant runtime, the total runtime of the simulator is $\mathcal{O}(q^{r+1})$. □

6.2 Intermediate Games and Distance Between Them

We will denote by $\mathsf{Sim}(\widetilde{\Pi}, \mathbf{P})$ the simulator with oracle access to the ideal cipher E and the randomness coming from $\mathbf{P}$. In order to prove the indistinguishability of the two systems $(\widetilde{\Pi}, \mathsf{Sim}(\widetilde{\Pi}, \mathbf{P}))$ and $(\mathsf{TEML}^{\mathbf{P}}, \mathbf{P})$, we will use an intermediate system $\Sigma_2 = (\mathsf{TEML}^{\mathsf{Sim}(\widetilde{\Pi}, \mathbf{P})}, \mathsf{Sim}(\widetilde{\Pi}, \mathbf{P}))$. In other words, the right oracle is the simulator $\mathsf{Sim}(\widetilde{\Pi}, \mathbf{P})$, with oracle access to an ideal cipher E as in Σ_1, but now the left oracle is the $r+3$-round TEML construction with oracle access to $\mathsf{Sim}(\widetilde{\Pi}, \mathbf{P})$ instead of independent random permutations.

6.2.1 Transition from Σ_1 to Σ_2

Definition 6.1. A pair $(\widetilde{\Pi}, \mathbf{P})$ is said to be good if the simulator Sim never overwrites an entry of its tables $(\Pi_i : i \in [\![1, r+3]\!])$ during an execution of $\mathcal{D}^{\Sigma_2(\widetilde{\Pi}, \mathbf{P})}$, otherwise the pair is called bad.

Note that an overwrite may only happen during a random assignment in Line (7) or when adapting a chain in Lines (24) and (25). Moreover, whether a pair is good depends on the queries of the distinguisher $\mathcal{D}$. We first upper bound the probability that a random pair $(\widetilde{\Pi}, \mathbf{P})$ is bad.

Lemma 6.2. *Consider a distinguisher $\mathcal{D}$ with total oracle query cost at most q, with $q^{r+1} \leq N/4$. Then a uniformly random pair $\widetilde{\Pi} \leftarrow\!\!\$\, \widetilde{\mathsf{P}}(rn, n)$ and $\mathbf{P} \in (\mathsf{P}(n))^{r+3}$ is bad, with respect to $\mathcal{D}$, with probability at most $\frac{16q^{2r+2}}{N}$.*

Proof. First, note that the total number of queries received by the simulator in Σ_2 is exactly q. Since $\Pi_2, \ldots, \Pi_{r+2}$ are never adapted, they can never be overwritten either. Therefore, we only consider the probability of the tables Π_1 and Π_{r+3}. Let $\mathsf{BadRand}$ be the event that an overwrite occurs during a random assignment, at line (7), and $\mathsf{BadAdapt}$ be the event that an overwrite occurs when adapting a chain $(v_2, \boldsymbol{a}, u_{r+2})$ at lines (24) and (25).

We first consider the probability of $\mathsf{BadRand}$. Let $i \in \{1, r+3\}$ and consider the assignments $\Pi_i(\delta, w) := w'$ and $\Pi_i(\bar{\delta}, w') := w$ where $w' := P_i(\delta, w)$ and P_i is some random permutation. By Lemma 6.1 (2), there at most $q^{r+1} + q$ random assignments in Π_1 and Π_{r+3}, so that w' is sampled out of a set of size at least $N - q^{r+1} - q$. Moreover, this assignment cannot overwrite a value that was previously added during a random assignment, but only a value that was added by $\mathsf{ForceVal}$, when adapting a chain, therefore by Lemma 6.1 (3) there are at most q^{r+1} such values. In conclusion, the probability w' hits a previously added value in table Π_i by a call to $\mathsf{ForceVal}$ is at most $\frac{q^{r+1}}{N-q^{r+1}-q}$. Summing over all possible random assignments in Π_1 and Π_{r+3}, we obtain the following upper bound,

$$\Pr\left[\mathsf{BadRand}\right] \leq 2(q^{r+1} + q) \cdot \frac{q^{r+1}}{N - q^{r+1} - q} \leq \frac{8q^{2r+2}}{N}.$$

Next, we consider the probability of $\mathsf{BadAdapt}$, conditioned on $\mathsf{BadRand}$ not occurring. Let $\mathsf{BadAdapt}_i$ be the event where a value is overwritten by the i-th call to $\mathsf{ForceVal}$. We will be interested in the probability

$$\Pr\left[\mathsf{BadAdapt}_i \middle| \neg\mathsf{BadRand} \wedge \left(\bigwedge_{j=1}^{i-1} \neg\mathsf{BadAdapt}_j\right)\right]$$

Consider the i-th execution of $\mathsf{CompleteChain}\,(v_2, \boldsymbol{a}, u_{r+2}, \ell)$ and assume that no value was overwritten before this i-th call to $\mathsf{CompleteChain}$. More precisely, consider the query $\mathsf{Query}\,(j, \delta, \cdot)$ that was triggered during the chain completion and the call to $\mathsf{ForceVal}\,(u_\ell, v_\ell, \ell)$. We must show that with high probability the entries of the tables $\Pi_\ell(+, u_\ell)$ and $\Pi_\ell(-, v_\ell)$ are undefined previously to this call. Distinguish between several cases. Assume $j = 2$, $\ell = 1$, (the case $i = r+2$, $\ell = r+3$ is symmetrical) and consider the value of v_1 given by $v_1 = u_2 \oplus \gamma_1(K)$ where $K = \gamma_{[\![2,r+1]\!]}(v_2 \oplus u_3, \ldots, v_{r+1} \oplus u_{r+2})$. Note that since $j = 2$, then either u_2 or v_2 is a random value. Consider when K is created and note that if v_2 is a random value and since the other values involved are fixed, then the value $\gamma_1(K)$ is a random variable that depends solely on the sampling of v_2. Therefore, v_1 takes a random value from a set of size at least $N - q$ (from u_2 or v_2). Hence, by Lemma 6.1 (2), the probability that v_1 takes a values from a defined value in table Π_1 is at most $\frac{q^{r+1}+q}{N-q}$. Next, we show that simulator never made the query $\widetilde{\Pi}(-, K, v_{r+3} \oplus \gamma_{r+3}(K))$ before nor received the value from a previous query to $\widetilde{\Pi}(+, K, u_1 \oplus \gamma_0(K))$. Assume otherwise, then there exists a chain $(v_2', \boldsymbol{a}', u_{r+2}')$ such that the query $\widetilde{\Pi}(-, K, v_{r+3} \oplus \gamma_{r+3}(K))$ appears during it's completion. Then, since both chains use the same master key (there is only one query to E for each chain completion), then $v_{r+3} = v_{r+3}'$. Hence, $u_{r+3} = u_{r+3}'$, which

implies that $v_{r+2} = v'_{r+2}$, since again they share the same master key. By going down with this recursive process we conclude that the chains are equal. Therefore, by Lemma 6.1 (3) and (2), the probability that $u_1 = x \oplus \gamma_0(K)$ hits one of the values in the table Π_1 is at most $\frac{q^{r+1}+q}{N-q^{r+1}}$, since that are at most q^{r+1} calls to E. Summing over all at most q^{r+1} calls to $\mathsf{CompleteChain}$, we conclude that,

$$\begin{aligned}\Pr\left[\mathsf{BadAdapt} \,|\neg\mathsf{BadRand}\right] &\leq \sum_{i=1}^{q^{r+1}} \Pr\left(\mathsf{BadAdapt}_i \middle| \neg\mathsf{BadRand} \wedge \left(\bigwedge_{j=1}^{i-1} \neg\mathsf{BadAdapt}_j\right)\right) \\ &\leq q^{r+1}\left(\frac{q^{r+1}+q}{N-q} + \frac{q^{r+1}+q}{N-q^{r+1}}\right) \leq \frac{8q^{2r+2}}{N}.\end{aligned}$$

Combining both upper bounds yields the result. □

Lemma 6.3. *For any distinguisher $\mathcal{D}$ of total oracle query cost at most q, one has,*

$$\left|\Pr\left[\mathcal{D}^{\Sigma_1(\widetilde{\Pi},\mathbf{P})} = 1\right] - \Pr\left[\mathcal{D}^{\Sigma_2(\widetilde{\Pi},\mathbf{P})} = 1\right]\right| \leq \frac{16q^{2r+2}}{N}.$$

where both probabilities are taken over $\widetilde{\Pi} \leftarrow_{\$} \widetilde{\mathsf{P}}(rn, n), \mathbf{P} \leftarrow_{\$} (\mathsf{P}(n))^{r+3}$.

Proof. We show that for any good pair $(\widetilde{\Pi}, \mathbf{P})$, the transcript of the interaction of $\mathcal{D}$ with $\Sigma_1(\widetilde{\Pi}, \mathbf{P})$ and $\Sigma_2(\widetilde{\Pi}, \mathbf{P})$ is identical. Since the distinguisher is sequential and they both share the same right oracle it is clear it is the same interaction during the first phase. For the second phase of the interaction, since the simulator never overwrites the tables Π_i for $i \in [\![1, r+3]\!]$, it follows that, for any $\delta \in \{+,-\}$ and $z \in \{0,1\}^n$, $\mathsf{TEML}^{\mathsf{Sim}(e,\mathbf{P})}(\delta, K, z) = \widetilde{\Pi}(\delta, K, z)$. Therefore, the interaction of $\mathcal{D}$ with $\Sigma_1(\widetilde{\Pi}, \mathbf{P})$ and $\Sigma_2(\widetilde{\Pi}, \mathbf{P})$ is identical in both phases. Hence,

$$\left|\Pr\left[\mathcal{D}^{\Sigma_1(\widetilde{\Pi},\mathbf{P})} = 1\right] - \Pr\left[\mathcal{D}^{\Sigma_2(\widetilde{\Pi},\mathbf{P})} = 1\right]\right| \leq \Pr\left[(\widetilde{\Pi}, \mathbf{P}) \text{ is bad}\right],$$

from which the result follows by Lemma 6.2. □

6.2.2 Transition From Σ_2 to Σ_3 and Randomness Mapping

To transition from Σ_2 to Σ_3 we will need the notion of a partial permutation.

Definition 6.2. A partial permutation is a function $P'_i : \{+,-\} \times \{0,1\}^n \to \{0,1\}^n \cup \{\perp\}$ such that for all $u, v \in \{0,1\}^n$, $P'_i(+,u) = v \neq \perp \Leftrightarrow P'_i(-,v) = u$.

For this, we define a map $\mathcal{A}$ mapping pairs $(\widetilde{\Pi}, \mathbf{P})$ either to the special symbol $\perp$ when $(\widetilde{\Pi}, \mathbf{P})$ is bad, or to a tuple of partial permutations as follows: run $\mathcal{D}^{\Sigma_2(\widetilde{\Pi},\mathbf{P})}$, and consider the tables $(\Pi_i : i \in [\![1, r+3]\!])$ at the end of the simulation, then fill all undefined entries of the tables with the special symbol $\perp$. Then define $\mathcal{A}(\widetilde{\Pi}, \mathbf{P}) = (\Pi_1, \ldots, \Pi_{r+3})$. Note that since $(\widetilde{\Pi}, \mathbf{P})$ is a good pair, the simulator never overwrites an entry in its tables, which implies that $\mathcal{A}(\widetilde{\Pi}, \mathbf{P})$ is indeed a partial permutation and $\mathcal{A}$ is well defined.

We say a tuple of partial permutations $\mathbf{P'} = (P'_1, \ldots, P'_{r+3})$ is good if there exists an ideal cipher E and a tuple of permutations $\mathbf{P} = (P_1, \ldots, P_{r+3})$, such that $\mathcal{A}(\widetilde{\Pi}, \mathbf{P}) = \mathbf{P'}$. We say that a tuple of permutations $\mathbf{P} = (P_1, \ldots, P_{r+3})$ extends a tuple of partial permutations $\mathbf{P'} = (P'_1, \ldots, P'_{r+3})$, denoted by $\mathbf{P} \vdash \mathbf{P'}$, if for any $i \in [\![1, r+3]\!]$, P_i and P'_i agree on all entries that are already defined in P'_i (where $P'_i(\delta, w) \neq \perp$).

Lemma 6.4. *For any distinguisher $\mathcal{D}$ of total oracle query cost at most q, one has,*

$$\left|\Pr\left[\mathcal{D}^{\Sigma_2(\widetilde{\Pi},\mathbf{P})}=1\right]-\Pr\left[\mathcal{D}^{\Sigma_3(\mathbf{P})}=1\right]\right| \leq \frac{\left((r+5)^2+16\right)q^{2r+2}}{N}.$$

where the first probability is take over $\widetilde{\Pi} \leftarrow_{\$} \widetilde{\mathsf{P}}(rn,n)$, $\mathbf{P} \leftarrow_{\$} (\mathsf{P}(n))^{r+3}$, and the second only over $\mathbf{P} \leftarrow_{\$} (\mathsf{P}(n))^{r+3}$.

Proof. Let

$$\varepsilon := \left|\Pr\left[\mathcal{D}^{\Sigma_2(\widetilde{\Pi},\mathbf{P})}=1\right]-\Pr\left[\mathcal{D}^{\Sigma_3(\mathbf{P})}=1\right]\right|$$

and assume without loss of generality that $\Pr\left[\mathcal{D}^{\Sigma_2(\widetilde{\Pi},\mathbf{P})}=1\right] \geq \Pr\left[\mathcal{D}^{\Sigma_3(\mathbf{P})}=1\right]$.

By the definition of the map $\mathcal{A}$, for any good tuple of partial permutations $\mathbf{P'}$, the outputs of $\mathcal{D}^{\Sigma_2(\widetilde{\Pi},\mathbf{P})}$ and $\mathcal{D}^{\Sigma_3(\mathbf{P})}$ are equal for any pair $(\widetilde{\Pi},\mathbf{P})$ such that $\mathcal{A}(\widetilde{\Pi},\mathbf{P})=\mathbf{P'}$, and any tuple of permutations $\mathbf{P}$ such that $\mathbf{P} \vdash \mathbf{P'}$. Let Θ_1 be the set of tuples of partial permutations $\mathbf{P'}$ such that $\mathcal{D}^{\Sigma_2(\widetilde{\Pi},\mathbf{P})}$ outputs 1 for any pair $(\widetilde{\Pi},\mathbf{P})$ such that $\mathcal{A}(\widetilde{\Pi},\mathbf{P})=\mathbf{P'}$. Then, we can conclude that,

$$\varepsilon \leq \Pr\left[(\widetilde{\Pi},\mathbf{P}) \text{ is bad}\right] + \sum_{P' \in \Theta_1}\left(\Pr\left[\mathcal{A}(\widetilde{\Pi},\mathbf{P})=\mathbf{P'}\right]-\Pr\left[\mathbf{P}\vdash\mathbf{P'}\right]\right). \tag{26}$$

Fix any good tuple of partial permutations $\mathbf{P'}$ and for any $i \in [\![1,r+3]\!]$ let

$$|P'_i| = |\{u \in \{0,1\}^n : P'_i(+,u) \neq \perp\}| = |\{v \in \{0,1\}^n : P'_i(-,v) \neq \perp\}|.$$

Then by definition of a partial permutation, one has,

$$\Pr\left[\mathbf{P} \leftarrow_{\$} (\mathsf{P}(n))^{r+3} : \mathbf{P} \vdash \mathbf{P'}\right] = \frac{1}{\prod_{i=1}^{r+3}(N)_{|P'_i|}}.$$

Fix now any good pre-image $(\widetilde{\widetilde{\Pi}},\tilde{\mathbf{P}})$ of $\mathbf{P'}$, where $\tilde{\mathbf{P}}=(\tilde{P}_1,\ldots,\tilde{P_{r+3}})$ and let q_e and let $(q_i : i \in [\![1,r+3]\!])$ be the number of queries made by the simulator to $\tilde{E}$ and $\tilde{P}_i$ respectively in the execution of $\mathcal{D}^{\Sigma_2(\widetilde{\Pi},\mathbf{P})}$. Note that for any pair $(\widetilde{\Pi},\mathbf{P})$, $\mathcal{A}(\widetilde{\Pi},\mathbf{P})=\mathbf{P'}$ if and only if the transcript of Sim with $(\widetilde{\Pi},\mathbf{P})$ in $\mathcal{D}^{\Sigma_2(\widetilde{\Pi},\mathbf{P})}$ is the same as the transcript of the interaction of Sim with $(\tilde{E},\tilde{\mathbf{P}})$ in $\mathcal{D}^{\Sigma_2(\tilde{E},\tilde{\mathbf{P}})}$. Then we conclude that,

$$\Pr\left[\widetilde{\Pi} \leftarrow_{\$} \widetilde{\mathsf{P}}(rn,n), \mathbf{P} \leftarrow_{\$} (\mathsf{P}(n))^{r+3} : \mathcal{A}(\widetilde{\Pi},\mathbf{P})=\mathbf{P'}\right] \leq \frac{1}{(N)_{q_e}\prod_{i=1}^{r+3}(N)_{q_i}},$$

since the probability is maximized when the same master key is used for all q_e queries to $\widetilde{\Pi}$. Moreover, since the number of executions of $\mathsf{ForceVal}$ made by the simulator, i.e., the number of chain adaptions, is exactly the number of queries made by the simulator to $\widetilde{\Pi}$, one has,

$$q_e + \sum_{i=1}^{r+3} q_i = \sum_{i=1}^{r+3} |P'_i| \leq 2q^{r+1} + (r+3)q. \tag{27}$$

where the last inequality follows by Lemma 6.1 (1) and (2). In conclusion, by Equation (27) we have that,

$$\begin{aligned}\frac{\Pr\left[\mathbf{P}\vdash\mathbf{P'}\right]}{\Pr\left[\mathcal{A}(\widetilde{\Pi},\mathbf{P})=\mathbf{P'}\right]} &= \frac{(N)_{q_e}\prod_{i=1}^{r+3}(N)_{q_i}}{\prod_{i=1}^{r+3}(N)_{|P'_i|}} \\ &\geq \frac{N^{q_e+\sum_{i=1}^{r+3}q_i}}{N^{\sum_{i=1}^{5}|P'_i|}}\prod_{j=1}^{q_e-1}\left(1-\frac{j}{N}\right)\prod_{i=1}^{r+3}\prod_{j=1}^{q_i-1}\left(1-\frac{j}{N}\right)\end{aligned}$$

$$\geq 1 - \frac{q_e^2 + \sum_{i=1}^{r+3} q_i^2}{N} \geq 1 - \frac{(2q^{r+1} + (r+3)q)^2}{N}$$
$$\geq 1 - \frac{(r+5)^2 q^{2r+2}}{N}$$

Combining (26) with (27), we obtain,

$$\begin{aligned}
\varepsilon &\leq \Pr\left((\widetilde{\Pi}, \mathbf{P}) \text{ is bad}\right) + \sum_{P' \in \Theta_1} \Pr\left[\mathcal{A}(\widetilde{\Pi}, \mathbf{P}) = \mathbf{P}'\right] \left(\frac{\Pr\left[\mathbf{P} \vdash \mathbf{P}'\right]}{\Pr\left[\mathcal{A}(\widetilde{\Pi}, \mathbf{P}) = \mathbf{P}'\right]}\right) \\
&\leq \Pr\left((\widetilde{\Pi}, \mathbf{P}) \text{ is bad}\right) + \frac{(r+5)^2 q^{2r+2}}{N} \sum_{P' \in \Theta_1} \Pr \mathcal{A}(\widetilde{\Pi}, \mathbf{P}) = \mathbf{P}' \\
&\leq \Pr\left((\widetilde{\Pi}, \mathbf{P}) \text{ is bad}\right) + \frac{(r+5)^2 q^{2r+2}}{N} \leq \frac{(r+5)^2 q^{2r+2}}{N} + \frac{16 q^{2r+2}}{N} \\
&= \frac{\left((r+5)^2 + 16\right) q^{2r+2}}{N}
\end{aligned}$$

□

Theorem 3.2 then follows from Lemma 6.3 and 6.4.

7 Conclusion

In this paper, we first showed that the $2r$-round Tweakable Even-Mansour with a specific class of linear tweak-key mixing, and αn-bit tweaks, is IND-CCA secure up to $2^{\frac{r-\alpha}{r}n}$ queries. The main ingredient of our proof is the well-known coupling technique. Our main technical contribution was a refreshed approach to get an upper bound on the probability of failure in coupling, which could be of independent interest. In particular, we think that this approach might also be useful in the analysis of the Feistel network with linear tweak and key absorption. As with several other coupling-based security bounds [LPS12, LS13b, CLS15], we believe that our IND-CCA bound is also not tight. Indeed, we conjecture that beyond a constant $c \geq 4$, the number of rounds can be effectively reduced by half, i.e., to r whenever $r \geq c$, while maintaining the same security level, i.e., up to $2^{\frac{r-\alpha}{r}n}$ queries.

Second, diverting our focus to the sequential indifferentiability setting, we showed that $(r+3)$ rounds are both necessary and sufficient for security of (Tweakable) Even-Mansour with rn-bit (twea)key and a special class of linear (twea)key mixing function.

As a direct consequence of our results, we gave a sound provable security footing for iterated round-based TBCs, notably following the design paradigm TWEAKEY, that employ a linear tweak-key mixing.

Acknowledgments

The authors would like to thank all the anonymous reviewers who reviewed and provided valuable comments on this paper. The initial work on this paper was conceived while Soumya Kanti Saha was a summer intern at CISPA Helmholtz Center for Information Security, and part of this work was written while Benoît Cogliati and Soumya Kanti Saha were respectively affiliated with CISPA, and Indian Statistical Institute Kolkata. Jordan Ethan and Ashwin Jha carried out this work under the framework of the French-German-Center for Cybersecurity, a collaboration of CISPA and LORIA.

References

[ABD+13] Elena Andreeva, Andrey Bogdanov, Yevgeniy Dodis, Bart Mennink, and John P. Steinberger. On the indifferentiability of key-alternating ciphers. In *Advances in Cryptology - CRYPTO 2013, Proceedings, Part I*, pages 531–550, 2013.

[Ald83] D. J. Aldous. Random walks on finite groups and rapidly mixing markov chains. In *Seminaire de Probabilites XVII, Lecture Notes in Mathematics*, pages 243–297. Springer, 1983.

[BGGS20] Zhenzhen Bao, Chun Guo, Jian Guo, and Ling Song. TNT: how to tweak a block cipher. In *Advances in Cryptology - EUROCRYPT 2020, Proceedings, Part II*, Lecture Notes in Computer Science, pages 641–673, 2020.

[BGIM19] Zhenzhen Bao, Jian Guo, Tetsu Iwata, and Kazuhiko Minematsu. ZOCB and ZOTR: tweakable blockcipher modes for authenticated encryption with full absorption. *IACR Trans. Symmetric Cryptol.*, 2019(2):1–54, 2019.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The skinny family of block ciphers and its low-latency variant mantis. In *Advances in Cryptology – CRYPTO 2016*, pages 123–153, 2016.

[BKN09] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. Distinguisher and related-key attack on the full AES-256. In *Advances in Cryptology - CRYPTO 2009, Proceedings*, pages 231–249, 2009.

[BLN18] Ritam Bhaumik, Eik List, and Mridul Nandi. ZCZ - achieving n-bit SPRP security with a minimal number of tweakable-block-cipher calls. In *Advances in Cryptology - ASIACRYPT 2018, Proceedings, Part I*, pages 336–366, 2018.

[CDJ+21] Avik Chakraborti, Nilanjan Datta, Ashwin Jha, Cuauhtemoc Mancillas-López, Mridul Nandi, and Yu Sasaki. Elastic-tweak: A framework for short tweak tweakable block cipher. In *Progress in Cryptology - INDOCRYPT 2021, Proceedings*, pages 114–137, 2021.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *ACM STOC 1998, Proceedings*, pages 209–218, 1998.

[CJPS22] Benoît Cogliati, Jérémy Jean, Thomas Peyrin, and Yannick Seurin. A long tweak goes a long way: High multi-user security authenticated encryption from tweakable block ciphers. *IACR Cryptol. ePrint Arch.*, page 846, 2022.

[CLL22] Wonseok Choi, Jooyoung Lee, and Yeongmin Lee. Building prfs from tprps: Beyond the block and the tweak length bounds. *IACR Cryptol. ePrint Arch.*, page 918, 2022.

[CLS15] Benoit Cogliati, Rodolphe Lampe, and Yannick Seurin. Tweaking even-mansour ciphers. In *Advances in Cryptology - CRYPTO 2015, Proceedings, Part I*, pages 189–208, 2015.

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

[CS08] Debrup Chakraborty and Palash Sarkar. A general construction of tweakable block ciphers and different modes of operations. *IEEE Trans. Information Theory*, 54(5):1991–2006, 2008.

[CS14] Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In *Advances in Cryptology - EUROCRYPT 2014, Proceedings*, pages 327–350, 2014.

[CS15a] Benoît Cogliati and Yannick Seurin. Beyond-birthday-bound security for tweakable even-mansour ciphers with linear tweak and key mixing. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015, Proceedings, Part II*, pages 134–158, 2015.

[CS15b] Benoit Cogliati and Yannick Seurin. On the provable security of the iterated even-mansour cipher against related-key and chosen-key attacks. In *Advances in Cryptology - EUROCRYPT 2015, Proceedings, Part I*, pages 584–613, 2015.

[CS16] Benoît Cogliati and Yannick Seurin. Strengthening the known-key security notion for block ciphers. In Thomas Peyrin, editor, *Fast Software Encryption - FSE 2016, Revised Selected Papers*, pages 494–513, 2016.

[DSST17] Yuanxi Dai, Yannick Seurin, John P. Steinberger, and Aishwarya Thiruvengadam. Indifferentiability of iterated even-mansour ciphers with non-idealized key-schedules: Five rounds are necessary and sufficient. In *Advances in Cryptology - CRYPTO 2017, Proceedings, Part III*, pages 524–555, 2017.

[Dut20] Avijit Dutta. Minimizing the two-round tweakable even-mansour cipher. In *Advances in Cryptology – ASIACRYPT 2020, Proceedings*, pages 601–629, 2020.

[GJMN16] Robert Granger, Philipp Jovanovic, Bart Mennink, and Samuel Neves. Improved masking for tweakable blockciphers with applications to authenticated encryption. In *Advances in Cryptology - EUROCRYPT 2016, Proceedings, Part I*, pages 263–293, 2016.

[GL16] Chun Guo and Dongdai Lin. Indifferentiability of 3-round even-mansour with random oracle key derivation. *IACR Cryptol. ePrint Arch.*, page 894, 2016.

[GLN19] Tony Grochow, Eik List, and Mridul Nandi. Dovemac: A tbc-based PRF with smaller state, full security, and high rate. *IACR Trans. Symmetric Cryptol.*, 2019(3):43–80, 2019.

[HT16] Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In *Advances in Cryptology - CRYPTO 2016, Proceedings, Part I*, pages 3–32, 2016.

[IMPS17] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A fast tweakable block cipher mode for highly secure message authentication. In *Advances in Cryptology - CRYPTO 2017, Proceedings, Part III*, pages 34–65, 2017.

[JLM+17] Ashwin Jha, Eik List, Kazuhiko Minematsu, Sweta Mishra, and Mridul Nandi. XHX - A framework for optimally secure tweakable block ciphers from classical block ciphers and universal hashing. In *Progress in Cryptology - LATINCRYPT 2017, Revised Selected Papers*, pages 207–227, 2017.

[JN18] Ashwin Jha and Mridul Nandi. On rate-1 and beyond-the-birthday bound secure online ciphers using tweakable block ciphers. *Cryptography and Communications*, 10(5):731–753, 2018.

[JN20] Ashwin Jha and Mridul Nandi. Tight security of cascaded LRW2. *J. Cryptol.*, 33(3):1272–1317, 2020.

[JNP14] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and keys for block ciphers: The TWEAKEY framework. In *Advances in Cryptology - ASIACRYPT 2014, Proceedings, Part II*, pages 274–288, 2014.

[KR07] Lars R. Knudsen and Vincent Rijmen. Known-key distinguishers for some block ciphers. In *Advances in Cryptology - ASIACRYPT 2007, Proceedings*, pages 315–324, 2007.

[LL18] ByeongHak Lee and Jooyoung Lee. Tweakable block ciphers secure beyond the birthday bound in the ideal cipher model. In *Advances in Cryptology - ASIACRYPT 2018, Proceedings, Part I*, pages 305–335, 2018.

[LPS12] Rodolphe Lampe, Jacques Patarin, and Yannick Seurin. An asymptotically tight security analysis of the iterated even-mansour cipher. In *Advances in Cryptology - ASIACRYPT 2012, Proceedings*, pages 278–295, 2012.

[LRW11] Moses Liskov, Ronald L. Rivest, and David A. Wagner. Tweakable block ciphers. *J. Crypto.*, 24(3):588–613, 2011.

[LS13a] Rodolphe Lampe and Yannick Seurin. How to construct an ideal cipher from a small set of public permutations. In *Advances in Cryptology - ASIACRYPT 2013, Proceedings, Part I*, pages 444–463, 2013.

[LS13b] Rodolphe Lampe and Yannick Seurin. Tweakable blockciphers with asymptotically optimal security. In *Fast Software Encryption - FSE 2013, Revised Selected Papers*, pages 133–151, 2013.

[LS14] Rodolphe Lampe and Yannick Seurin. Security analysis of key-alternating feistel ciphers. In *Fast Software Encryption - FSE 2014, Revised Selected Papers*, pages 243–264, 2014.

[LST12] Will Landecker, Thomas Shrimpton, and R. Seth Terashima. Tweakable blockciphers with beyond birthday-bound security. In *Advances in Cryptology - CRYPTO 2012, Proceedings*, pages 14–30, 2012.

[Men15a] Bart Mennink. Optimally secure tweakable blockciphers. In *Fast Software Encryption - FSE 2015, Revised Selected Papers*, pages 428–448, 2015.

[Men15b] Bart Mennink. Optimally secure tweakable blockciphers. *IACR Cryptology ePrint Archive*, 2015:363, 2015.

[Men18] Bart Mennink. Towards tight security of cascaded LRW2. In *Theory of Cryptography - TCC 2018, Proceedings, Part II*, pages 192–222, 2018.

[Min06] Kazuhiko Minematsu. Improved security analysis of XEX and LRW modes. In *Selected Areas in Cryptography - SAC 2006, Revised Selected Papers*, pages 96–113, 2006.

[Min09] Kazuhiko Minematsu. Beyond-birthday-bound security based on tweakable block cipher. In *Fast Software Encryption - FSE 2009, Revised Selected Papers*, pages 308–326, 2009.

[MPS12] Avradip Mandal, Jacques Patarin, and Yannick Seurin. On the public indifferentiability and correlation intractability of the 6-round feistel construction. In *Theory of Cryptography - TCC 2012, Proceedings*, pages 285–302, 2012.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In *Theory of Cryptography - TCC 2004, Proceedings*, pages 21–39, 2004.

[Nai15] Yusuke Naito. Full prf-secure message authentication code based on tweakable block cipher. In *Provable Security - ProvSec 2015, Proceedings*, pages 167–182, 2015.

[NI19] Ryota Nakamichi and Tetsu Iwata. Iterative block ciphers from tweakable block ciphers with long tweaks. *IACR Trans. Symmetric Cryptol.*, 2019(4):54–80, 2019.

[Pat08] Jacques Patarin. The "coefficients H" technique. In *Selected Areas in Cryptography - SAC 2008, Revised Selected Papers*, pages 328–345, 2008.

[PS16] Thomas Peyrin and Yannick Seurin. Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers. In *Advances in Cryptology - CRYPTO 2016, Proceedings, Part I*, pages 33–63, 2016.

[Rog04] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In *Advances in Cryptology - ASIACRYPT 2004, Proceedings*, pages 16–31, 2004.

[RZ11] Phillip Rogaway and Haibin Zhang. Online ciphers from tweakable blockciphers. In *Topics in Cryptology - CT-RSA 2011, Proceedings*, pages 237–249, 2011.

[XDG22] Shanjie Xu, Qi Da, and Chun Guo. Minimizing even-mansour ciphers for sequential indifferentiability (without key schedules). In *Progress in Cryptology - INDOCRYPT 2022, Proceedings*, pages 125–145, 2022.