Tight Security of TNT and Beyond

Attacks, Proofs and Possibilities for the Cascaded LRW Paradigm *

Ashwin Jha[1], Mustafa Khairallah[2], Mridul Nandi[3], and Abishanka Saha[3]

[1]CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
ashwin.jha@cispa.de
[2]Seagate Research Group, Singapore, Singapore
mustafa.khairallah@seagate.com
[3]Indian Statistical Institute, Kolkata, India
mridul.nandi@gmail.com,sahaa.1993@gmail.com

Abstract. Liskov, Rivest and Wagner laid the theoretical foundations for tweakable block ciphers (TBC). In a seminal paper, they proposed two (up to) birthday-bound secure design strategies — LRW1 and LRW2— to convert any block cipher into a TBC. Several of the follow-up works consider cascading of LRW-type TBCs to construct beyond-the-birthday bound (BBB) secure TBCs. Landecker et al. demonstrated that just two-round cascading of LRW2 can already give a BBB security. Bao et al. undertook a similar exercise in context of LRW1 with TNT— a three-round cascading of LRW1— that has been shown to achieve BBB security as well. In this paper, we present a CCA distinguisher on TNT that achieves a non-negligible advantage with $O(2^{n/2})$ queries, directly contradicting the security claims made by the designers. We provide a rigorous and complete advantage calculation coupled with experimental verifications that further support our claim. Next, we provide new and simple proofs of birthday-bound CCA security for both TNT and its single-key variant, which confirm the tightness of our attack. Furthering on to a more positive note, we show that adding just one more block cipher call, referred as 4-LRW1, does not just reestablish the BBB security, but also amplifies it up to $2^{3n/4}$ queries. As a side-effect of this endeavour, we propose a new abstraction of the cascaded LRW-design philosophy, referred to as the LRW+ paradigm, comprising two block cipher calls sandwiched between a pair of tweakable universal hashes. This helps us to provide a modular proof approach covering all cascaded LRW constructions with at least 2 rounds, including 4-LRW1, and its more established relative, the well-known CLRW2, or more aptly, 2-LRW2.

Keywords: TNT, LRW1, 4-LRW1, CLRW2, birthday-bound attack

* This article is an amalgamation and extension of prior work of the same authors. Concretely, it combines and significantly extends the contents of IACR ePrint articles 2023/1212 (by Khairallah), and 2023/1233 (by Jha, Nandi, and Saha) that appeared in August 2023 on closely related topics into a single edited document. This article should be seen as a successor of both these IACR ePrint articles.

1 Introduction

Tweakable Block Cipher or TBC is a highly versatile symmetric-key primitives that has found applications in almost all verticals of modern information security, including encryption schemes [7], message authentication codes [19], authenticated encryption [23,35], and even leakage resillience [39]. The popularity of TBCs is largely credited to the simplicity of TBC-based constructions, and more importantly, comparatively simpler proofs of beyond-the-birthday bound (BBB) security.

In a seminal paper [27] at CRYPTO 2002, Liskov, Rivest, and Wagner (LRW) formalized the notion of tweakable block ciphers (TBCs), although the high level idea already appeared in some AES candidates such as Hasty Pudding [38] and Misty [10]. Over the years, the design landscape of TBCs has changed progressively. The design of a TBC mainly falls into one of the two categories: adhoc designs based ob well-established primitive design paradigms, or provably secure designs based on block ciphers or cryptographic permutations. In recent years, the popularity of adhoc designs has gained momentum with the advent of the TWEAKEY framework [20], its chief example being Deoxys-TBC [21], Skinny [5] and Qarma [1]. These designs are built from scratch, and their security mainly depends on cryptanalysis. On the other hand, the security of provably secure designs is directly linked to the security of the underlying primitives, such as a block cipher, a permutation, or a pseudorandom function. Some prominent examples include LRW's original constructions [27] LRW1 and LRW2, XEX [37] by Rogaway, and its extensions by Chakraborty and Sarkar [8], Minematsu [30], and Granger et al. [14]. Note that, all these schemes are inherently birthday bound secure due to detectable internal collisions.

Cascading LRW2: Landecker et al. were the first to notice [25] that a cascading of two independent instances of LRW2 results in a BBB secure TBC construction. They proved that 2-round cascaded LRW2 is secure up to approx. $2^{2n/3}$ CCA queries, where n denotes the block size in bits. The initial proof was flawed [36], and superseded by a corrected proof by both Landecker et al. and Procter [36]. The construction was later found [28,22] to be tightly secure up to $2^{3n/4}$ CCA queries. For any arbitrary $r \geq 2$-round independent cascading of LRW2, denoted r-LRW2, Lampe and Seurin proved [24] CCA security up to approx. $2^{\frac{rn}{r+2}}$ queries.

Cascading LRW1: The idea to cascade LRW1 came quite later in [2], where Bao et al. showed that 3-round cascading of LRW1, referred as TNT, is CCA secure up to $2^{2n/3}$ queries. The design is highly appreciated in the community for its simple design and high provable security guarantee. In fact, the CPA security was later improved to $2^{3n/4}$ queries, essentially matching the bound for 2-round LRW1. Since this later result, it is widely believed that the CPA improvement carries over to the CCA setting as well. For the more general case of arbitrary $r \geq 3$, denoted r-LRW1, Zhang et al. proved [40] CCA security up to approx. $2^{\frac{r-1}{r+1}n}$ queries.

1.1 Motivation

The primary motivation behind this work is a peculiar non-random behavior exhibited by TNT in the CCA setting.

Suppose $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3$ are three independent random permutations of $\{0,1\}^n$. The TNT construction (see Fig. 1) based on $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3$ is a TBC with n-bit tweak and n-bit block input, defined by the mapping

$$(t, m) \xmapsto{\mathsf{TNT}} \boldsymbol{\pi}_3(t \oplus \boldsymbol{\pi}_2(t \oplus \boldsymbol{\pi}_1(m))).$$

As can be noticed by the definition of TNT, it has a peculiar property, that

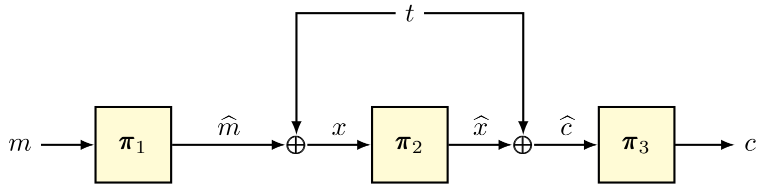

Fig. 1: The TNT construction [2].

we refer as the *final-block cancellation* property. Specifically, suppose we have a triple (t, m, c) such that $\mathsf{TNT}(t, m) = c$. Then, it is easy to see that any inverse query of the form (t', c) would result in a cancellation of the call to $\boldsymbol{\pi}_3$, and this is independent of the tweak values t and $t' = t \oplus \delta$. Essentially, the construction boils down to the one in Fig. 2. Let's call it TNT_δ for some fixed $\delta \neq 0^n$. Now,

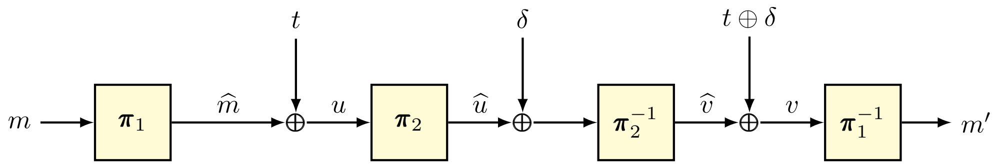

Fig. 2: TNT with final-block cancellation.

suppose the adversary can find a pair of tweaks (t_1, t_2) such that for a fixed message m, there is a collision at the output, i.e.,

$$(m'_1 = m'_2) \iff (v_1 = v_2) \iff (\widehat{v}_1 \oplus \widehat{v}_2 = t_1 \oplus t_2 = u_1 \oplus u_2)$$

So, an output collision happens if and only if $\widehat{v}_1 \oplus \widehat{v}_2 = u_1 \oplus u_2$. Interestingly, for TNT_δ, we have the following property:

$$(\widehat{u}_1 \oplus \widehat{u}_2 = \delta) \implies (\widehat{v}_1 \oplus \widehat{v}_2 = u_1 \oplus u_2),$$

which implies that there are two sources of collisions in TNT_δ. A collision happens whenever $\widehat{u}_1 \oplus \widehat{u}_2 = \delta$, or $\widehat{u}_1 \oplus \widehat{u}_2 \neq \delta$ and $\widehat{v}_1 \oplus \widehat{v}_2 = u_1 \oplus u_2$. This indicates that one can expect more collisions in TNT_δ as compared to a random function.

1.2 Contributions

Our contributions are threefold:

1. Birthday-bound CCA Attack on TNT: In section 3, we start by giving a heuristic distinguisher using the previously mentioned non-random behavior of TNT. We provide a heuristic analysis of this distinguisher using random permutation statistics and an analysis of the behaviour of difference equations and difference distribution tables (DDTs) of random permutations. Our analysis strongly indicates a global non-random phenomena that can be detected in roughly $O(2^{n/2}$ CCA queries. We verify these abnormal statistics experimentally on small instances of TNT. Based on the heuristics and experimental verification, we identify and exploit the final-block cancellation property of TNT, to furnish a formal CCA distinguisher between TNT and uniform tweakable random permutation. We provide rigorous analysis of the query complexity and advantage of our distinguisher, which clearly shows that the distinguisher achieves a non-negligible advantage using $O(2^{n/2})$ CCA queries.
   Since the attack clearly contradicts the security claims of the designers of TNT, we study their security proof in Appendix A and identify a bug, where a random variable is erroneously assumed to have a uniform distribution, leading to an over estimation of the security.
2. Birthday-bound CCA Security of TNT: In section 4, we provide a simple proof of birthday-bound CCA security for TNT. Note that, the CCA security bound also follows from the results in [40]. Nevertheless, given the flaws in TNT's original analysis, we believe that multiple security proofs using different techniques will lead to a greater confidence in the revised security claim. In addition to the original TNT, we also analyze the single-keyed variant of TNT, and show that it retains the same level of CCA security as well.
3. A Generalization of Cascaded LRW Paradigm: In a more abstract direction, in section 5, we present a generalized view of the cascaded LRW design strategy for any arbitrary number of rounds $r \geq 2$, called the LRW+ construction. It consists of two block cipher calls sandwiched between a pair of tweakable universal hashes. We show that as long as the tweakable hashes are sufficiently[1] universal, the LRW+ construction is CCA secure up to $2^{3n/4}$ queries. Note that, LRW+ encompasses both 2-LRW2 and 4-LRW1. Thus, as a direct side-effect of our analysis, in section 6, we show that 2-LRW2 and 4-LRW1 are CCA secure up to $2^{3n/4}$ queries. In case of 2-LRW2, our bound matches the tight analysis in [22], and in case of 4-LRW1, we provide

[1] Having approx. 2^{-n}-AU bound.

a significant improvement over an independent and concurrent result [13], which only guarantees security up to $2^{2n/3}$ queries.
Note that, the result on $\mathsf{LRW+}$ directly shows that r-$\mathsf{LRW1}$ is at least $3n/4$-bit secure for any $r \geq 4$, improving on the results for $r \leq 8$. Similarly, for r-$\mathsf{LRW2}$ it shows at least $3n/4$-bit security for any $r \geq 2$, improving on the results for $r \leq 6$. See Table 1 for a summary of the state-of-the-art on the security of cascaded LRW constructions.

Table 1: Summary of security bounds for LRW based construction. We have assumed all hash functions to be 2^{-n}-(XOR) universal. The bottom four rows present our results. $\mathsf{LRW+}$ generalizes both 2-$\mathsf{LRW2}$ and 4-$\mathsf{LRW1}$. So the bound on $\mathsf{LRW+}$ implies similar bounds for 2-$\mathsf{LRW2}$ and 4-$\mathsf{LRW1}$.

Construction	BC calls	Hash calls	Security bound	Tightness
LRW1 [27]	1	0	$2^{n/2}$ (CPA) [27]	✓
LRW2 [27]	1	1	$2^{n/2}$ [27]	✓
3-LRW1 (TNT [15])	3	0	$2^{2n/3}$ [15]	(flawed)
4-LRW1	4	0	$2^{2n/3}$ [13]	–
2-LRW2 (CLRW2 [25])	2	2	$2^{3n/4}$ [22]	✓ [28]
r-LRW1 [40]	r odd	0	$2^{\frac{r-1}{r+1}n}$ [40]	–
	r even		$2^{\frac{r-2}{r}n}$	–
r-LRW2 [24]	r odd	r	$2^{\frac{r-1}{r+1}n}$ [24]	–
	r even		$2^{\frac{r}{r+2}n}$	–
3-LRW1 (TNT)	3	0	$2^{n/2}$	✓
1k-TNT	3	0	$2^{n/2}$	✓
LRW+	2	2^*	$2^{3n/4}$	–
4-LRW1	4	0	$2^{3n/4}$	–

A Note on the Impact of Our Birthday-bound Attack: As mentioned before, the authors of [2] claimed the CCA security of TNT to be $2n/3$ bits. In Asiacrypt 2020, the authors of [16] conjectured that the CCA security of TNT is probably $3n/4$ bits. In [41], the authors have stated:

> A natural open problem is the exact security of r-$\mathsf{LRW1}$. Unlike , exact security of r-$\mathsf{LRW1}$ for $r = 3$ already appears challenging, and might require new proof approaches.

We believe this work answers a critical research question of both practical and theoretical implications. On one hand, it studies the exact security of an efficient construction that has several practical applications. On the other hand, it offers another cautionary tale on how to use statistical proof techniques such as the χ^2 method.[2]

Additionally, the attack applies to practical instances of TNT: TNT-AES in [2] and TNT-SM4-128 in [17]. The authors of [17] also introduced TNT-SM4-32, where the tweak size is limited to 32-bits. Our distinguisher requires $O(2^{n/2})$ tweaks, where $n = 128$ in case of TNT-SM4. Hence, the distinguisher directly applies to TNT-SM4-128, which has a tweak size of 128 bits. It does not directly apply to TNT-SM4-32, since the tweak space is too small. However, since our distinguisher breaks the the BBB security proof in [2], the exact security of TNT-SM4-32 and whether it is has BBB security is an open question.

We note that in Eurocrypt 2023, a full-round distinguisher on TNT-AES using truncated boomerang attacks was presented in [4]. However, the attack is particular to TNT-AES and requires almost 2^n queries. Our attack, applied to any 128-bit instantiation of TNT, including TNT-AES, requires $\leq 2^{69}$ queries to have an almost 100% success rate, making it the best known distinguisher for any 128-bit TNT variant, without relying on the properties of the underlying block cipher. We sum up all known distinguishers on TNT-AES in Table 2, which indicates that our distinguisher is not only theoretical, but outperforms all cryptanalytic efforts on TNT, so far.

Table 2: Known distinguishers against TNT-AES. CCA stands for adaptive Chosen Ciphertext Adversary. NCPA stands for Non-adaptive Chosen Plaintext Adversary. **Rounds** is the number of AES rounds in π_1, π_2 and π_3, respectively. $\star$ means any number of rounds. Generic attacks do not reply on any AES properties and apply to TNT instantiated with any 128-bit block cipher. 2^{69} is the complexity for which our attack is expected to have 100% success rate, while 2^{68} is expected to have 99% success rate.

Ref.	**Type**	**Data**	**Time**	**Adversary**	**Rounds**
[2]	Boomerang	2^{126}	2^{126}	CCA	$\star - 5 - \star$
[16]	Impossible Differential	$2^{113.6}$	$2^{113.6}$	NCPA	$5 - \star - \star$
[16]	Generic	$2^{99.5}$	$2^{99.5}$	NCPA	$\star - \star - \star$
[4]	Truncated Boomerang	2^{76}	2^{76}	CCA	$\star - 5 - \star$
[4]	Truncated Boomerang	2^{87}	2^{87}	CCA	$5 - 5 - \star$
[4]	Truncated Boomerang	$2^{127.8}$	$2^{127.8}$	CCA	$\star - 6 - \star$
This paper	Generic	$\leq 2^{69}$	$\leq 2^{69}$	CCA	$\star - \star - \star$

[2] Refer to [6] for another example of erroneously estimated distributions.

2 Preliminaries

NOTATIONAL SETUP: For $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2, \ldots, n\}$, $\{0,1\}^n$ denotes the set of bit strings of length n, and $\mathsf{Perm}(n)$ denotes the set of all permutations over $\{0,1\}^n$. For $n, \tau \in \mathbb{N}$, $\widetilde{\mathsf{Perm}}(\tau, n)$ denotes the set of all families of permutations $\pi_t := \pi(t, \cdot) \in \mathsf{Perm}(n)$, indexed by $t \in \{0,1\}^\tau$. For $n, r \in \mathbb{N}$, such that $n \geq r$, we define the falling factorial $(n)_r := n!/(n-r)! = n(n-1)\cdots(n-r+1)$. We define $(n)_0 := 1$.

For $q \in \mathbb{N}$, x^q denotes the q-tuple $(x_1, x_2, \ldots, x_q)$, and in this context, $\mathtt{M}(x^q)$ and $\mathtt{S}(x^q)$ respectively denote the multiset and set corresponding to $\{x_i : i \in [q]\}$. For a set $\mathcal{I} \subseteq [q]$ and a q-tuple x^q, $x^{\mathcal{I}}$ denotes the tuple $(x_i)_{i \in \mathcal{I}}$. For a pair of tuples x^q and y^q, (x^q, y^q) denotes the 2-ary q-tuple $((x_1, y_1), \ldots, (x_q, y_q))$. An n-ary q-tuple is defined analogously. For $q \in \mathbb{N}$, for any set $\mathcal{X}$, $(\mathcal{X})_q$ denotes the set of all q-tuples with distinct elements from $\mathcal{X}$. For $q \in \mathbb{N}$, a 2-ary tuple (x^q, y^q) is called permutation compatible, denoted $x^q \leftrightsquigarrow y^q$, if $x_i = x_j \iff y_i = y_j$. Extending notations, a 3-ary tuple (t^q, x^q, y^q) is called tweakable permutation compatible, denoted by $(t^q, x^q) \leftrightsquigarrow (t^q, y^q)$, if $(t_i, x_i) = (t_j, x_j) \iff (t_i, y_i) = (t_j, y_j)$. For any tuple $x^q \in \mathcal{X}^q$, and for any function $f : \mathcal{X} \to \mathcal{Y}$, $f(x^q)$ denotes the tuple $(f(x_1), \ldots, f(x_q))$. We use short hand notation $\exists^*$ to represent the phrase "there exists distinct".

Unless stated otherwise, upper and lower case letters denote variables and values, respectively, and Serif font letters are used to denote random variables. For a finite set $\mathcal{X}$, $\mathsf{X} \leftarrow\!\!\$\ \mathcal{X}$ denotes the uniform and random sampling of X from $\mathcal{X}$. We write $\mathsf{X}^q \xleftarrow{\text{wor}} \mathcal{X}$ to denote WOR (without replacement sampling) of a q-tuple X^q from the set $\mathcal{X}$, where $|\mathcal{X}| \geq q$ is obvious. More precisely, $\mathsf{X}^q \leftarrow\!\!\$\ (\mathcal{X})_q$.

2.1 Some Useful Inequalities

Definition 1 ([22]). *For $r \geq s$, let $a = (a_i)_{i \in [r]}$ and $b = (b_j)_{j \in [s]}$ be two sequences over $\mathbb{N}$. We say that a compresses to b, if there exists a partition $\mathcal{P}$ of $[r]$ such that $\mathcal{P}$ contains exactly s cells, say $\mathcal{P}_1, \ldots, \mathcal{P}_s$, and $\forall i \in [s]$, $\ b_i = \sum_{j \in \mathcal{P}_i} a_j$.*

Proposition 1 ([22]). *For $r \geq s$, let $a = (a_i)_{i \in [r]}$ and $b = (b_j)_{j \in [s]}$ be sequences over $\mathbb{N}$, such that a compresses to b. Then for any $n \in \mathbb{N}$, such that $2^n \geq \sum_{i=1}^r a_i$, we have $\prod_{i=1}^r (2^n)_{a_i} \geq \prod_{j=1}^s (2^n)_{b_j}$.*

Proposition 2 ([22]). *For $r \geq 2$, let $c = (c_i)_{i \in [r]}$ and $d = (d_i)_{i \in [r]}$ be two sequences over $\mathbb{N}$. Let $a_1, a_2, b_1, b_2 \in \mathbb{N}$, such that $c_i \leq a_j$, $c_i + d_i \leq a_j + b_j$ for all $i \in [r]$ and $j \in [2]$, and $\sum_{i=1}^r d_i = b_1 + b_2$. Then, for any $n \in \mathbb{N}$, such that $a_j + b_j \leq 2^n$ for $j \in [2]$, we have $\prod_{i=1}^r (2^n - c_i)_{d_i} \geq (2^n - a_1)_{b_1} (2^n - a_2)_{b_2}$.*

2.2 (Tweakable) Block Ciphers and Random Permutations

A block cipher with key size κ and block size n is a family of permutations $E \in \widetilde{\mathsf{Perm}}(\kappa, n)$. For $k \in \{0,1\}^\kappa$, we denote $E_k(\cdot) := E(k, \cdot)$, and $E_k^{-1}(\cdot) := E^{-1}(k, \cdot)$.

A tweakable block cipher with key size κ, tweak size τ and block size n is a family of permutations $\widetilde{E} \in \widetilde{\mathsf{Perm}}(\kappa\tau, n)$. For $k \in \{0,1\}^\kappa$ and $t \in \{0,1\}^\tau$, we denote $\widetilde{E}_k(t, \cdot) := \widetilde{E}(k, t, \cdot)$, and $\widetilde{E}_k^{-1}(t, \cdot) := \widetilde{E}^{-1}(k, t, \cdot)$. Throughout this paper, we fix $\kappa, \tau, n \in \mathbb{N}$ as the key size, tweak size and block size, respectively, of the given (tweakable) block cipher.

We say that $\boldsymbol{\pi}$ is an (ideal) random permutation on block space $\{0,1\}^n$ to indicate that $\boldsymbol{\pi} \leftarrow\!\!\$\ \mathsf{Perm}(n)$. Similarly, we say that $\widetilde{\boldsymbol{\pi}}$ is an (ideal) tweakable random permutation on tweak space $\{0,1\}^\tau$ and block space $\{0,1\}^n$ to indicate that $\widetilde{\boldsymbol{\pi}} \leftarrow\!\!\$\ \widetilde{\mathsf{Perm}}(\tau, n)$.

2.3 (T)SPRP Security Definitions

In this paper, we assume that the distinguisher 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. Let $\mathbb{A}(q, t)$ be the class of all non-trivial distinguishers limited to q oracle queries, and t computations. In our analyses, especially security proofs, it will be convenient to work in the information-theoretic setting. Accordingly, we always skip the boilerplate hybrid steps, and often assume that the adversary is computationally unbounded, i.e., $t = \infty$, and deterministic.

A computational equivalent of all our security proofs can be easily obtained by a simple hybrid argument.

(Tweakable) Strong Pseudorandom Permutation (SPRP): The SPRP advantage of distinguisher $\mathscr{A}$ against E instantiated with a key $\mathsf{K} \leftarrow\!\!\$\ \{0,1\}^\kappa$ is defined as

$$\mathbf{Adv}_E^{\mathsf{sprp}}(\mathscr{A}) = \mathbf{Adv}_{E^\pm;\boldsymbol{\pi}^\pm}(\mathscr{A}) := \left| \mathsf{Pr}\left(\mathscr{A}^{E_\mathsf{K}^\pm} = 1\right) - \mathsf{Pr}\left(\mathscr{A}^{\boldsymbol{\pi}^\pm} = 1\right) \right|. \quad (1)$$

The SPRP security of E is defined as $\mathbf{Adv}_E^{\mathsf{sprp}}(q, t) := \max_{\mathscr{A} \in \mathbb{A}(q,t)} \mathbf{Adv}_E^{\mathsf{sprp}}(\mathscr{A})$. Similarly, the TSPRP advantage of distinguisher $\mathscr{A}$ against $\widetilde{E}$ instantiated with a key $\mathsf{K} \leftarrow\!\!\$\ \{0,1\}^\kappa$ is defined as

$$\mathbf{Adv}_{\widetilde{E}}^{\mathsf{tsprp}}(\mathscr{A}) = \mathbf{Adv}_{\widetilde{E}^\pm;\widetilde{\boldsymbol{\pi}}^\pm}(\mathscr{A}) := \left| \mathsf{Pr}\left(\mathscr{A}^{\widetilde{E}_\mathsf{K}^\pm} = 1\right) - \mathsf{Pr}\left(\mathscr{A}^{\widetilde{\boldsymbol{\pi}}^\pm} = 1\right) \right|. \quad (2)$$

The TSPRP security of $\widetilde{E}$ is defined as $\mathbf{Adv}_{\widetilde{E}}^{\mathsf{tsprp}}(q, t) := \max_{\mathscr{A} \in \mathbb{A}(q,t)} \mathbf{Adv}_{\widetilde{E}}^{\mathsf{tsprp}}(\mathscr{A})$.

2.4 The Expectation Method

Let $\mathscr{A}$ be a computationally unbounded and deterministic distinguisher that tries to distinguish between two oracles $\mathcal{O}_0$ and $\mathcal{O}_1$ via black box interaction with one of them. We denote the query-response tuple of $\mathscr{A}$'s interaction with its oracle by a transcript ω. This may also include any additional information the oracle chooses to reveal to the distinguisher at the end of the query-response phase of the game. We denote by Θ_1 (res. Θ_0) the random transcript variable when $\mathscr{A}$ interacts with $\mathcal{O}_1$ (res. $\mathcal{O}_0$). The probability of realizing a given transcript ω in the security game with an oracle $\mathcal{O}$ is known as the *interpolation probability* of ω with respect to $\mathcal{O}$. Since $\mathscr{A}$ is deterministic, this probability depends only on the oracle $\mathcal{O}$ and the transcript ω. A transcript ω is said to be *attainable* if $\mathsf{Pr}(\Theta_0 = \omega) > 0$. The expectation method [18] (stated below) is a generalization of Patarin's H-coefficients technique [33], which is quite useful in obtaining improved bounds in many cases [18,22].

Lemma 1 (Expectation Method [18]). *Let Ω be the set of all transcripts. For some $\epsilon_{\mathsf{bad}} \geq 0$ and a non-negative function $\epsilon_{\mathsf{ratio}} : \Omega \to [0, \infty)$, suppose there is a set $\Omega_{\mathsf{bad}} \subseteq \Omega$ satisfying the following:*

- $\mathsf{Pr}(\Theta_0 \in \Omega_{\mathsf{bad}}) \leq \epsilon_{\mathsf{bad}}$;
- *For any $\omega \notin \Omega_{\mathsf{bad}}$, ω is attainable and* $\dfrac{\mathsf{Pr}(\Theta_1 = \omega)}{\mathsf{Pr}(\Theta_0 = \omega)} \geq 1 - \epsilon_{\mathsf{ratio}}(\omega)$.

Then for any distinguisher $\mathscr{A}$ trying to distinguish between $\mathcal{O}_1$ and $\mathcal{O}_0$, we have the following bound on its distinguishing advantage:

$$\mathbf{Adv}_{\mathcal{O}_1;\mathcal{O}_0}(\mathscr{A}) \leq \epsilon_{\mathsf{bad}} + \mathsf{Ex}(\epsilon_{\mathsf{ratio}}(\Theta_0)).$$

When $\epsilon_{\mathsf{ratio}}$ is a constant function, we get the following corollary of the expectation method, otherwise known as the H-coefficients technique.

Corollary 1 (H-coefficients Technique [33]). *Let Ω be the set of all transcripts. For some $\epsilon_{\mathsf{bad}} \geq 0$ and $\epsilon_{\mathsf{ratio}} \geq 0$, suppose there is a set $\Omega_{\mathsf{bad}} \subseteq \Omega$ satisfying the following:*

- $\mathsf{Pr}(\Theta_0 \in \Omega_{\mathsf{bad}}) \leq \epsilon_{\mathsf{bad}}$;
- *For any $\omega \notin \Omega_{\mathsf{bad}}$, ω is attainable and* $\dfrac{\mathsf{Pr}(\Theta_1 = \omega)}{\mathsf{Pr}(\Theta_0 = \omega)} \geq 1 - \epsilon_{\mathsf{ratio}}$.

Then for any distinguisher $\mathscr{A}$ trying to distinguish between $\mathcal{O}_1$ and $\mathcal{O}_0$, we have the following bound on its distinguishing advantage:

$$\mathbf{Adv}_{\mathcal{O}_1;\mathcal{O}_0}(\mathscr{A}) \leq \epsilon_{\mathsf{bad}} + \epsilon_{\mathsf{ratio}}.$$

2.5 Some Results on Universal Hash Functions

An (s, n)-hash function family $\mathcal{H}$, is a family of functions $\{h : \{0,1\}^s \to \{0,1\}^n\}$, keyed implicitly by the choice of h. A pair of distinct elements (t, t') from $\{0,1\}^s$ is said to be colliding for a function $h \in \mathcal{H}$, if $h(t) = h(t')$. An (s, n)-hash function family $\mathcal{H}$ is called an ϵ-amost universal hash family (AUHF) if for all $t \neq t' \in \{0,1\}^s$,

$$\mathsf{Pr}(\mathsf{H} \leftarrow\!\!\$\, \mathcal{H} \;:\; \mathsf{H}(t) = \mathsf{H}(t')) \leq \epsilon. \tag{3}$$

Throughout this section, we fix $t^q = (t_1, \ldots, t_q) \in (\mathcal{T})_q$. For a randomly chosen hash function $\mathsf{H} \leftarrow\!\!\$\, \mathcal{H}$, the probability of having at least one colliding pair in t^q is at most $\binom{q}{2} \cdot \epsilon$. This is straightforward from the union bound.

Lemma 2 (Alternating Collisions Lemma [22]). *Suppose* $\mathsf{H}_1, \mathsf{H}_2$ *are two uniformly and independently drawn functions from an* ϵ*-AUHF* $\mathcal{H}$ *and* $t^q \in (\{0,1\}^s)_q$*. Then,*

$$\Pr\left(\exists^* i,j,k,l \in [q], \mathsf{H}_1(t_i) = \mathsf{H}_1(t_j) \wedge \mathsf{H}_1(t_k) = \mathsf{H}_1(t_l) \wedge \mathsf{H}_2(t_j) = \mathsf{H}_2(t_k)\right) \leq q^2 \epsilon^{1.5}.$$

Lemma 3 (Alternating Events Lemma [22]). *Let* $\mathsf{X}^q = (\mathsf{X}_1, \ldots, \mathsf{X}_q)$ *be a* q*-tuple of random variables. Suppose for all* $i < j \in [q]$, $\mathtt{E}_{i,j}$ *are events associated with* X_i *and* X_j*, possibly dependent. Each event holds with probability at most* ϵ*. Moreover, for any distinct* $i,j,k,l \in [q]$, $\mathtt{F}_{i,j,k,l}$ *are events associated with* X_i, X_j, X_k *and* X_l*, which holds with probability at most* ϵ'*. Moreover, the collection of events* $(\mathtt{F}_{i,j,k,l})_{i,j,k,l}$ *is independent with the collection of event* $(\mathtt{E}_{i,j})_{i,j}$*. Then,*

$$\Pr\left(\exists^* i,j,k,l \in [q], \mathtt{E}_{i,j} \ \wedge\ \mathtt{E}_{k,l} \ \wedge\ \mathtt{F}_{i,j,k,l}\right) \leq q^2 \cdot \epsilon \cdot \sqrt{\epsilon'}$$

Let $\mathsf{X}^q = \mathsf{H}(t^q)$. We define an equivalence relation $\sim$ on $[q]$ as: $\alpha \sim \beta$ if and only if $\mathsf{X}_\alpha = \mathsf{X}_\beta$ (i.e. $\sim$ is simply the multicollision relation). Let $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_r$ denote those equivalence classes of $[q]$ corresponding to $\sim$, such that $\nu_i = |\mathcal{P}_i| \geq 2$ for all $i \in [r]$.

Lemma 4 ([22]). *Let* C *denote the number of colliding pairs in* X^q*. Then, we have*

$$\mathsf{Ex}\left(\sum_{i=1}^{r} {\nu_i}^2\right) \leq 2q^2\epsilon.$$

Corollary 2 ([31,22]). *Let* $\nu_{\max} = \max\{\nu_i : i \in [r]\}$*. Then, for some* $a \geq 2$*, we have*

$$\Pr\left(\nu_{\max} \geq a\right) \leq \frac{2q^2\epsilon}{a^2}.$$

2.6 Patarin's Mirror Theory

We will use the Mennink and Neves interpretation [29] of mirror theory. For ease of understanding and notational coherency, we sometimes use different parametrization and naming conventions. Let $q \geq 1$ and let $\mathcal{L}$ be the system of linear equations

$$\{e_1 : Y_1 \oplus V_1 = \delta_1, \quad e_2 : Y_2 \oplus V_2 = \delta_2, \quad \ldots, \quad e_q : Y_q \oplus V_q = \delta_q\}$$

where Y^q and V^q are unknowns, and $\delta^q \in (\{0,1\}^n)^q$ are constants. In addition there are (in)equality restrictions on Y^q and V^q, which uniquely determine $\mathtt{S}Y^q$ and $\mathtt{S}V^q$. We assume that $\mathtt{S}(Y^q)$ and $\mathtt{S}(V^q)$, are indexed in an arbitrary order by the index sets $[q_Y]$ and $[q_V]$, where $q_Y = |\mathtt{S}(Y^q)|$ and $q_V = |\mathtt{S}(V^q)|$. This assumption is without any loss of generality as this does not affect the system $\mathcal{L}$. Given such an ordering, we can view $\mathtt{S}(Y^q)$ and $\mathtt{S}(V^q)$ as ordered sets $\{Y'_1, \ldots, Y'_{q_Y}\}$ and $\{V'_1, \ldots, V'_{q_V}\}$, respectively. We define two surjective index mappings:

$$\varphi_Y : \begin{cases} [q] \to [q_Y] \\ i \mapsto j \text{ if and only if } Y_i = Y'_j. \end{cases} \qquad \varphi_V : \begin{cases} [q] \to [q_V] \\ i \mapsto k \text{ if and only if } V_i = V'_k. \end{cases}$$

It is easy to verify that $\mathcal{L}$ is uniquely determined by $(\varphi_Y, \varphi_V, \delta^q)$, and vice-versa. Consider a labeled bipartite graph $\mathcal{G}(\mathcal{L}) = ([q_Y], [q_V], \mathcal{E})$ associated with $\mathcal{L}$, where $\mathcal{E} = \{(\varphi_Y(i), \varphi_V(i), \delta_i) : i \in [q]\}$, δ_i being the label of edge. Clearly, each equation in $\mathcal{L}$ corresponds to a unique labeled edge (assuming no duplicate equations). We give three definitions with respect to the system $\mathcal{L}$ using $\mathcal{G}(\mathcal{L})$.

Definition 2 (cycle-freeness). *$\mathcal{L}$ is said to be cycle-free if and only if $\mathcal{G}(\mathcal{L})$ is acyclic.*

Definition 3 ($\xi_{\max}$-component). *Two distinct equations (or unknowns) in $\mathcal{L}$ are said to be in the same component if and only if the corresponding edges (res. vertices) in $\mathcal{G}(\mathcal{L})$ are in the same component. The size of any component $\mathcal{C}$ in $\mathcal{L}$, denoted $\xi(\mathcal{C})$, is the number of vertices in the corresponding component of $\mathcal{G}(\mathcal{L})$, and the maximum component size is denoted by $\xi_{\max}(\mathcal{L})$ (or simply $\xi_{\max}$).*

Definition 4 (non-degeneracy). *$\mathcal{L}$ is said to be non-degenerate if and only if there does not exist a path of even length at least 2 in $\mathcal{G}(\mathcal{L})$ such that the labels along the edges on this path sum up to zero.*

Isolated and Star Components: In an edge-labeled bipartite graph $\mathcal{G} = (\mathcal{Y}, \mathcal{V}, \mathcal{E})$, an edge (y, v, δ) is called *isolated* edge if both y and v have degree 1. A component $\mathcal{S}$ of $\mathcal{G}$ is called star, if $\xi(\mathcal{S}) \geq 3$ and there exists a unique vertex v in $\mathcal{S}$ with degree $\xi(\mathcal{S}) - 1$. We call v the center of $\mathcal{S}$. Further, we call $\mathcal{S}$ a $\mathcal{Y}$-$\star$ (res. $\mathcal{V}$-$\star$) component if its center lies in $\mathcal{Y}$ (res. $\mathcal{V}$).

Mirror Theory for Tweakable Permutation Setting. Consider a system of equation $\mathcal{L}$

$$\{e_1 : Y_1 \oplus V_1 = \delta_1, \quad e_2 : Y_2 \oplus V_2 = \delta_2, \quad \ldots, \quad e_q : Y_q \oplus V_q = \delta_q\},$$

such that each component in $\mathcal{G}(\mathcal{L})$ is either an isolated edge or a star. Let c_1, c_2, and c_3 denote the number of components of isolated, $\mathcal{Y}$-$\star$, and $\mathcal{V}$-$\star$ types, respectively. Let q_1, q_2, and q_3 denote the number of equations of isolated, $\mathcal{Y}$-$\star$, and $\mathcal{V}$-$\star$ types, respectively. Therefore, $c_1 = q_1$. Note that the equations in $\mathcal{L}$ can be arranged in any arbitrary order without affecting the number of solutions. For the sake of simplicity, we fix the ordering in such a way that all isolated edges occur first, followed by the star components. Let $(\delta'_1, \delta'_2, \cdots, \delta'_s)$ be an arbitrary ordering of $\mathtt{S}(\delta^q)$, and for all $i \in [s]$, let ν_i denote the multiplicity of δ'_i in the multiset $\mathtt{M}(\delta^q)$, i.e., $s \leq q$ and $\sum_{i=1}^{s} \nu_i = q$.
In [22], Jha and Nandi proved the following result.

Theorem 1 ([22]). *Let $\mathcal{L}$ be the system of linear equations as described above with $q < 2^{n-2}$ and $\xi_{\max} q \leq 2^{n-1}$. Then, the number of tuples $(y_1, \ldots, y_{q_Y}, v_1, \ldots, v_{q_V})$ that satisfy $\mathcal{L}$, denoted h_q, such that $y_i \neq y_j$ and $v_i \neq v_j$, for all $i \neq j$, satisfies:*

$$h_q \geq \left(1 - \frac{13q^4}{2^{3n}} - \frac{2q^2}{2^{2n}} - \left(\sum_{i=1}^{c_2+c_3} \eta_{c_1+i}^2\right) \frac{4q^2}{2^{2n}}\right) \times \frac{(2^n)_{q_1+c_2+q_3} (2^n)_{q_1+q_2+c_3}}{\prod_{i \in [s]} (2^n)_{\nu_i}},$$

where $\eta_j = \xi_j - 1$ and ξ_j denotes the size (number of vertices) of the j-th component, for all $j \in [c_1 + c_2 + c_3]$.

2.7 Poisson Distribution

The Poisson distribution is a discrete distribution with parameter λ and its Probability Mass Function (PMF) is defined as:

$$\text{Poisson}(i;\lambda) = \Pr[X = i] = \frac{\lambda^i e^{-\lambda}}{i!}$$

where the mean and variance are both equal to λ.

2.8 Difference Distribution Tables

Let $\pi : \{0,1\}^n \to \{0,1\}^n$ be a permutation. The equation

$$\pi(x \oplus \delta) \oplus \pi(x) = \Delta$$

is known as the difference equation (δ, Δ) over π, where $\delta, \Delta \in \{0,1\}^n$ and $\oplus$ is addition in the Galois Field GF(2^n). Since π is a permutation, then any difference equation must have an even number of solutions; either no solutions at all (0), or an even non-zero number of solutions. Note that if M is a solution for the difference equation (δ, Δ), then $M \oplus \delta$ must also be a solution. A Difference Distribution Table (DDT) is a $2^n \times 2^n$ table constructed by counting the number of solutions of each possible difference equation. It looks like Table 3. Each row or column adds up to 2^n and all the entries are even. The entry $(0,0)$ is always 2^n and the rest of the entries of the first row and first column are all zero. If all the entries are either 0 or 2^n, then the permutation is linear. If all the entries are either 0 or 2, except one, then the permutation is known as an Almost Perfect Non-linear (APN) permutation. A random permutation is likely to fall somewhere in between.

Table 3: An example of a DDT.

δ \ Δ	0	1	2	...	2^n-1
0	2^n	0	0	...	0
1	0	0	2	...	4
2	0	2	0	...	0
...	...	...	...	...	...
2^n-1	0	0	8	...	2

3 Cryptanalysis of TNT

In our discussions on TNT and cascaded LRW1, we always fix $\tau = n$. Hereafter, we only consider the TNT construction in information-theoretic setting.

Accordingly, we instantiate TNT based on three independent random permutations $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$, and $\boldsymbol{\pi}_3$ of $\{0,1\}^n$. Recall that, the TNT construction is defined by the mapping

$$(t, m) \xmapsto{\mathsf{TNT}} \boldsymbol{\pi}_3(t \oplus \boldsymbol{\pi}_2(t \oplus \boldsymbol{\pi}_1(m))), \tag{4}$$

3.1 A Non-Random Behavior in TNT

Consider a random function $F : \{0,1\}^n \to \{0,1\}^n$ that is constructed as

$$F(T) = \tilde{\pi}^{-1}(T \oplus \Delta, \tilde{\pi}(T, M_0)),$$

where $M_0 \in \{0,1\}^n$ and $\Delta \in \{0,1\}^n \setminus \{0^n\}$ are constants. It is easy to see that F is indistinguishable from a random function if $\tilde{\pi}$ is an ideal tweakable random permutation. We show in this section that if the same function is instantiated with TNT instead of $\tilde{\pi}$, it is distinguishable from a random function. This implies a distinguisher against the STPRP security of TNT. The distinguisher $\mathbf{D}$ is parameterized by the complexity q and a threshold $\theta(q)$. It makes q forward queries and q backward queries. It is described in Algorithm 1.

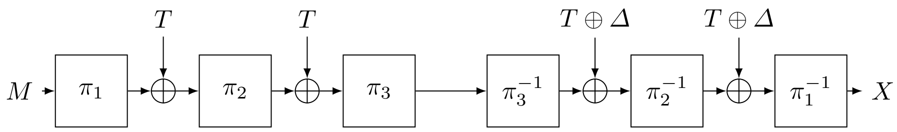

Fig. 3: One iteration of the distinguisher in Algorithm 1.

The description of the distinguisher is quite simple: *Cascade the forward and inverse queries, with tweaks T and $T \oplus \Delta$ where Δ and the plaintext M are fixed for all queries, and $\Delta \neq 0$. Make sure that for all $0 < i < q$, $0 < j < q$ and $i \neq j$, $T_i \neq T_j$ and $T_i \neq T_j \oplus \Delta$. Count the number of collisions at the output of backward queries.* One iteration of the distinguisher is visually depicted in Figure 3, and Figure 4 depicts the effective behavior as the effect of π_3 is removed and we have an XOR with a constant Δ between the forward and backward queries. Figure 5 shows the internal values in the effective trace during one iteration.

Analysis of the distinguisher In the ideal world, each query uses a unique tweak and a new uniform random permutation is sampled for each query. Hence, all the responses X are uniformly distributed. Given two queries, the probability of collision is $1/2^n$, and the behaviour follows the birthday collision search. The input space of the construction in Figure 4 is $\mathcal{T}$ and has size of 2^n possibilities. Thus, the expected number of collisions in the range of X can be estimated by

$$\frac{\binom{2^n}{2}}{2^n} = \frac{2^n - 1}{2}.$$

Algorithm 1 The distinguisher $\mathbf{D}$ against the CCA security of TNT.

$M \xleftarrow{\$} \{0,1\}^n$
$\mathit{\Delta} \leftarrow \mathtt{itob}_n(2^{n-1})$
$L \leftarrow [0 \;\forall\; 1 \le i \le 2^n]$
$\mathtt{coll} \leftarrow 0$
for $i \in \{0, 1, \cdots, q-1\}$ **do**
  $C \leftarrow \tilde{E}(\mathtt{itob}_n(i), M)$
  $X \leftarrow \tilde{E}^{-1}(\mathtt{itob}_n(i) \oplus \mathit{\Delta}, C)$
  $\mathtt{coll} \leftarrow \mathtt{coll} + L[\mathtt{btoi}(X)]$
  $L[\mathtt{btoi}(X)] \leftarrow L[\mathtt{btoi}(X)] + 1$
end for
if $\mathtt{coll} \ge \theta(q)$ **then**
  return 1
else
  return 0
end if

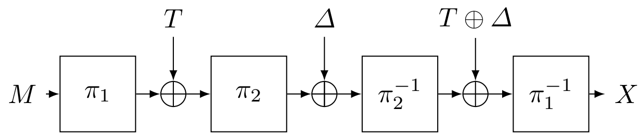

Fig. 4: The effective iteration of the distinguisher in Algorithm 1.

In the real world, we have a relation that is maintained across all queries:

$$V_o \oplus V_e = \mathit{\Delta}.$$

Furthermore, each query defines a difference equation over π_2:

$$\pi_2(U_o \oplus \delta) \oplus \pi_2(U_o) = \mathit{\Delta},$$

where $\delta = U_o \oplus U_e$. By construction, this equation must have at least two solutions. The first is is U_o and the second is U_e. Hence, the query $(T^\star, M)$, where $T^\star = U_e \oplus S_o$ collides with (T, M). However, whether this is the only collision that leads to X, or not, depends on the difference distribution of the permutation π_2. For now, let the set of solutions to the difference equation

$$\pi_2(x \oplus \beta) \oplus \pi_2(x) = \mathit{\Delta}$$

be $\mathcal{S}_{\beta,\mathit{\Delta}}$. Consider an equation $\pi_2(x \oplus \delta) \oplus \pi_2(x) = \mathit{\Delta}$ that has four solutions:

$$\pi_2(U_o \oplus \delta) \oplus \pi_2(U_o) = \mathit{\Delta}$$

$$\pi_2((U_o \oplus \delta) \oplus \delta) \oplus \pi_2(U_o \oplus \delta) = \mathit{\Delta}$$

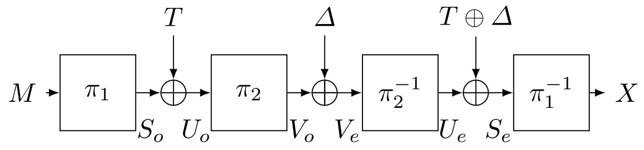

Fig. 5: The internal values of an iteration of the distinguisher in Algorithm 1.

$$\pi_2(U_o \oplus \gamma \oplus \delta) \oplus \pi_2(U_o \oplus \gamma) = \Delta$$
$$\pi_2((U_o \oplus \gamma \oplus \delta) \oplus \delta) \oplus \pi_2(U_o \oplus \gamma \oplus \delta) = \Delta$$

and the four corresponding tweaks

$$S_o \oplus U_o$$
$$S_o \oplus U_o \oplus \delta$$
$$S_o \oplus U_o \oplus \gamma$$
$$S_o \oplus U_o \oplus \gamma \oplus \delta.$$

Then,

$$\begin{aligned} S_e^{(0)} &= (U_o \oplus \delta) \oplus S_o \oplus U_o \oplus \Delta = S_o \oplus \delta \oplus \Delta \\ S_e^{(1)} &= U_o \oplus (\delta \oplus S_o \oplus U_o) \oplus \Delta = S_o \oplus \delta \oplus \Delta \\ S_e^{(2)} &= (U_o \oplus \gamma \oplus \delta) \oplus (S_o \oplus U_o \oplus \gamma) \oplus \Delta = S_o \oplus \delta \oplus \Delta \\ S_e^{(3)} &= (U_o \oplus \gamma) \oplus (\delta \oplus S_o \oplus U_o \oplus \gamma) \oplus \Delta = S_o \oplus \delta \oplus \Delta. \end{aligned}$$

Thus,

$$S_e^{(0)} = S_e^{(1)} = S_e^{(2)} = S_e^{(3)}$$

and they form a multi-collision. The value propagation of this example is visually depicted in Figure 6. This multi-collision gives us an insight on the different types of collisions that can occur. Either the collision consists of two instances where the first instance has $U_o = X, U_e = X \oplus \delta$ and the second instance has $U_e = X, U_o = X \oplus \delta$, *i.e.*, the two values are flipped, or it consists of $U_o = X, U_e = X \oplus \delta$ and $U_o = X \oplus \gamma, U_e = X \oplus \delta \oplus \gamma$ where X and $X \oplus \gamma$ are different solutions to the required difference equation.

If we know the exact values of $|\mathcal{S}_{\beta,\Delta}|$ $\forall \beta \in \{0,1\}^n$, we can calculate the exact number of collisions in the range of X. However, since the permutation is secret, such information is not available. The next best thing is to know for a given Δ, how many equations have 0 solutions, how many equations have 2 solutions,...etc. Let Q_i be the number of values β such that $\pi_2(x \oplus \beta) \oplus \pi_2(x) = \Delta$ has i solutions and m is the maximum number of possible solutions for any such equation. Then, the number of collisions is given by

$$\texttt{coll} = Q_2 + Q_4 * 6 + Q_6 * 15 + Q_8 * 28 + \ldots + Q_m \binom{m}{2}. \tag{5}$$

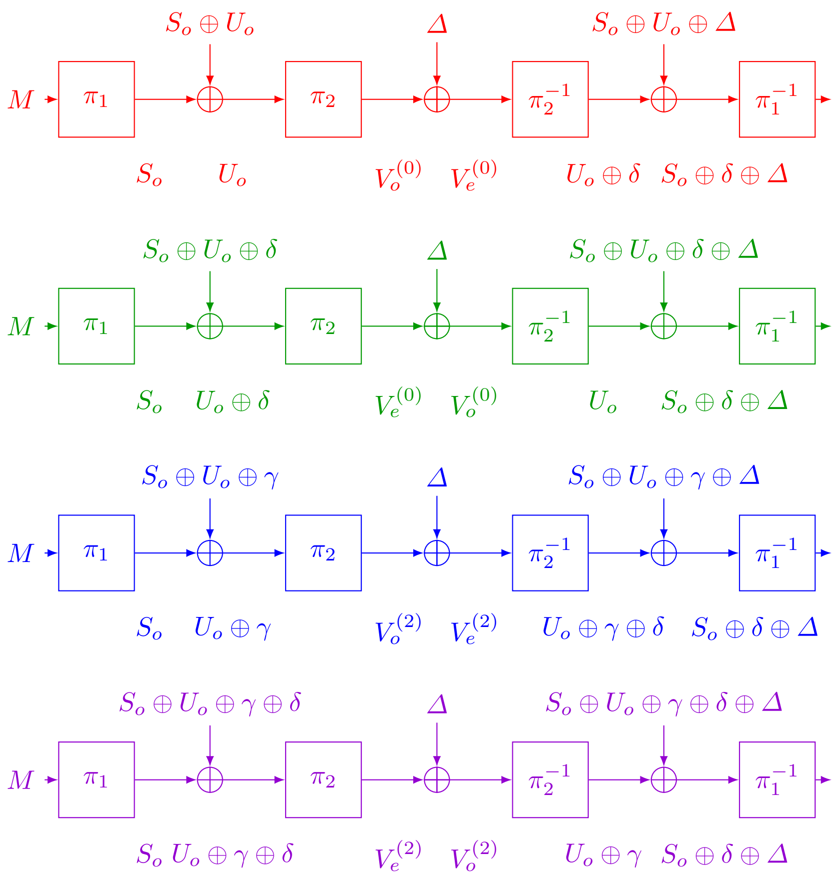

Fig. 6: The propagation in a four-way multi-collision.

If π_2 is an APN, then $\texttt{coll} = 2^{n-1}$. This means that on the average, the APN case has half a collision more than the ideal case. This may not be enough to distinguish between the two cases. However, if π_2 deviates in the slightest from being an APN, *e.g.,* if one of the considered equations has 4 solutions, we get

$$\texttt{coll} = 2^{n-1} - 2 + 6 = 2^{n-1} + 4,$$

which is 4.5 more than the ideal case. As we consider that more equations have more than two solutions, the number of expected collisions increases. The worst case scenario is when π_2 is an affine permutation, in which case, the expected number of collisions is $\binom{2^n}{2}$. However, this case is not relevant for attacks on designs based on block ciphers. We are interested in the expected number of collisions when π_2 is a random permutation. We show below that the expected number of collisions is 2^n, twice that of the ideal world.

On the Statistics of Random Permutations A random permutation over n bits is sampled uniformly from the set of all possible permutations over n bits. We recall that the DDT of a permutation π is a $2^n \times 2^n$ table that for an input difference β and output difference Δ includes the number of solutions of the difference equation:

$$\pi(X \oplus \beta) \oplus \pi(X) = \Delta.$$

O'Connor showed in Eurocrypt 93 [32] that the expected percentage of zeros in such table for a random permutation is 60.65%. If π is an APN, then the percentage of zeros will be slightly higher than 50%. This already shows that the distinguishing advantage is non-negligible, as the relatively high percentage of zeros will be offset by many entries that are larger than 2, since each row and column in the DDT must add up to 2^n. In fact, Daemen and Rijmen [11] showed that the distribution of the entries of the DDT is given by Poisson's distribution. Particularly,

$$\Pr[|S_{\beta,\Delta}| = x] = \frac{0.5^{x/2} e^{-0.5}}{(x/2)!}.$$

Using Bayes' theorem, then for $x > 0$,

$$\begin{gathered}\Pr[|S_{\beta,\Delta}| = x \,|\, |S_{\beta,\Delta}| > 0] = \\ \frac{\Pr[|S_{\beta,\Delta}| = x] \Pr[|S_{\beta,\Delta}| > 0 \,|\, |S_{\beta,\Delta}| = x]}{\Pr[|S_{\beta,\Delta}| > 0]} = \\ \frac{\Pr[|S_{\beta,\Delta}| = x]}{\Pr[|S_{\beta,\Delta}| > 0]} = \\ \frac{0.5^{x/2} e^{-0.5}}{(x/2)!(1 - e^{-0.5})}\end{gathered}$$

These distributions can be used to estimate Equation 5. Let $x = 2i$, then[3]

$$E[\texttt{coll}] = e^{-0.5} \cdot 2^n \sum_{i>0} \frac{0.5^i \binom{2i}{2}}{i!} =$$

[3] $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$ and $e^x = \sum_{i>b} \frac{x^{i-b-1}}{(i-b-1)!}$.

$$e^{-0.5} \cdot 2^n \sum_{i>0} \frac{0.5^i \frac{2i(2i-1)}{2}}{i!} =$$

$$e^{-0.5} \cdot 2^{n+1} \sum_{i>0} \frac{0.5^i i(i-0.5)}{i!} =$$

$$e^{-0.5} \cdot 2^{n+1} \sum_{i>0} \frac{0.5^i i(i-1+0.5)}{i!} =$$

$$e^{-0.5} \cdot 2^{n+1} \sum_{i>0} \left(\frac{0.5^i i(i-1)}{i!} + \frac{0.5^i i \times 0.5}{i!} \right) =$$

$$e^{-0.5} \cdot 2^{n+1} \left(\sum_{i>0} \frac{0.5^i i(i-1)}{i!} + \sum_{i>0} \frac{0.5^i i \times 0.5}{i!} \right) =$$

$$e^{-0.5} \cdot 2^{n+1} \left(\sum_{i>1} \frac{0.5^i}{(i-2)!} + 0.5 \sum_{i>0} \frac{0.5^i}{(i-1)!} \right) =$$

$$e^{-0.5} \cdot 2^{n+1} \left(0.5^2 \sum_{i>1} \frac{0.5^{i-2}}{(i-2)!} + 0.5^2 \sum_{i>0} \frac{0.5^{i-1}}{(i-1)!} \right) =$$

$$e^{-0.5} \cdot 2^{n+1} \left(0.5^2 e^{0.5} + 0.5^2 e^{0.5} \right) = 0.5 \cdot 2^{n+1}.$$

Therefore,

$$E[\texttt{coll}] = 2^n,$$

which means that the distinguisher in Algorithm 1 is expected to have twice as many collisions in the real world as in the ideal world. $\theta(q)$ can be generalized as:

$$\theta(q) = 2^{2d-1} + 2^{2d-2}$$

when $q = 2^{n/2+d}$, which is $\approx 1.5\times$ the expected number of collisions in the ideal case.

To verify that the sampled permutations follow the same distribution, we have implemented a Monte-Carlo experiment to estimate the probability distribution of the number of solutions of a difference equation given that solutions exist by generating many random permutations for $16 \geq n \geq 30$. Almost all the generated permutations satisfied that the percentage of zero entries is around 60.65%. We found that the distribution settles around approximately the distribution in Table 4.

Table 4: The estimated probability distribution of the number of solutions for a difference equation over a random permutation, when it is known that solutions exist.

x	2	4	6	≥ 8
$\Pr(x)$	0.772	0.192	0.032	0.004

The distribution in Table 4 helps us estimate the number of expected solutions and the probability of a collision. Note that while stopped at 8 solutions, including mores solutions only increases the probability of collision. Since the probability of more than 8 solutions seems to be very small, we believe estimation to be a good enough approximation. Assuming the maximum number of solutions is 8, we can estimate Q_i as

$$E[Q_i] = 0.3935 \times \Pr[i] \times 2^n.$$

By substituting in Equation 5, we get

$$E[\texttt{coll}] = 0.3935 \times 2^n(0.772 + 0.192 \times 6 + 0.032 \times 15 + 0.004 \times 28) =$$

$$0.3935 \times 2.516 \times 2^n \approx 2^n.$$

This estimation indicates that when π_2 is a random permutation (or a well-designed block cipher), the expected number of collisions is twice that of the ideal world. Hence, by setting $q = c2^{n/2}$ for a small constant c, and setting the appropriate θ, in Algorithm 1, we get a distinguisher that succeeds with very high probability.

Experimental Verification In order to gain more confidence in the attack, we have implemented two experiments to verify the distinguishing advantage. In the first experiment, we used random permutations generated using Python NumPy's `shuffle` and `argsort` functions, to generate and invert a permutation, respectively. We generated permutations of sizes 16, 20, 24, 28 and 32 bits and performed the distinguishing attack on each generated permutation. Results where taken over an average of $1,000 \sim 10,000$ random generations (each consisting of 3 independent permutations). In the ideal world, random values are sampled, since the uniqueness of the tweak ensures each permutation is sampled at most once. Table 5 includes the average number of collisions for $n = 16$ and $n = 20$, which matches the number of collisions observed for other values of n, as well. The distinguisher reaches 16 expected collisions in the real world $4\times$ faster than the distinguisher in [16] for $n = 16$ and $16\times$ faster for $n = 20$.

Table 5: Average number of collisions using random permutations.

n	16					
$log_2(q)$	6	7	8	9	10	11
real	0.06	0.27	0.96	3.72	15.62	63.59
ideal	0.023	0.12	0.48	1.98	7.91	31.17

n	20					
$log_2(q)$	8	9	10	11	12	13
real	0.073	0.203	1.02	4.01	15.69	63.63
ideal	0.023	0.11	0.47	1.94	7.92	32.57

Table 6 shows the success rate for the different values of n and different parameters q and $\theta(q)$. The distinguisher reaches $\geq 85\%$ with complexity $2^{n/2+3}$ and and 99% success rate with complexity $2^{n/2+4}$, since each iteration includes two queries to the construction. For large n, the factors 2^3 and 2^4 are small. For a visual representation, Figure 7 shows the comparison between the complexity of the distinguisher against the birthday bound and the claim in [3]. The distinguisher breaks the claim with $\geq 85\%$ success rate for $18 < n \leq 24$, and breaks it with $\geq 99\%$ for $n > 24$. With complexity $2^{n/2+5}$, we get a success rate of almost 100%, and an attack that breaks the security claim for In practice, $n \geq 64$.

Table 6: The success rate achieved for different values of n and q.

n	q (85%)	$\theta(q)$ (85%)	Success Rate	q (99%)	$\theta(q)$ (99%)	Success Rate
16	10	12	87.2%	11	48	99%
20	12	12	86.6%	13	48	99%
24	14	12	90%	15	48	99%
28	16	12	85%	17	48	99%
32	18	12	87.5%	19	48	99%

In order to validate our experiments further, and eliminate any issues that may arise from Python's random generation, we ran a second experiment using the implementation of the 16-bit cipher Small-Present-16 [26] provided by the authors of [16]. The number of collisions is taken as an average over 10,000 executions of the attack. The results are presented in Table 7. The results statistically match the random permutation case. A sample of the distribution of the number of solutions for a input difference against all possible output differences and for a given key is given in Table 8. The distribution follows closely the simulated distribution in Table 4, which both validates our simulations and indicates that Small-Present-16 behaves closely to a randomly selected permutation. We have also replicated the success rate experiment and got 90.9% for $q = 2^{10}$ and 99.7% for $q = 2^{11}$.

Table 7: Average number of collisions using Small-Present-16.

n	16					
$log_2(q)$	6	7	8	9	10	11
real	0.058	0.25	0.98	4.02	16	63.94
ideal	0.027	0.12	0.49	1.98	7.98	31.92

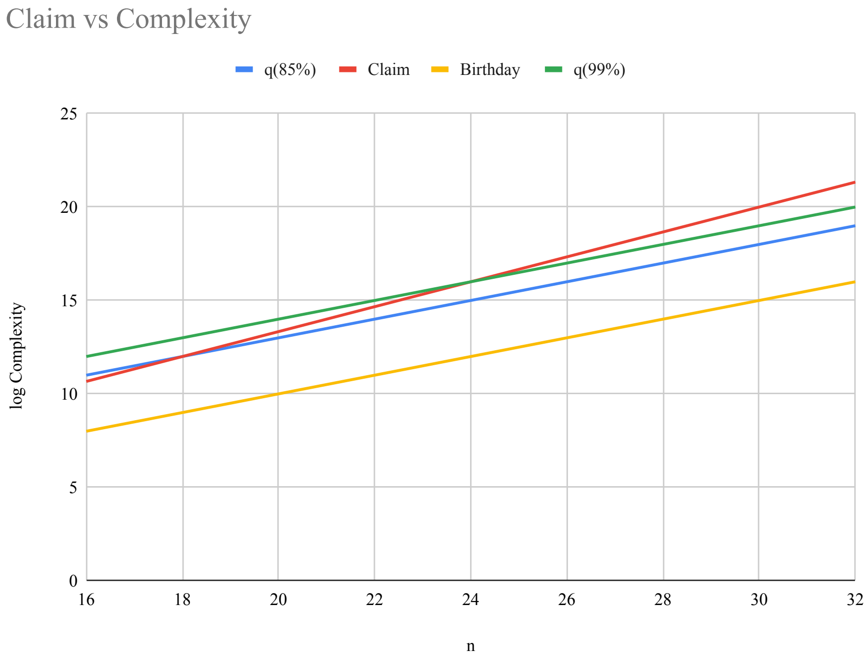

Fig. 7: The complexity of the distinguisher for different success rates compared to the claim of [3] and the birthday bound.

Table 8: A sample of the distribution of the number of solutions for a difference equation defined over Small-Present-16 for a given secret key.

x	2	4	6	≥ 8
$\Pr(x)$	0.773	0.191	0.031	0.005

3.2 Formal Attack Algorithm $\mathcal{A}^*$

Based on the previous observations and experimentatl verifications, we now give a formal attack with rigorous advantage calculations.
Fix a message $m \in \{0,1\}^n$, a subspace $\mathcal{T} \subseteq \{0,1\}^n$ of size q (assuming q is a power of 2), and a $\Delta \notin \mathcal{T}$. We write $\mathcal{T} = \{t_1, \ldots, t_q\}$. Let $\boldsymbol{\pi}_1(m) = \widehat{\mathsf{M}}$ (unknown secret). For all $t_i \in \mathcal{T}$:

1. Make encryption query (t_i, m) and suppose the response is C_i.
2. Make decryption query $(t_i \oplus \Delta, \mathsf{C}_i)$ and suppose the response is X_i.
3. Return 1, if for some $j < i$, $\mathsf{X}_i = \mathsf{X}_j$.

Note that, the time and space complexity of the attack algorithm are both dominated by the query complexity.

3.3 Advantage Calculation

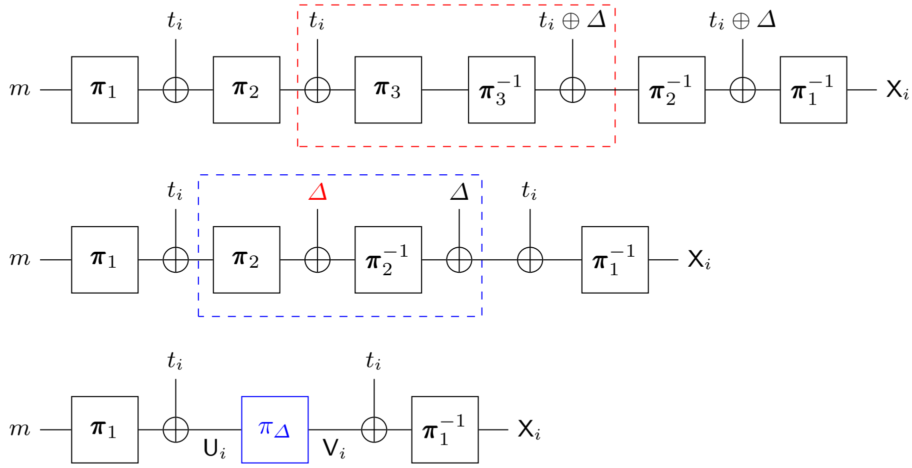

Fig. 8: Cancellation of $\boldsymbol{\pi}_3$.

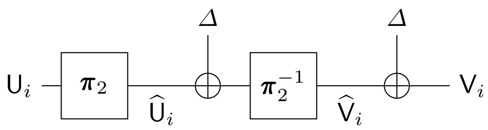

Fig. 9: The π_Δ permutation.

Ideal world collision probability. The ideal world probability of obtaining a collision can be derived as follows

$$\Pr_{\mathsf{Id}}[\exists i, j \in [q] : \mathsf{X}_i = \mathsf{X}_j] = 1 - \mathsf{Pr}\left(\forall i, j \in [q] : \mathsf{X}_i \neq \mathsf{X}_j\right)$$
$$= 1 - \frac{(2^n)_q}{2^{nq}}$$

We denote this ideal probability as $\mathbf{cp}(q) := 1 - \frac{(2^n)_q}{2^{nq}}$ for future use.

Real world collision probability. Note that since the same message m is used in every query by the attacker we have $\mathsf{U}_i \oplus \mathsf{U}_j = t_i \oplus t_j$ for all $i, j \in [q]$. If two responses collide, i.e., $\mathsf{X}_i = \mathsf{X}_j$ then we must have $\mathsf{V}_i \oplus \mathsf{V}_j = t_i \oplus t_j$.

Therefore, there will be a collision in the i-th and j-th responses if and only if $\mathsf{U}_i \oplus \mathsf{V}_i = \mathsf{U}_j \oplus \mathsf{V}_j$.

From Fig. 9 we can observe that $\mathsf{U}_i \oplus \mathsf{V}_i = \mathsf{U}_j \oplus \mathsf{V}_j$, or equivalently $\mathsf{U}_i \oplus \widehat{\mathsf{V}}_i = \mathsf{U}_j \oplus \widehat{\mathsf{V}}_j$, holds if and only if:

- Either $\widehat{\mathsf{U}}_i \oplus \widehat{\mathsf{U}}_j = \mathit{\Delta}$
- Or $(\widehat{\mathsf{U}}_i \oplus \widehat{\mathsf{U}}_j \neq \mathit{\Delta}) \wedge \left(\boldsymbol{\pi}_2^{-1}(\widehat{\mathsf{U}}_i \oplus \mathit{\Delta}) \oplus \mathsf{U}_i = \boldsymbol{\pi}_2^{-1}(\widehat{\mathsf{U}}_j \oplus \mathit{\Delta}) \oplus \mathsf{U}_j\right)$

Let us define the following three events

$$\begin{aligned}
\mathtt{E}_0 &:= (\exists i, j \in [q] : \mathsf{U}_i \oplus \mathsf{V}_i = \mathsf{U}_j \oplus \mathsf{V}_j) \\
\mathtt{E}_1 &:= \left(\exists i, j \in [q] : \widehat{\mathsf{U}}_i \oplus \widehat{\mathsf{U}}_j = \mathit{\Delta}\right) \\
\mathtt{E}_2 &:= \left(\exists i, j \in [q] : \boldsymbol{\pi}_2^{-1}(\widehat{\mathsf{U}}_i \oplus \mathit{\Delta}) \oplus \mathsf{U}_i = \boldsymbol{\pi}_2^{-1}(\widehat{\mathsf{U}}_j \oplus \mathit{\Delta}) \oplus \mathsf{U}_j\right)
\end{aligned}$$

The above observation says that $\mathtt{E}_0 \Leftrightarrow \mathtt{E}_1 \cup \mathtt{E}_2$. Then we can write

$$\mathsf{Pr}\left(\mathtt{E}_0\right) = \mathsf{Pr}\left(\mathtt{E}_1\right) + \mathsf{Pr}\left(\mathtt{E}_1^c \wedge \mathtt{E}_2\right) \tag{6}$$

Calculating $\mathsf{Pr}\left(\mathtt{E}_1^c\right)$*.* Assuming the underlying permutation $\boldsymbol{\pi}_2$ as a random permutation, $\mathsf{Pr}\left(\mathtt{E}_1^c\right)$ is same as the probability that $\widehat{\mathsf{U}}_i \oplus \widehat{\mathsf{U}}_j \neq \mathit{\Delta}$, $\forall i, j \in [q]$, where $\widehat{\mathsf{U}}_1, \ldots, \widehat{\mathsf{U}}_q \xleftarrow{\text{wor}} \{0,1\}^n$. Suppose, for some $t < q$, $\widehat{\mathsf{U}}_1, \ldots, \widehat{\mathsf{U}}_t$ is chosen such that

$$(\widehat{\mathsf{U}}_i \oplus \widehat{\mathsf{U}}_j \neq \mathit{\Delta}) \wedge (\widehat{\mathsf{U}}_i \neq \widehat{\mathsf{U}}_j),\ \forall i, j \in [t] \tag{7}$$

Then the possible choices for $\widehat{\mathsf{U}}_{t+1}$ are exactly $\{0,1\}^n \setminus \mathcal{S}_t$, where

$$\mathcal{S}_t := \{\widehat{\mathsf{U}}_1, \ldots, \widehat{\mathsf{U}}_t\} \cup \{\widehat{\mathsf{U}}_1 \oplus \mathit{\Delta}, \ldots, \widehat{\mathsf{U}}_t \oplus \mathit{\Delta}\}.$$

Since $\widehat{\mathsf{U}}_1, \ldots \widehat{\mathsf{U}}_t$ satisfies condition (7) we have $|\mathcal{S}_t| = 2t$. Thus total number of ways of selecting $\widehat{\mathsf{U}}_1, \ldots, \widehat{\mathsf{U}}_q$ is $2^n(2^n - 2) \cdots (2^n - 2q + 2)$. Hence we have,

$$\begin{aligned}
\mathsf{Pr}\left(\mathtt{E}_1^c\right) &= \frac{2^n(2^n-2)\cdots(2^n-2q+2)}{(2^n)_q} \\
&\leq \frac{(2^n)_q}{2^{nq}} = (1 - \mathbf{cp}(q))
\end{aligned} \tag{8}$$

Hence, $\mathsf{Pr}\left(\mathtt{E}_1\right) \geq \mathbf{cp}(q)$. So, we get that the probability of collision of responses in the real world is bounded as follows

$$\begin{aligned}
\mathsf{Pr}_{\mathsf{Re}}\left(\exists i, j \in [q] : \mathsf{X}_i = \mathsf{X}_j\right) &= \mathsf{Pr}\left(\mathtt{E}_0\right) \\
&= \mathsf{Pr}\left(\mathtt{E}_1\right) + \mathsf{Pr}\left(\mathtt{E}_1^c \wedge \mathtt{E}_2\right) \\
&\geq \mathbf{cp}(q) + \mathsf{Pr}\left(\mathtt{E}_1^c \wedge \mathtt{E}_2\right) \\
&= \mathsf{Pr}_{\mathsf{Id}}\left(\exists i, j \in [q] : \mathsf{X}_i = \mathsf{X}_j\right) + \mathsf{Pr}\left(\mathtt{E}_1^c \wedge \mathtt{E}_2\right)
\end{aligned}$$

Hence the advantage of our distinguisher $\mathcal{A}^*$ will be

$$\mathbf{Adv}_{\mathsf{TNT}}^{\mathrm{tsprp}}(\mathcal{A}^*) \geq \mathsf{Pr}\left(\mathtt{E}_1^c \wedge \mathtt{E}_2\right) \tag{9}$$

So, it is sufficient to provide a lower bound for $\mathsf{Pr}\left(\mathtt{E}_1^c \wedge \mathtt{E}_2\right)$ which is the same as $\mathsf{Pr}\left(\mathtt{E}_2 \mid \mathtt{E}_1^c\right) \times \mathsf{Pr}\left(\mathtt{E}_1^c\right)$.

$$\begin{aligned}
\mathsf{Pr}\left(\mathtt{E}_1^c\right) &= \frac{2^n(2^n-2)\cdots(2^n-2q+2)}{(2^n)_q} \\
&= (1-\mathbf{cp}(q))\prod_{i=1}^{q-1}\left(1-\frac{i^2}{(2^n-i)^2}\right) \\
&\geq (1-\mathbf{cp}(q))\prod_{i=1}^{q-1}\left(1-\frac{i^2}{(2^n-q)^2}\right) \\
&\geq (1-\mathbf{cp}(q))\big(1-\frac{q(q-1)(2q-1)}{6(2^n-q)^2}\big) \\
&\geq (1-\mathbf{cp}(q))(1-\frac{2q^3}{2^{2n}})
\end{aligned} \tag{10}$$

In the last inequality, we assume that $q \leq 2^{n-1}$ as we eventually use $q = O(2^{n/2})$.

Calculating $\mathsf{Pr}\left(\mathtt{E}_2 | \mathtt{E}_1^c\right)$. Given the condition that $\widehat{\mathsf{U}}_i \oplus \mathit{\Delta} \neq \widehat{\mathsf{U}}_j$, $\forall i,j \in [q]$, we have that $\boldsymbol{\pi}_2^{-1}(\widehat{\mathsf{U}}_i \oplus \mathit{\Delta}) \notin \mathcal{U} := \{\mathsf{U}_1, \ldots, \mathsf{U}_q\}$. Note that the set $\mathcal{U} = \mathcal{T} \oplus \boldsymbol{\pi}_1(m)$ is the affine space obtained from the subspace $\mathcal{T}$ by translating it by $\boldsymbol{\pi}_1(m)$. Now, declaring the variables $\widehat{\mathsf{V}}_i := \boldsymbol{\pi}_2^{-1}(\widehat{\mathsf{U}}_i \oplus \mathit{\Delta})$ and noting that $\mathsf{U}_i \oplus \mathsf{U}_j = t_i \oplus t_j$, we have that $(\mathtt{E}_2|\mathtt{E}_1^c)$ is same as the event,

$$\bigvee_{i \neq j \in [q]} \left(\widehat{\mathsf{V}}_i \oplus \widehat{\mathsf{V}}_j = t_i \oplus t_j\right), \text{ where } \widehat{\mathsf{V}}_1, \ldots, \widehat{\mathsf{V}}_q \xleftarrow{\text{wor}} \mathcal{U}^c := \{0,1\}^n \setminus \mathcal{U}.$$

For every $i \neq j \in [q]$, we define the events $\mathtt{E}_{\{i,j\}} = (\widehat{\mathsf{V}}_i \oplus \widehat{\mathsf{V}}_j = t_i \oplus t_j)$ where $\widehat{\mathsf{V}}_1, \ldots, \widehat{\mathsf{V}}_q \xleftarrow{\text{wor}} \mathcal{U}^c$. Note that for any distinct i, j,

$$\widehat{\mathsf{V}}_i, \widehat{\mathsf{V}}_j \xleftarrow{\text{wor}} \mathcal{U}^c.$$

In general, any subset follows WOR distribution. Using this observation we have $\mathsf{Pr}\left(\mathtt{E}_{\{i,j\}}\right) = (2^n - q - 1)^{-1}$. This is true because any choice of $\widehat{\mathsf{V}}_i$ from the set $\mathcal{U}^c$, we have $\widehat{\mathsf{V}}_i \oplus t_i \oplus t_j \notin \mathcal{U}$. By using a similar argument, one can show that

$$\mathsf{Pr}\left(\mathtt{E}_{\{i,j\}} \wedge \mathtt{E}_{\{k,\ell\}}\right) \leq \frac{1}{(2^n-q-1)(2^n-q-3)}.$$

Hence, by using Bonferroni's inequality and denoting $\alpha(q) := \frac{\binom{q}{2}}{2^n-q-1}$ we have

$$\begin{aligned}
\mathsf{Pr}\left(\bigvee_{i\neq j\in[q]} \mathtt{E}_{\{i,j\}}\right) &\geq \frac{\binom{q}{2}}{2^n-q-1} - \frac{\binom{q}{2}^2}{2(2^n-q-1)(2^n-q-3)} \\
&= \alpha(q)\big(1 - \frac{\binom{q}{2}}{2(2^n-q-3)}\big) \\
&\geq \alpha(q)\big(1 - \frac{\alpha(q)}{2}(1 + \frac{2}{(2^n-q-3)})\big) \qquad (11)
\end{aligned}$$

Note that $\mathbf{cp}(q) \leq \alpha(q)$ (by union bound). Thus, using (9)-(11), we have the following result on the TSPRP advantage of $\mathcal{A}^*$.

Theorem 2. *For $q \leq 2^{n-1}$, and $\alpha(q) = \frac{q(q-1)}{2^n-q-1}$, we have*

$$\mathbf{Adv}^{\mathrm{tsprp}}_{\mathsf{TNT}}(\mathcal{A}^*) \geq \alpha(q)(1-\alpha(q))(1 - \frac{\alpha(q)}{2} - \frac{\alpha(q)}{2^n-q-3})(1 - \frac{2q^3}{2^{2n}}).$$

Specifically, suppose q_0 be the value such that $\alpha(q_0) = 1/2$. Clearly, $q_0 = O(2^{n/2})$. So, for $q = \lceil q_0 \rceil$, we have

$$\mathbf{Adv}^{\mathrm{tsprp}}_{\mathsf{TNT}}(\mathcal{A}^*) \geq \frac{1}{8} - \lambda(n),$$

where $\lambda(n) = O(\frac{q_0^3}{N^2}) = O(2^{-n/2})$ is negligible function of n.

4 Birthday-bound Security of TNT and Its Single Key Variant

One can rely on the TSPRP bound by Zhang et al. to demonstrate the tightness of the proposed attacks. However, we observe that the generic bound in [40] introduces some constant factors, and in general, an independent security proof, using a different proof technique, will instill greater confidence in the revised security claims of TNT.

In light of the above discussion, it is clear that the security of TNT is in a limbo. Here, we salvage a birthday-bound security for TNT based on three independent random permutations $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$, and $\boldsymbol{\pi}_3$ of $\{0,1\}^n$.

Theorem 3. *For all $q \geq 1$, we have*

$$\mathbf{Adv}^{\mathrm{tsprp}}_{\mathsf{TNT}}(q) \leq \frac{q^2}{2^n}.$$

Proof. The statement is vacuously true for $q \geq 2^{n/2}$. We will use the H-coefficient technique (see Corollary 1) to prove the statement for $1 \leq q < 2^{n/2}$.

Let $\mathcal{O}_0$ and $\mathcal{O}_1$ be the oracles corresponding to TNT and a tweakable random permutation $\widetilde{\boldsymbol{\pi}}$, respectively. If $(\mathsf{T}_i, \mathsf{M}_i)$ is the encryption query with a tweak T_i we write the response as C_i. Similarly, if $(\mathsf{T}_i, \mathsf{C}_i)$ is the decryption query with a tweak T_i we write the response as M_i. After all queries have been made, the two oracles release some additional data to the adversary, who is obviously free to ignore this additional information, X^q and Y^q.

In the real world, X^q and Y^q correspond to the output of $\boldsymbol{\pi}_1$ and input of $\boldsymbol{\pi}_3$, respectively, and thus they are well defined from the definition of TNT. The real world transcript is thus defined as the tuple

$$\Theta_1 := (\mathsf{T}^q, \mathsf{M}^q, \mathsf{C}^q, \mathsf{X}^q, \mathsf{Y}^q).$$

In the ideal system $\widetilde{\boldsymbol{\pi}}$, we sample $\mathsf{X}^q, \mathsf{Y}^q$ as follows for all $i \in [q]$:

1. $\mathsf{X}_i = \mathsf{X}_j$ whenever $\mathsf{M}_i = \mathsf{M}_j$ for $j < i$. Otherwise (for all $j < i$, $\mathsf{M}_j \neq \mathsf{M}_i$), we sample
$$\mathsf{X}_i \leftarrow\!\!\$\ \{0,1\}^n \setminus \{x \in \{0,1\}^n : \exists j < i, \mathsf{X}_j = x\}.$$
2. $\mathsf{Y}_i = \mathsf{Y}_j$ whenever $\mathsf{C}_j = \mathsf{C}_i$ for $j < i$. Otherwise (for all $j < i$, $\mathsf{C}_j \neq \mathsf{C}_i$), we sample
$$\mathsf{Y}_i \leftarrow\!\!\$\ \{0,1\}^n \setminus \{y \in \{0,1\}^n : \exists j < i, \mathsf{Y}_j = y\}.$$

The ideal world transcript is defined as

$$\Theta_0 := (\mathsf{T}^q, \mathsf{M}^q, \mathsf{C}^q, \mathsf{X}^q, \mathsf{Y}^q).$$

Note that, we use the same notation to denote the random variables in both the worlds. However, their probability distributions will be unambiguously determined at the time of probability computations.

Bad Transcript and Its Analysis. A transcript $(t^q, m^q, c^q, x^q, y^q)$ is called *bad* if and only if

- there is a collision among u^q values where $u_i = x_i + t_i$; or
- there is a collision among v^q values where $v_i = y_i + t_i$.

Let Ω_{bad} denote the set of all bad transcripts. Now, $\Theta_0 \in \Omega_{\mathsf{bad}}$ if either for some $i < j$, $\mathsf{X}_i + \mathsf{T}_i = \mathsf{X}_j + \mathsf{T}_j$ or $\mathsf{Y}_i + \mathsf{T}_i = \mathsf{Y}_j + \mathsf{T}_j$. It is easy to see that for any fixed $i < j$, $\mathsf{Pr}\left(\mathsf{X}_i + \mathsf{T}_i = \mathsf{X}_j + \mathsf{T}_j\right) \leq (2^n - 1)^{-1}$ and similarly for the other case. So, by using the union bound,

$$\mathsf{Pr}\left(\in \Omega_{\mathsf{bad}}\right) \leq \frac{q(q-1)}{2^n - 1} \leq \frac{q^2}{2^n}.$$

Analysis of Good Transcripts. For a good transcript $\tau = (t^q, m^q, c^q, x^q, y^q)$, we know that (m^q, x^q), (y^q, c^q), and (u^q, v^q) are permutation consistent and hence for the real world we have

$$\begin{aligned}\mathsf{Pr}\left(\Theta_1 = \omega\right) &= \mathsf{Pr}\left(\boldsymbol{\pi}_1(m^q) = x^q\right) \times \mathsf{Pr}\left(\boldsymbol{\pi}_2(u^q) = v^q\right) \times \mathsf{Pr}\left(\boldsymbol{\pi}_3(y^q) = c^q\right)\\ &= \frac{1}{(2^n)_r} \times \frac{1}{(2^n)_q} \times \frac{1}{(2^n)_s}\end{aligned}$$

where r and s denote the the number of distinct values present in m^q and c^q respectively. In the ideal world, we have,

$$\begin{aligned}\mathsf{Pr}\left(\Theta_0 = \omega\right) &= \mathsf{Pr}\left(\widetilde{\boldsymbol{\pi}}(t^q, m^q) = c^q\right) \times \frac{1}{(2^n)_r} \times \frac{1}{(2^n)_s} \\ &\leq \frac{1}{(2^n)_q} \times \frac{1}{(2^n)_r} \times \frac{1}{(2^n)_s},\end{aligned}$$

where the final inequality follows from the fact that $\mathsf{Pr}\left(\widetilde{\boldsymbol{\pi}}(t^q, m^q) = c^q\right)$ maximizes when $t_i = t_j$ for all $1 \leq i < j \leq q$. The result follows from the H-coefficients technique.

4.1 Birthday-bound Security of single key-variant of TNT

Now we show that even the single key-variant of TNT, which we denote as $\mathsf{1k\text{-}TNT}$, is sufficient to achieve birthday-bound security. Here we replace the three underlying blockciphers by the same random permutations $\boldsymbol{\pi}$ of $\{0,1\}^n$.

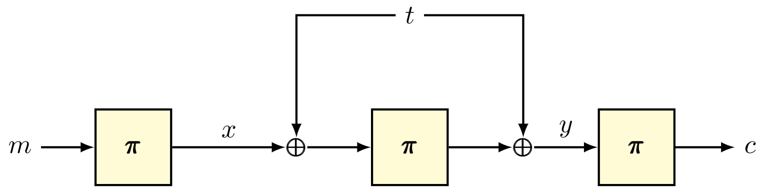

Fig. 10: The single-keyed TNT construction.

Theorem 4. *For all $q \geq 1$, we have*

$$\mathbf{Adv}^{\mathrm{tsprp}}_{\mathsf{1k\text{-}TNT}}(q) \leq \frac{13q^2}{2^n}.$$

Proof. The statement is vacuously true for $q \geq 2^{n/2}$. We will use the H-coefficient technique (see Corollary 1) to prove the statement for $1 \leq q < 2^{n/2}$.

Let $\mathcal{O}_0$ and $\mathcal{O}_1$ be the oracles corresponding to $\mathsf{1k\text{-}TNT}$ and a tweakable random permutation $\widetilde{\boldsymbol{\pi}}$, respectively. If $(\mathsf{T}_i, \mathsf{M}_i)$ is the encryption query with a tweak T_i we write the response as C_i. Similarly, if $(\mathsf{T}_i, \mathsf{C}_i)$ is the decryption query with a tweak T_i we write the response as M_i. After all queries have been made, the two oracles release some additional data to the adversary, who is obviously free to ignore this additional information, X^q and Y^q.

In the real world, X^q and Y^q correspond to the output of first permutation and input of the third permutation, respectively, and thus they are well defined from the definition of $\mathsf{1k\text{-}TNT}$. The real world transcript is thus defined as the tuple

$$\Theta_1 := (\mathsf{T}^q, \mathsf{M}^q, \mathsf{C}^q, \mathsf{X}^q, \mathsf{Y}^q).$$

In the ideal system $\widetilde{\boldsymbol{\pi}}$, we sample $\mathsf{X}^q, \mathsf{Y}^q$ as follows: For every $i \in [q]$,

1. $\mathsf{X}_i = \mathsf{X}_j$ whenever $\mathsf{M}_i = \mathsf{M}_j$ for $j < i$. Otherwise (for all $j < i$, $\mathsf{M}_j \neq \mathsf{M}_i$), we sample
$$\mathsf{X}_i \leftarrow\!\!\$\ \{0,1\}^n \setminus \{x \in \{0,1\}^n : \exists j < i, \mathsf{X}_j = x\}.$$
2. $\mathsf{Y}_i = \mathsf{Y}_j$ whenever $\mathsf{C}_j = \mathsf{C}_i$ for $j < i$. Otherwise (for all $j < i$, $\mathsf{C}_j \neq \mathsf{C}_i$), we sample
$$\mathsf{Y}_i \leftarrow\!\!\$\ \{0,1\}^n \setminus \{y \in \{0,1\}^n : \exists j < i, \mathsf{Y}_j = y\}.$$

The ideal world transcript is defined as
$$\Theta_0 := (\mathsf{T}^q, \mathsf{M}^q, \mathsf{C}^q, \mathsf{X}^q, \mathsf{Y}^q).$$

Note that, we use the same notation to denote the random variables in both the worlds. However, their probability distributions will be unambiguously determined at the time of probability computations.

Bad Transcript and Its Analysis. A transcript $(t^q, m^q, c^q, x^q, y^q)$ is called *bad* if and only if any of the following bad events occur:

bad_{1a}: there is a collision between x^q and c^q values.
bad_{1b}: there is a collision between y^q and m^q values.
bad_{2a}: there is a collision among u^q values where $u_i = x_i + t_i$;
bad_{2b}: there is a collision among v^q values where $v_i = y_i + t_i$.
bad_{3a}: there is a collision between u^q and m^q values.
bad_{3b}: there is a collision between v^q and c^q values.
bad_{4a}: there is a collision between u^q and y^q values.
bad_{4b}: there is a collision between v^q and x^q values.

Let Ω_{bad} denote the set of all bad transcripts.

- $\mathsf{Pr}\left(\mathsf{bad}_{1a}^c\right) = \mathsf{Pr}\left(\mathsf{bad}_{1b}^c\right) \geq (2^n - q)_q/(2^n)_q \geq 1 - 2q^2/2^n$, this is because for bad_{1a}^c to hold, x_i has to be chosen from $\{0,1\}^n \setminus m^q$ and it has to be distinct from $x_1, \ldots, x_{i-1}$.
- $\mathsf{Pr}\left(\mathsf{bad}_{2a}\right) \leq \sum_{i<j} \mathsf{Pr}\left(\mathsf{X}_i + \mathsf{T}_i = \mathsf{X}_j + \mathsf{T}_j\right) \leq \binom{q}{2}/(2^n - 1) \leq q^2/2^{n+1}$. The same bound holds for $\mathsf{Pr}\left(\mathsf{bad}_{2b}\right)$.
- $\mathsf{Pr}\left(\mathsf{bad}_{2a}^c\right) = \mathsf{Pr}\left(\mathsf{bad}_{2b}^c\right) \geq (2^n - q)_q/(2^n)_q \geq 1 - 2q^2/2^n$, this is because for bad_{2a}^c to hold, x_i has to be chosen from $\{0,1\}^n \setminus (m^q + t_i)$ and it has to be distinct from $x_1, \ldots, x_{i-1}$.
- Given the y^q values the the probability of bad_{4a}^c can be bounded in the same way as bad_{3a}^c. Similarly, given the x^q values the the probability of bad_{4b}^c can be bounded in the same way as bad_{3b}^c. Hence $\mathsf{Pr}\left(\mathsf{bad}_{4a}\right) = \mathsf{Pr}\left(\mathsf{bad}_{4b}\right) \leq 2q^2/2^n$.

Thus, we have
$$\mathsf{Pr}\left(\Theta_0 \in \Omega_{\mathsf{bad}}\right) \leq \frac{13q^2}{2^n}.$$

Analysis of Good Transcripts. For a good transcript $\tau = (t^q, m^q, c^q, x^q, y^q)$, we know that (m^q, x^q), (y^q, c^q), and (u^q, v^q) are permutation consistent non-overlapping input-output pairs and hence for the real world we have

$$\begin{aligned}\mathsf{Pr}\left(\Theta_1 = \omega\right) &= \mathsf{Pr}\left(\boldsymbol{\pi}(m^q) = x^q\right) \times \mathsf{Pr}\left(\boldsymbol{\pi}(u^q) = v^q\right) \times \mathsf{Pr}\left(\boldsymbol{\pi}(y^q) = c^q\right) \\ &= \frac{1}{(2^n)_{r+q+s}}\end{aligned}$$

where r and s denote the the number of distinct values present in m^q and c^q respectively. In the ideal world, we have,

$$\begin{aligned}\mathsf{Pr}\left(\Theta_0 = \omega\right) &= \mathsf{Pr}\left(\widetilde{\boldsymbol{\pi}}(t^q, m^q) = c^q\right) \times \frac{1}{(2^n)_r} \times \frac{1}{(2^n)_s} \\ &\leq \frac{1}{(2^n)_q} \times \frac{1}{(2^n)_r} \times \frac{1}{(2^n)_s},\end{aligned}$$

where the final inequality follows from the fact that $\mathsf{Pr}\left(\widetilde{\boldsymbol{\pi}}(t^q, m^q) = c^q\right)$ maximizes when $t_i = t_j$ for all $1 \leq i < j \leq q$. Thus

$$\frac{\mathsf{Pr}\left(\Theta_1 = \omega\right)}{\mathsf{Pr}\left(\Theta_0 = \omega\right)} \geq \frac{(2^n)_q \times (2^n)_r \times (2^n)_s}{(2^n)_{q+r+s}} \geq 1$$

Now the result follows from the H-coefficients technique.

5 The Generalized LRW Paradigm

Throughout, we fix two positive integers τ and n to denote the tweak and block size in bits.

Let $\widetilde{\mathcal{H}}$ be a (τ, n)-tweakable permutation family, and $\mathcal{H}$ be a (τ, n)-hash function family. Let $\widehat{\mathcal{H}} = (\widetilde{\mathcal{H}}^2 \times \mathcal{H})$, $(\widetilde{\mathsf{H}}_1, \widetilde{\mathsf{H}}_2, \mathsf{H}) \leftarrow \mathsf{KG}\left(\widehat{\mathcal{H}}\right)$, and $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \leftarrow\!\$\ \mathsf{Perm}(n)$, where $\mathsf{KG}\left(\widehat{\mathcal{H}}\right)$ is an efficient probabilistic algorithm that returns a random triple from $\widehat{\mathcal{H}}$.

The LRW+ construction is a (τ, n)- tweakable permutation family, defined by the following mapping (see Figure 11 for an illustration):

$$(t, m) \mapsto \widetilde{\mathsf{H}}_2^{-1}\left(t, \boldsymbol{\pi}_2\left(\mathsf{H}(t) \oplus \boldsymbol{\pi}_1\left(\widetilde{\mathsf{H}}_1\left(t, m\right)\right)\right)\right). \tag{12}$$

5.1 Security of LRW+

We say that $\mathsf{KG}\left(\widehat{\mathcal{H}}\right)$ is a pairwise independent sampling mechanism or PISM, if $(\widetilde{\mathsf{H}}_1, \widetilde{\mathsf{H}}_2, \mathsf{H}) \leftarrow \mathsf{KG}\left(\widehat{\mathcal{H}}\right)$ is a pairwise independent tuple.

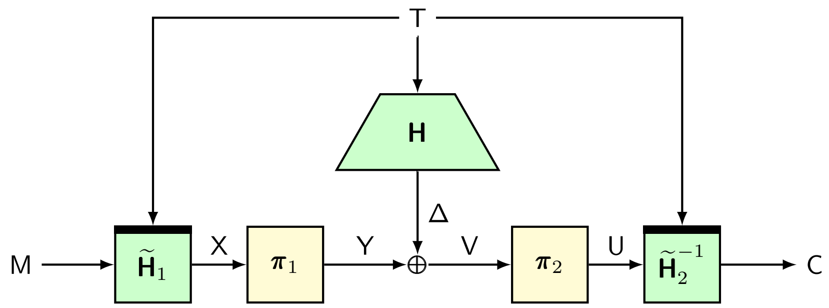

Fig. 11: The LRW+ construction.

We say that $\widetilde{\mathcal{H}}$ is an ϵ-almost universal tweakable permutation family (AUTPF) if and only if for all distinct $(t, m), (t', m') \in \{0,1\}^\tau \times \{0,1\}^n$,

$$\mathsf{Pr}\left(\widetilde{\mathsf{H}} \leftarrow\!\!\$\; \widetilde{\mathcal{H}} \ :\ \widetilde{\mathsf{H}}(t, m) = \widetilde{\mathsf{H}}(t', m')\right) \leq \epsilon.$$

Theorem 5. *Let $\tau, n \in \mathbb{N}$, and $\epsilon_1, \epsilon_2 \in [0,1]$. If $\widetilde{\mathcal{H}}$ and $\mathcal{H}$ are respectively ϵ_1-AUTPF and ϵ_2-AUHF, and $\mathsf{KG}\left(\widehat{\mathcal{H}}\right)$ is a PISM, then, for $q \leq 2^{n-2}$, we have*

$$\mathbf{Adv}^{\mathsf{tsprp}}_{\mathsf{LRW+}}(q) \leq \epsilon(q, n),$$

where

$$\epsilon(q, n) = 2q^2\epsilon_1^{1.5} + \frac{9q^4\epsilon_1^2}{2^n} + \frac{32q^4\epsilon_1}{2^{2n}} + \frac{13q^4}{2^{3n}} + q^2\epsilon_1^2 + q^2\epsilon_1\epsilon_2 + \frac{2q^2}{2^{2n}}. \tag{13}$$

The proof is simply a generalization of Jha and Nandi's proof [22] for 2-LRW2. In particular, we use the expectation method with JN's adaptation of mirror theory [34,9] in the tweakable permutation settings. The complete proof is given in the remainder of this section.

Note that, we are in the information-theoretic setting. In other words, we consider computationally unbounded distinguisher $\mathscr{A}$. Without loss of generality, we assume that $\mathscr{A}$ is deterministic and non-trivial.

5.2 Oracle Description

The two oracles of interest are: $\mathcal{O}_1$, the real oracle, that implements LRW+; and, $\mathcal{O}_0$, the ideal oracle, that implements $\widetilde{\boldsymbol{\pi}} \leftarrow\!\!\$\; \widetilde{\mathsf{Perm}}(\tau, n)$. We consider an extended version of these oracles, the one in which they release some additional information. We use notations analogously as given in Figure 11 to describe the transcript generated by $\mathscr{A}$'s interaction with its oracle.

Description of the real oracle, $\mathcal{O}_1$: The real oracle $\mathcal{O}_1$ faithfully runs *glrw*. We denote the transcript random variable generated by $\mathscr{A}$'s interaction with $\mathcal{O}_1$ by the usual notation Θ_1, which is an 11-ary q-tuple

$$(\mathsf{T}^q, \mathsf{M}^q, \mathsf{C}^q, \mathsf{X}^q, \mathsf{Y}^q, \mathsf{V}^q, \mathsf{U}^q, \Delta^q, \widetilde{\mathbf{H}}_1, \widetilde{\mathbf{H}}_2, \mathbf{H}),$$

defined as follows: The initial transcript consists of $(\mathsf{T}^q, \mathsf{M}^q, \mathsf{C}^q)$, where for all $i \in [q]$:

$$\mathsf{T}_i: \ i\text{-th tweak value} \quad \mathsf{M}_i: \ i\text{-th plaintext value} \quad \mathsf{C}_i: \ i\text{-th ciphertext value},$$

where, $\mathsf{C}^q = \mathsf{LRW}{+}(\mathsf{T}^q, \mathsf{M}^q)$. At the end of the query-response phase $\mathcal{O}_1$ releases some additional information $(\mathsf{X}^q, \mathsf{Y}^q, \mathsf{V}^q, \mathsf{U}^q, \Delta^q, \widetilde{\mathbf{H}}_1, \widetilde{\mathbf{H}}_2, \mathbf{H})$, such that for all $i \in [q]$:

- $(\mathsf{X}_i, \mathsf{Y}_i)$: i-th input-output pair for $\boldsymbol{\pi}_1$,
- $(\mathsf{V}_i, \mathsf{U}_i)$: i-th input-output pair for $\boldsymbol{\pi}_2$,
- Δ_i: i-th internal masking, $\widetilde{\mathbf{H}}_1, \widetilde{\mathbf{H}}_2, \mathbf{H}$: are the hash keys.

Note that X^q, U^q, and Δ^q are completely determined by the hash keys $\widetilde{\mathbf{H}}_1, \widetilde{\mathbf{H}}_2, \mathbf{H}$, and the initial transcript $(\mathsf{T}^q, \mathsf{M}^q, \mathsf{C}^q)$. We include them anyhow for the sake of convenience.

Description of the ideal oracle, $\mathcal{O}_0$: The ideal oracle $\mathcal{O}_0$ has access to $\widetilde{\boldsymbol{\pi}}$. Since $\mathcal{O}_1$ releases some additional information, $\mathcal{O}_0$ must generate these values as well. The ideal transcript random variable Θ_0 is also an 11-ary q-tuple

$$(\mathsf{T}^q, \mathsf{M}^q, \mathsf{C}^q, \mathsf{X}^q, \mathsf{Y}^q, \mathsf{V}^q, \mathsf{U}^q, \Delta^q, \widetilde{\mathbf{H}}_1, \widetilde{\mathbf{H}}_2, \mathbf{H}),$$

defined below. The initial transcript consists of $(\mathsf{T}^q, \mathsf{M}^q, \mathsf{C}^q)$, where for all $i \in [q]$:

$$\mathsf{T}_i: \ i\text{-th tweak value} \quad \mathsf{M}_i: \ i\text{-th plaintext value} \quad \mathsf{C}_i: \ i\text{-th ciphertext value},$$

where $\mathsf{C}^q = \widetilde{\boldsymbol{\pi}}(\mathsf{T}^q, \mathsf{M}^q)$. Once the query-response phase is over $\mathcal{O}_0$ first samples $(\widetilde{\mathbf{H}}_1, \widetilde{\mathbf{H}}_2, \mathbf{H}) \leftarrow\!\!\$\ \mathsf{KG}\left(\widehat{\mathcal{H}}\right)$, and then computes $(\mathsf{X}^q, \mathsf{U}^q, \Delta^q)$, as follows:

$$\mathsf{X}^q := \widetilde{\mathbf{H}}_1(\mathsf{T}^q, \mathsf{M}^q) \qquad \mathsf{U}^q := \widetilde{\mathbf{H}}_2(\mathsf{T}^q, \mathsf{C}^q) \qquad \Delta^q := \mathbf{H}(\mathsf{T}^q).$$

Note that, the conditional distributions of $(\mathsf{X}^q, \mathsf{U}^q, \Delta^q, \widetilde{\mathbf{H}}_1, \widetilde{\mathbf{H}}_2, \mathbf{H})$, given $(\mathsf{T}^q, \mathsf{M}^q, \mathsf{C}^q)$ is identical in both the worlds. This means that X^q, U^q, and Δ^q are defined honestly.

Given the partial transcript $\Theta_0' := (\mathsf{T}^q, \mathsf{M}^q, \mathsf{C}^q, \mathsf{X}^q, \mathsf{U}^q, \Delta^q, \widetilde{\mathbf{H}}_1, \widetilde{\mathbf{H}}_2, \mathbf{H})$ we wish to characterize the hash key $\widehat{\mathbf{H}} := (\widetilde{\mathbf{H}}_1, \widetilde{\mathbf{H}}_2, \mathbf{H})$ as good or bad. We write $\widehat{\mathcal{H}}_{\mathsf{bad}}$ for the set of bad hash keys, and $\widehat{\mathcal{H}}_{\mathsf{good}} := \widehat{\mathcal{H}} \setminus \widehat{\mathcal{H}}_{\mathsf{bad}}$. We say that the hash key $\widehat{\mathbf{H}} \in \widehat{\mathcal{H}}_{\mathsf{bad}}$ (or $\widehat{\mathbf{H}}$ is bad) if and only if one of the following predicates is true:

1. $\mathtt{H}_1$: $\exists^* i, j \in [q]$ such that $\mathsf{X}_i = \mathsf{X}_j \wedge \mathsf{U}_i = \mathsf{U}_j$.

2. $\mathtt{H}_2$: $\exists^* i, j \in [q]$ such that $\mathsf{X}_i = \mathsf{X}_j \wedge \Delta_i = \Delta_j$.
3. $\mathtt{H}_3$: $\exists^* i, j \in [q]$ such that $\mathsf{U}_i = \mathsf{U}_j \wedge \Delta_i = \Delta_j$.
4. $\mathtt{H}_4$: $\exists^* i, j, k, \ell \in [q]$ such that $\mathsf{X}_i = \mathsf{X}_j \wedge \mathsf{U}_j = \mathsf{U}_k \wedge \mathsf{X}_k = \mathsf{X}_\ell$.
5. $\mathtt{H}_5$: $\exists^* i, j, k, \ell \in [q]$ such that $\mathsf{U}_i = \mathsf{U}_j \wedge \mathsf{X}_j = \mathsf{X}_k \wedge \mathsf{U}_k = \mathsf{U}_\ell$.
6. $\mathtt{H}_6$: $\exists k \geq 2^n/2q, \exists^* i_1, i_2, \ldots, i_k \in [q]$ such that $\mathsf{X}_{i_1} = \cdots = \mathsf{X}_{i_k}$.
7. $\mathtt{H}_7$: $\exists k \geq 2^n/2q, \exists^* i_1, i_2, \ldots, i_k \in [q]$ such that $\mathsf{U}_{i_1} = \cdots = \mathsf{U}_{i_k}$.

Case 1. $\widehat{\mathsf{H}}$ is bad: If the hash key $\widehat{\mathsf{H}}$ is bad, then Y^q and V^q values are sampled degenerately as $\mathsf{Y}_i = \mathsf{V}_i = 0$ for all $i \in [q]$. It means that we sample without maintaining any specific conditions, which will almost certainly lead to inconsistencies.

Case 2. $\widehat{\mathsf{H}}$ is good: To characterize the transcript corresponding to a good hash key, it will be useful to study a random bipartite edge-labeled graph associated with $(\mathsf{X}^q, \mathsf{U}^q, \Delta^q)$.

Definition 5 (Transcript Graph). *A transcript graph $\mathcal{G} = (\mathcal{X}, \mathcal{U}, \mathcal{E})$ associated with $(\mathsf{X}^q, \mathsf{U}^q, \Delta^q)$, denoted $\mathcal{G}(\mathsf{X}^q, \mathsf{U}^q, \Delta^q)$, is an undirected bipartite graph, where $\mathcal{X} := \{(\mathsf{X}_i, 0) : i \in [q]\}$ and $\mathcal{U} := \{(\mathsf{U}_i, 1) : i \in [q]\}$ are the two partitions of the vertex-set, and $\mathcal{E} := \{((\mathsf{X}_i, 0), (\mathsf{U}_i, 1)) : i \in [q]\}$ denotes the edge-set. We also associate the label Δ_i with edge $((\mathsf{X}_i, 0), (\mathsf{U}_i, 1)) \in \mathcal{E}$.*

For all practical purposes we may drop the partition markers 0 and 1, for each vertex $(\mathsf{X}_i, 0) \in \mathcal{X}$ and $(\mathsf{U}_i, 1) \in \mathcal{U}$, as they can be easily distinguished from the context and notations. Note that, the event $\mathsf{X}_i = \mathsf{X}_j$ and $\mathsf{U}_i = \mathsf{U}_j$, although extremely unlikely, will result in a parallel edge in $\mathcal{G}$. Finally, each edge $(\mathsf{X}_i, \mathsf{U}_i) \in \mathcal{E}$ corresponds to a query index $i \in [q]$, so we can equivalently view (and call) the edge $(\mathsf{X}_i, \mathsf{U}_i)$ as index (or query) i.

Consider the random transcript graph $\mathcal{G}(\mathsf{X}^q, \mathsf{U}^q)$ arising due to $\widehat{\mathsf{H}} \in \widehat{\mathcal{H}}_{\mathsf{good}}$. Lemma 5 and Figure 12 characterizes the different types of possible components in $\mathcal{G}(\mathsf{X}^q, \mathsf{U}^q)$.

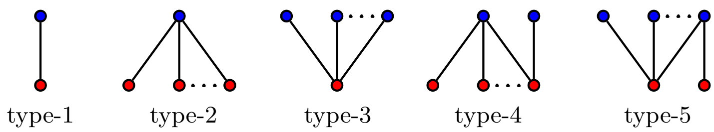

Fig. 12: Enumerating all possible types of components of a transcript graph corresponding to a good hash key: type-1 is the only possible component of size = 1 edge; type-2 and type-3 are star components with center in $\mathcal{X}$ and $\mathcal{U}$, respectively; type-4 and type-5 are the only possible components that are not isolated or star (can have degree 2 vertices in both $\mathcal{X}$ and $\mathcal{U}$). Note that, the vertex-coloring is only for illustration purposes.

Lemma 5. *The transcript graph $\mathcal{G}(\mathsf{X}^q, \mathsf{U}^q, \Delta^q)$ generated by a good hash key $\widehat{\mathsf{H}}$ has the following properties:*

1. *$\mathcal{G}$ is simple, acyclic and has no isolated vertices.*
2. *$\mathcal{G}$ has no two adjacent edges i and j such that $\Delta_i \oplus \Delta_j = 0$.*
3. *$\mathcal{G}$ has no component of size $\geq 2^n/2q$ edges.*
4. *$\mathcal{G}$ has no component such that it has 2 distinct degree 2 vertices in $\mathcal{X}$ or $\mathcal{U}$.*

In fact the all possible types of components in $\mathcal{G}$ are enumerated in Figure 12.

The proof of Lemma 5 is elementary and left as an exercise for the reader.

In what follows, we describe the sampling of Y^q and V^q conditioned on the fact that $\widehat{\mathsf{H}} \in \widehat{\mathcal{H}}_{\mathsf{good}}$. We collect the indices $i \in [q]$ corresponding to the edges in all type-1, type-2, type-3, type-4, and type-5 components, in the index sets $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_3$, $\mathcal{I}_4$, and $\mathcal{I}_5$, respectively. Clearly, the five sets are disjoint, and $[q] = \mathcal{I}_1 \sqcup \mathcal{I}_2 \sqcup \mathcal{I}_3 \sqcup \mathcal{I}_4 \sqcup \mathcal{I}_5$. Let $\mathcal{I} = \mathcal{I}_1 \sqcup \mathcal{I}_2 \sqcup \mathcal{I}_3$. Consider a constrained system of equations

$$\mathcal{L} = \{Y_i \oplus V_i = \Delta_i \ : \ i \in \mathcal{I}\},$$

with the constraint

$$\phi : \ \mathsf{X}^q \leftrightsquigarrow Y^q \wedge \mathsf{U}^q \leftrightsquigarrow V^q.$$

The solution space for $\mathcal{L}$, satisfying the constraint ϕ, is precisely the set

$$\mathcal{S} = \{(y^{\mathcal{I}}, v^{\mathcal{I}}) : y^{\mathcal{I}} \leftrightsquigarrow \mathsf{X}^{\mathcal{I}} \wedge v^{\mathcal{I}} \leftrightsquigarrow \mathsf{U}^{\mathcal{I}} \wedge y^{\mathcal{I}} \oplus v^{\mathcal{I}} = \Delta^{\mathcal{I}}\}.$$

Given these definitions, the ideal oracle $\mathcal{O}_0$ samples $(\mathsf{Y}^q, \mathsf{V}^q)$ as follows:

- $(\mathsf{Y}^{\mathcal{I}}, \mathsf{V}^{\mathcal{I}}) \leftarrow\!\!\$\ \mathcal{S}$, i.e., $\mathcal{O}_0$ uniformly samples one valid assignment from the set of all valid assignments for $\mathsf{Y}^{\mathcal{I}}$ and $\mathsf{V}^{\mathcal{I}}$.
- Let $\mathcal{G} \setminus \mathcal{C}_{\mathcal{I}}$ denote the subgraph of $\mathcal{G}$ after the removal of all type-1, type-2, and type-3 components. For each component $\mathcal{C}$ of $\mathcal{G} \setminus \mathcal{C}_{\mathcal{I}}$:
  - Suppose $(\mathsf{X}_i, \mathsf{U}_i) \in \mathcal{C}$ corresponds to an edge in $\mathcal{C}$, where both X_i and U_i have degree ≥ 2. Then, $\mathsf{Y}_i \leftarrow\!\!\$\ \{0,1\}^n$ and $\mathsf{V}_i = \mathsf{Y}_i \oplus \Delta_i$.
  - For each edge $(\mathsf{X}_{i'}, \mathsf{U}_{i'}) \neq (\mathsf{X}_i, \mathsf{U}_i) \in \mathcal{C}$, either $\mathsf{X}_{i'} = \mathsf{X}_i$ or $\mathsf{U}_{i'} = \mathsf{U}_i$. Suppose, $\mathsf{X}_{i'} = \mathsf{X}_i$. Then, $\mathsf{Y}_{i'} = \mathsf{Y}_i$ and $\mathsf{V}_{i'} = \mathsf{Y}_{i'} \oplus \Delta_{i'}$. Now, suppose $\mathsf{U}_{i'} = \mathsf{U}_i$. Then, $\mathsf{V}_{i'} = \mathsf{V}_i$ and $\mathsf{Y}_{i'} = \mathsf{V}_{i'} \oplus \Delta_{i'}$.

At this point, $\Theta_0 = (\mathsf{T}^q, \mathsf{M}^q, \mathsf{C}^q, \mathsf{X}^q, \mathsf{Y}^q, \mathsf{V}^q, \mathsf{U}^q, \Delta^q, \widetilde{\mathsf{H}}_1, \widetilde{\mathsf{H}}_2, \mathsf{H})$ is completely defined. In this way we maintain both the consistency of equations of the form $\mathsf{Y}_i \oplus \mathsf{V}_i = \Delta_i$ (as in the case of real world), and the permutation consistency within each component, given that $\widehat{\mathsf{H}} \in \widehat{\mathcal{H}}_{\mathsf{good}}$. However, there might be collisions among Y or V values from different components.

5.3 Definition and Analysis of Bad Transcripts

Given the description of the transcript random variable corresponding to the ideal oracle we can define the set of transcripts Ω as the set of all tuples $\omega = (t^q, m^q, c^q, x^q, y^q, v^q, u^q, \delta^q, \widetilde{h}_1, \widetilde{h}_2, h)$, where $t^q \in (\{0,1\}^\tau)^q$; $m^q, c^q, y^q, v^q \in (\{0,1\}^n)^q$; $\widehat{h} = (\widetilde{h}_1, \widetilde{h}_2, h) \in \widehat{\mathcal{H}}$; $x^q = \widetilde{h}_1(t^q, m^q)$; $u^q = \widetilde{h}_2(t^q, c^q)$; $\delta^q = h(t^q)$; and $(t^q, m^q) \leftrightsquigarrow (t^q, c^q)$.

Our bad transcript definition is inspired by two requirements:

1. Eliminate all x^q, u^q, and δ^q tuples such that both y^q and v^q are trivially restricted by way of linear dependence. For example, consider the condition $\mathtt{H}_2$. This leads to $y_i = y_j$, which would imply $v_i = y_i \oplus \delta_i = y_j \oplus \delta_j = v_j$. Assuming $i > j$, v_i is trivially restricted $(= v_j)$ by way of linear dependence. This may lead to $u^q \not\leftrightsquigarrow v^q$ as u_i may not be equal to u_j.
2. Eliminate all x^q, u^q, y^q, v^q tuples such that $x^q \not\leftrightsquigarrow y^q$ or $u^q \not\leftrightsquigarrow v^q$.

Among the two, requirement 2 is trivial as $x^q \leftrightsquigarrow y^q$ and $u^q \leftrightsquigarrow v^q$ is always true for real world transcript. Requirement 1 is more of a technical one that helps in the ideal world sampling of y^q and v^q.

Bad Transcript Definition: Throughout the discussion, we consider the transcript

$$\omega = (t^q, m^q, c^q, x^q, y^q, v^q, u^q, \delta^q, \widehat{h})$$

to characterize the bad transcripts.

We first designate certain transcripts as bad depending upon the characterization of hash keys. Inspired by the ideal world description, we say that a hash key $\widehat{h} \in \widehat{\mathcal{H}}_{\mathsf{bad}}$ (or $\widehat{h}$ is bad) if and only if the following predicate is true:

$$\mathtt{H}_1 \vee \mathtt{H}_2 \vee \mathtt{H}_3 \vee \mathtt{H}_4 \vee \mathtt{H}_5 \vee \mathtt{H}_6 \vee \mathtt{H}_7.$$

We say that ω is *hash-induced* bad transcript, if $\widehat{h} \in \mathcal{H}_{\mathsf{bad}}$. We write this event as $\mathtt{BAD1}$, and by a slight abuse of notations,[4] we have

$$\mathtt{BAD1} = \bigcup_{i=1}^{7} \mathtt{H}_i. \tag{14}$$

This takes care of the first requirement. For the second one we have to enumerate all the conditions which might lead to $x^q \not\leftrightsquigarrow y^q$ or $u^q \not\leftrightsquigarrow v^q$. Since we sample degenerately when the hash key is bad, the transcript is *trivially inconsistent* in this case. For good hash keys, if $x_i = x_j$ (or $u_i = u_j$) then we always have $y_i = y_j$ (res. $v_i = v_j$); hence the inconsistency won't arise. So, given that the hash key is good, we say that ω is *sampling-induced* bad transcript, if one of the following conditions is true:
for some $\alpha \in [5]$ and $\beta \in \{\alpha, \ldots, 5\}$, we have

- $\mathtt{Ycoll}_{\alpha\beta}:\ \exists i \in \mathcal{I}_\alpha, j \in \mathcal{I}_\beta$, such that $x_i \neq x_j \wedge y_i = y_j$, and
- $\mathtt{Vcoll}_{\alpha\beta}:\ \exists i \in \mathcal{I}_\alpha, j \in \mathcal{I}_\beta$, such that $u_i \neq u_j \wedge v_i = v_j$,

[4] We use the notation $\mathtt{H}_i$ to denote the event that the predicate $\mathtt{H}_i$ is true.

where $\mathcal{I}_i$ is defined as before in section 5.2. By varying α and β over all possible values, we get all 30 conditions which might lead to $x^q \not\leftrightarrow y^q$ or $u^q \not\leftrightarrow v^q$. Here we remark that some of these 30 conditions are never satisfied due to the sampling mechanism prescribed in section 5.2. These are $\texttt{Ycoll}_{11}$, $\texttt{Ycoll}_{12}$, $\texttt{Ycoll}_{13}$, $\texttt{Ycoll}_{22}$, $\texttt{Ycoll}_{23}$, $\texttt{Ycoll}_{33}$, $\texttt{Vcoll}_{11}$, $\texttt{Vcoll}_{12}$, $\texttt{Vcoll}_{13}$, $\texttt{Vcoll}_{22}$, $\texttt{Vcoll}_{23}$, and $\texttt{Vcoll}_{33}$. We listed them here only for the sake of completeness. We write the combined event that one of the 30 conditions hold as $\texttt{BAD2}$. Again by an abuse of notations, we have

$$\texttt{BAD2} = \bigcup_{\alpha \in [5], \beta \in \{\alpha, \ldots, 5\}} \left(\texttt{Ycoll}_{\alpha\beta} \cup \texttt{Vcoll}_{\alpha\beta} \right). \tag{15}$$

Finally, a transcript ω is called bad, i.e. $\omega \in \Omega_{\mathsf{bad}}$, if it is either a hash-induced or a sampling-induced bad transcript. All other transcripts are called good. It is easy to see that all good transcripts are attainable (as required in the H-coefficient technique or the expectation method).

Bad Transcript Analysis: We analyze the probability of realizing a bad transcript in the ideal world. By definition, this is possible if and only if one of $\texttt{BAD1}$ or $\texttt{BAD2}$ occurs. So, we have

$$\begin{aligned} \epsilon_{\mathsf{bad}} = \mathsf{Pr}\left(\Theta_0 \in \Omega_{\mathsf{bad}}\right) &= \mathsf{Pr}_{\Theta_0}\left(\texttt{BAD1} \cup \texttt{BAD2}\right) \\ &\leq \underbrace{\mathsf{Pr}_{\Theta_0}\left(\texttt{BAD1}\right)}_{\epsilon_{\mathsf{h}}} + \underbrace{\mathsf{Pr}_{\Theta_0}\left(\texttt{BAD2}\right)}_{\epsilon_{\mathsf{s}}}. \end{aligned} \tag{16}$$

Lemma 6 upper bounds ϵ_{h} to $q^2\epsilon_1^2 + q^2\epsilon_1\epsilon_2 + 2q^2\epsilon_1^{1.5} + 16q^4\epsilon_1 2^{-2n}$ and Lemma 7 upper bounds ϵ_{s} to $9q^4\epsilon_1^2 2^{-n}$. Substituting these values in (16), we get

$$\epsilon_{\mathsf{bad}} \leq q^2\epsilon_1^2 + q^2\epsilon_1\epsilon_2 + 2q^2\epsilon_1^{1.5} + \frac{16q^4\epsilon_1}{2^{2n}} + \frac{9q^4\epsilon_1^2}{2^n}. \tag{17}$$

Lemma 6. $\epsilon_{\mathsf{h}} \leq q^2\epsilon_1^2 + q^2\epsilon_1\epsilon_2 + 2q^2\epsilon_1^{1.5} + \dfrac{16q^4\epsilon_1}{2^{2n}}$.

Proof. Using (14) and (16), we have

$$\epsilon_{\mathsf{h}} = \mathsf{Pr}\left(\texttt{H}_1 \cup \texttt{H}_2 \cup \texttt{H}_3 \cup \texttt{H}_4 \cup \texttt{H}_5 \cup \texttt{H}_6 \cup \texttt{H}_7\right) \leq \sum_{i=1}^{7} \mathsf{Pr}\left(\texttt{H}_i\right).$$

$\texttt{H}_1$ is true if for some distinct i, j both $\mathsf{X}_i = \mathsf{X}_j$, and $\mathsf{U}_i = \mathsf{U}_j$. Now $\mathsf{T}_i = \mathsf{T}_j \implies \mathsf{M}_i \neq \mathsf{M}_j$. Hence $\mathsf{X}_i \neq \mathsf{X}_j$ (since $\widetilde{\mathsf{H}}_1$ is a tweakable permutation) and $\texttt{H}_1$ is not true. So suppose $\mathsf{T}_i \neq \mathsf{T}_j$. Then, using the fact that $\widetilde{\mathcal{H}}$ is an ϵ-AUHF and KG is a PISM, for a fixed i, j we get an upper bound of ϵ_1^2. Furthermore, we have at most $\binom{q}{2}$ pairs of (i, j). Thus, $\mathsf{Pr}\left(\texttt{H}_1\right) \leq \binom{q}{2}\epsilon_1^2$.

Following a similar line of argument one can bound $\mathsf{Pr}\left(\texttt{H}_2\right) \leq \binom{q}{2}\epsilon_1\epsilon_2$ and $\mathsf{Pr}\left(\texttt{H}_3\right) \leq \binom{q}{2}\epsilon_1\epsilon_2$.

In the remaining, we bound the probability of $\mathtt{H}_4$ and $\mathtt{H}_6$, while the probability of $\mathtt{H}_5$ and $\mathtt{H}_7$ can be bounded analogously. Now, $\mathtt{H}_4$ is true if for some pairwise distinct i, j, k, ℓ,

$$\widetilde{\mathsf{H}}_1(\mathsf{T}_i, \mathsf{M}_i) = \widetilde{\mathsf{H}}_1(\mathsf{T}_j, \mathsf{M}_j)\widetilde{\mathsf{H}}_2(\mathsf{T}_j, \mathsf{C}_j) \quad = \widetilde{\mathsf{H}}_2(\mathsf{T}_k, \mathsf{C}_k)\widetilde{\mathsf{H}}_1(\mathsf{T}_k, \mathsf{M}_k) = \widetilde{\mathsf{H}}_1(\mathsf{T}_\ell, \mathsf{M}_\ell).$$

Again, using the fact that KG is a PISM, we have that the second equation is independent of the other two equations. Using Lemma 2, we have

$$\mathsf{Pr}\left(\mathtt{H}_4\right) \leq q^2 \epsilon_1^{1.5}.$$

For $\mathtt{H}_6$, for some $i_1, \ldots, i_k$, we have

$$\mathsf{X}_{i_1} = \mathsf{X}_{i_2} = \cdots = \mathsf{X}_{i_k},$$

where $k \geq 2^n/2q$. Since, $(t_{i_j}, m_{i_j}) \neq (t_{i_l}, m_{i_l})$ for all $j \neq l$, we can apply Corollary 2 to get

$$\mathsf{Pr}\left(\mathtt{H}_6\right) \leq \frac{8q^4\epsilon_1}{2^{2n}}. \qquad \square$$

Lemma 7. $\epsilon_{\mathsf{s}} \leq \dfrac{9q^4\epsilon_1^2}{2^n}$.

Proof. Using (15) and (16), we have

$$\begin{aligned}\epsilon_{\mathsf{s}} &= \mathsf{Pr}\left(\bigcup_{\alpha\in[5],\beta\in\{\alpha,\ldots,5\}} (\mathtt{Ycoll}_{\alpha\beta} \cup \mathtt{Vcoll}_{\alpha\beta})\right) \\ &\leq \sum_{\alpha\in[5]} \sum_{\beta\in\{\alpha,\ldots,5\}} \left(\mathsf{Pr}\left(\mathtt{Ycoll}_{\alpha\beta}\right) + \mathsf{Pr}\left(\mathtt{Vcoll}_{\alpha\beta}\right)\right).\end{aligned}$$

We bound the probabilities of the events on the right hand side in groups as given below:

1. Bounding $\sum_{\alpha\in[3],\beta\in\{\alpha,\ldots,3\}} \mathsf{Pr}\left(\mathtt{Ycoll}_{\alpha\beta}\right) + \mathsf{Pr}\left(\mathtt{Vcoll}_{\alpha\beta}\right)$: Recall that the sampling of Y and V values is always done consistently for indices belonging to $\mathcal{I} = \mathcal{I}_1 \sqcup \mathcal{I}_2 \sqcup \mathcal{I}_3$. Hence,

$$\sum_{\alpha\in[3],\beta\in\{\alpha,\ldots,3\}} \mathsf{Pr}\left(\mathtt{Ycoll}_{\alpha\beta}\right) + \mathsf{Pr}\left(\mathtt{Vcoll}_{\alpha\beta}\right) = 0, \tag{18}$$

2. Bounding $\sum_{\alpha\in[3],\beta\in\{4,5\}} \mathsf{Pr}\left(\mathtt{Ycoll}_{\alpha\beta}\right) + \mathsf{Pr}\left(\mathtt{Vcoll}_{\alpha\beta}\right)$: Let's consider the event $\mathtt{Ycoll}_{14}$, which translates to there exist indices $i \in \mathcal{I}_1$ and $j \in \mathcal{I}_4$ such that $\mathsf{X}_i \neq \mathsf{X}_j \wedge \mathsf{Y}_i = \mathsf{Y}_j$. Since $j \in \mathcal{I}_4$, there must exist $k, \ell \in \mathcal{I}_4 \setminus \{j\}$, such that one of the following happens

$$\begin{aligned}\mathsf{X}_j &= \mathsf{X}_k \wedge \mathsf{U}_k = \mathsf{U}_\ell \\ \mathsf{U}_j &= \mathsf{U}_k \wedge \mathsf{X}_k = \mathsf{X}_\ell \\ \mathsf{X}_j &= \mathsf{X}_k \wedge \mathsf{U}_j = \mathsf{U}_\ell.\end{aligned}$$

We analyze the first case, while the other two cases can be similarly bounded. To bound the probability of $\texttt{Ycoll}_{14}$, we can look at the joint event

$$\texttt{E}: \ \exists i \in \mathcal{I}_1, \exists^* j, k, \ell \in \mathcal{I}_4, \text{ such that } \mathsf{Y}_i = \mathsf{Y}_j \wedge \mathsf{X}_j = \mathsf{X}_k \wedge \mathsf{U}_k = \mathsf{U}_\ell.$$

Note that the event $\mathsf{Y}_i = \mathsf{Y}_j$ occurs with exactly 2^{-n} probability conditioned on the event $\mathsf{X}_j = \mathsf{X}_k \wedge \mathsf{U}_k = \mathsf{U}_\ell$. Thus, we get

$$\begin{aligned}\mathsf{Pr}\left(\texttt{E}\right) &= \mathsf{Pr}\left(\exists i \in \mathcal{I}_1, \exists^* j, k, \ell \in \mathcal{I}_4, \text{ such that } \mathsf{Y}_i = \mathsf{Y}_j \wedge \mathsf{X}_j = \mathsf{X}_k \wedge \mathsf{U}_k = \mathsf{U}_\ell\right) \\ &\leq \sum_{i \in \mathcal{I}_1} \sum_{j<k<\ell \in \mathcal{I}_4} \mathsf{Pr}\left(\mathsf{X}_j = \mathsf{X}_k \wedge \mathsf{U}_k = \mathsf{U}_\ell\right) \times \mathsf{Pr}\left(\mathsf{Y}_i = \mathsf{Y}_j \mid \mathsf{X}_j = \mathsf{X}_k \wedge \mathsf{U}_k = \mathsf{U}_\ell\right) \\ &\leq q\binom{q}{3}\frac{\epsilon_1^2}{2^n},\end{aligned}$$

where the last inequality follows from the AUHF property of $\widetilde{\mathcal{H}}$, the PISM property of KG, and the uniform randomness of Y_j. The probability of the other two cases are identically bounded, whence we get

$$\mathsf{Pr}\left(\texttt{Ycoll}_{14}\right) \leq 3q\binom{q}{3}\frac{\epsilon_1^2}{2^n}.$$

We can bound the probabilities of $\texttt{Ycoll}_{24}$, $\texttt{Ycoll}_{34}$, $\texttt{Ycoll}_{\alpha 5}$, $\texttt{Vcoll}_{\alpha 4}$, and $\texttt{Vcoll}_{\alpha 5}$, for $\alpha \in [3]$, in a similar manner as in the case of $\texttt{Ycoll}_{14}$. So, we skip the argumentation for these cases, and summarize the probability for this group as

$$\sum_{\alpha \in [3], \beta \in \{4,5\}} \mathsf{Pr}\left(\texttt{Ycoll}_{\alpha\beta}\right) + \mathsf{Pr}\left(\texttt{Vcoll}_{\alpha\beta}\right) \leq \frac{6q^4\epsilon_1^2}{2^n}. \tag{19}$$

3. Bounding $\sum_{\alpha \in \{4,5\}, \beta \in \{\alpha,5\}} \mathsf{Pr}\left(\texttt{Ycoll}_{\alpha\beta}\right) + \mathsf{Pr}\left(\texttt{Vcoll}_{\alpha\beta}\right)$: Consider the event $\texttt{Ycoll}_{44}$, which translates to there exists distinct indices $i, j \in \mathcal{I}_4$ such that $\mathsf{X}_i \neq \mathsf{X}_j \wedge \mathsf{Y}_i = \mathsf{Y}_j$. Here as $i, j \in \mathcal{I}_4$, there must exist $k, \ell \in \mathcal{I}_4 \setminus \{j\}$ such that one of the following happens

$$\begin{aligned}\mathsf{X}_j &= \mathsf{X}_k \wedge \mathsf{U}_k = \mathsf{U}_\ell \\ \mathsf{U}_j &= \mathsf{U}_k \wedge \mathsf{X}_k = \mathsf{X}_\ell \\ \mathsf{X}_j &= \mathsf{X}_k \wedge \mathsf{U}_j = \mathsf{U}_\ell.\end{aligned}$$

The analysis of these cases is similar to 2 above. So, we skip it and provide the final bound

$$\mathsf{Pr}\left(\texttt{Ycoll}_{44}\right) \leq 3q\binom{q}{3}\frac{\epsilon_1^2}{2^n}.$$

The probabilities of all the remaining events in this group can be bounded in a similar fashion.

$$\sum_{\alpha \in \{4,5\}, \beta \in \{\alpha,5\}} \mathsf{Pr}\left(\texttt{Ycoll}_{\alpha\beta}\right) + \mathsf{Pr}\left(\texttt{Vcoll}_{\alpha\beta}\right) \leq \frac{3q^4\epsilon_1^2}{2^n}. \tag{20}$$

The result follows by combining (18)-(20), followed by some simplifications. □

5.4 Good Transcript Analysis

From section 5.2, we know the types of components present in the transcript graph corresponding to a good transcript ω are exactly as in Figure 12. Let $\omega = (t^q, m^q, c^q, x^q, y^q, v^q, u^q, \delta^q, \widetilde{h}_1, \widetilde{h}_2, h)$ be the good transcript at hand. From the bad transcript description of section 5.3, we know that for a good transcript $(t^q, m^q) \leftrightsquigarrow (t^q, c^q)$, $x^q \leftrightsquigarrow y^q$, $v^q \leftrightsquigarrow u^q$, and $y^q \oplus v^q = \delta^q$.

First, we add some new parameters with respect to ω to aid the remaining analysis.

For $i \in [5]$, let $c_i(\omega)$ and $q_i(\omega)$ denote the number of components and number of indices (corresponding to the edges), respectively of type-i in ω. Note that $q_1(\omega) = c_1(\omega)$, $q_i(\omega) \geq 2c_i(\omega)$ for $i \in \{2,3\}$, and $q_i(\omega) \geq 3c_i(\omega)$ for $i \in \{4,5\}$. Obviously, for a good transcript $q = \sum_{i=1}^{5} q_i(\omega)$.

Let $(t'_1, t'_2, \cdots, t'_r)$ be an arbitrary ordering of $\mathtt{S}(t^q)$, and for all $i \in [r]$, let μ_i denote the multiplicity of t'_i in the multiset $\mathtt{M}(t^q)$, i.e., $r \leq q$ and $\sum_{i=1}^{r} \mu_i = q$. In addition, let μ'_i denote the multiplicity of t'_i in the multiset $\mathtt{M}(t^{\mathcal{I}})$, i.e., $\sum_{i=1}^{r} \mu'_i = |\mathcal{I}|$.

Let $(\delta'_1, \delta'_2, \cdots, \delta'_s)$ be an arbitrary ordering of $\mathtt{S}(\delta^{\mathcal{I}})$, and for all $i \in [s]$, let ν_i denote the multiplicity of δ'_i in the multiset $\mathtt{M}(\delta^{\mathcal{I}})$, i.e., $s \leq |\mathcal{I}|$ and $\sum_{i=1}^{s} \nu_i = |\mathcal{I}|$.

For all these parameters, we will drop the ω parametrization whenever it is understood from the context.

Interpolation probability for the real oracle: In the real oracle, $\widehat{\mathsf{H}} \leftarrow \mathsf{KG}\left(\widehat{\mathcal{H}}\right)$, $\boldsymbol{\pi}_1$ is called exactly $p_1 + 2c_4 + q_5 - c_5$ times and $\boldsymbol{\pi}_2$ is called exactly $p_2 + q_4 - c_4 + 2c_5$ times, where $p_1 := q_1 + c_2 + q_3$ and $p_2 := q_1 + q_2 + c_3$. Thus, we have

$$\mathsf{Pr}\left(\Theta_1 = \omega\right) = \mathsf{Pr}_{\mathsf{KG}}\left(\widehat{\mathsf{H}} = \widehat{h}\right) \times \frac{1}{(2^n)_{p_1+2c_4+q_5-c_5}} \times \frac{1}{(2^n)_{p_2+q_4-c_4+2c_5}}. \quad (21)$$

Interpolation probability for the ideal oracle: In the ideal oracle, the sampling is done in parts:

I. $\widetilde{\boldsymbol{\pi}}$ *sampling*: We have

$$\mathsf{Pr}\left(\widetilde{\boldsymbol{\pi}}(t^q, m^q) = c^q\right) \leq \frac{1}{\prod_{i=1}^{r} (2^n)_{\mu_i}}.$$

II. *Hash key sampling*: This is identical to the real world, and simply given by $\mathsf{Pr}_{\mathsf{KG}}\left(\widehat{\mathsf{H}} = \widehat{h}\right)$.

III. *Internal variables sampling*: The internal variables Y^q and V^q are sampled in two stages.

(A). *type-*1*, type-*2 *and type-*3 *sampling*: Recall the sets $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$, from section 5.3. Consider the system of equation

$$\mathcal{L} = \{\mathsf{Y}_i \oplus \mathsf{V}_i = \delta_i \ : \ i \in \mathcal{I}\}.$$

From Figure 12 we know that $\mathcal{L}$ is cycle-free and non-degenerate. Further, $\xi_{\max}(\mathcal{L}) \leq 2^n/2q$, since the transcript is good. So, we can apply

Theorem 1 to get a lower bound on the the number of valid solutions, $|\mathcal{S}(\mathcal{L})|$ for $\mathcal{L}$. Using the fact that $(\mathsf{Y}^{\mathcal{I}}, \mathsf{V}^{\mathcal{I}}) \leftarrow\!\!\$\, \mathcal{S}(\mathcal{L})$, and Theorem 1, we have

$$\mathsf{Pr}\left((\mathsf{Y}^{\mathcal{I}}, \mathsf{V}^{\mathcal{I}}) = (y^{\mathcal{I}}, v^{\mathcal{I}})\right) \leq \frac{\prod_{i=1}^{s}(2^n)_{\nu_i}}{\zeta(\omega)(2^n)_{q_1+c_2+q_3}(2^n)_{q_1+q_2+c_3}},$$

where

$$\zeta(\omega) = \left(1 - \frac{13q^4}{2^{3n}} - \frac{2q^2}{2^{2n}} - \left(\sum_{i=1}^{c_2+c_3} \eta_{c_1+i}^2\right)\frac{4q^2}{2^{2n}}\right),$$

and η_i denotes the number of edges in the i-th component for all $i \in [c_1 + c_2 + c_3]$.

(B). *type-4, and type-5 sampling*: For the remaining indices, one value is sampled uniformly for each of the components, i.e. we have

$$\mathsf{Pr}\left(\left(\mathsf{Y}^{[q]\setminus\mathcal{I}}, \mathsf{V}^{[q]\setminus\mathcal{I}}\right) = \left(y^{[q]\setminus\mathcal{I}}, v^{[q]\setminus\mathcal{I}}\right)\right) = \frac{1}{2^{n(c_4+c_5)}}.$$

By combining I, II, III, and rearranging the terms, we have

$$\mathsf{Pr}\left(\Theta_0 = \omega\right) \leq \mathsf{Pr}_{\mathsf{KG}}\left(\widehat{\mathsf{H}} = \widehat{h}\right) \times \frac{1}{\zeta(\omega)} \times \frac{\prod_{i=1}^{s}(2^n)_{\nu_i}}{\prod_{i=1}^{r}(2^n)_{\mu_i}(2^n)_{p_1}(2^n)_{p_2}(2^n)^{c_4+c_5}}. \tag{22}$$

5.5 Ratio of Interpolation Probabilities

On dividing (21) by (22), and simplifying the expression, we get

$$\begin{aligned}
\frac{\mathsf{Pr}\left(\Theta_1 = \omega\right)}{\mathsf{Pr}\left(\Theta_0 = \omega\right)} &\geq \zeta(\omega) \cdot \frac{\prod_{i=1}^{r}(2^n)_{\mu_i}}{\prod_{i=1}^{s}(2^n)_{\nu_i}(2^n - p_1 - c_4)_{c_4+q_5-c_5}(2^n - p_2 - c_5)_{q_4-c_4+c_5}} \\
&\overset{1}{\geq} \zeta(\omega) \cdot \frac{\prod_{i=1}^{r}(2^n)_{\mu_i'}\prod_{i=1}^{r}(2^n - \mu_i')_{\mu_i-\mu_i'}}{\prod_{i=1}^{s}(2^n)_{\nu_i}(2^n - p_1 - c_4)_{c_4+q_5-c_5}(2^n - p_2 - c_5)_{q_4-c_4+c_5}} \\
&\overset{2}{\geq} \left.\zeta(\omega) \cdot \frac{\prod_{i=1}^{r}(2^n - \mu_i')_{\mu_i-\mu_i'}}{(2^n - p_1 - c_4)_{c_4+q_5-c_5}(2^n - p_2 - c_5)_{q_4-c_4+c_5}}\right\} A \\
&\overset{3}{\geq} \zeta(\omega).
\end{aligned} \tag{23}$$

At inequality 1, we simply rewrite the numerator. Further, $r \geq s$, as number of distinct internal masking values is at most the number of distinct tweaks, and $\mathtt{S}(t^{\mathcal{I}})$ compresses to $\mathtt{S}(\delta^{\mathcal{I}})$. So, using Proposition 1, we can justify inequality 2. At inequality 2, for $i \in \{2, 3, 4, 5\}$, $c_i(\omega) > 0$ if and only if $r \geq 2$. Also, $\mu_i' \leq c_1 + c_2 + c_3 \leq p_1 + c_4$ and $\mu_i' \leq p_2 + c_5$ for $i \in [r]$. Similarly, $\mu_i \leq c_1 + c_2 + c_3 + 2c_4 + 2c_5 \leq p_1 + 2c_4 + q_5 - c_5$, and $\mu_i \leq p_2 + q_4 - c_4 + 2c_5$. Also, $\sum_{i=1}^{r} \mu_i - \mu_i' = q_4 + q_5$. Thus, A satisfies the conditions laid out in Proposition 2, and hence $A \geq 1$. This justifies inequality 3.

We define $\epsilon_{\mathsf{ratio}} : \Omega \to [0, \infty)$ by the mapping

$$\epsilon_{\mathsf{ratio}}(\omega) = 1 - \zeta(\omega).$$

Clearly $\epsilon_{\mathsf{ratio}}$ is non-negative and the ratio of real to ideal interpolation probabilities is at least $1 - \epsilon_{\mathsf{ratio}}(\omega)$ (using (23)). Thus, we can use the expectation method to get

$$\mathbf{Adv}^{\mathsf{tsprp}}_{\mathsf{LRW+}}(q) \leq \frac{2q^2}{2^{2n}} + \frac{13q^4}{2^{3n}} + \frac{4q^2}{2^{2n}} \mathsf{Ex}\left(\sum_{i=1}^{c_2+c_3} \eta^2_{c_1+i}\right) + \epsilon_{\mathsf{bad}}. \tag{24}$$

Let $\sim_1$ (res. $\sim_2$) be an equivalence relation over $[q]$, such that $\alpha \sim_1 \beta$ (res. $\alpha \sim_2 \beta$) if and only if $\mathsf{X}_\alpha = \mathsf{X}_\beta$ (res. $\mathsf{U}_\alpha = \mathsf{U}_\beta$). Now, each η_i random variable denotes the cardinality of some non-singleton equivalence class of $[q]$ with respect to either $\sim_1$ or $\sim_2$. Let $\mathcal{P}^1_1, \ldots, \mathcal{P}^1_r$ and $\mathcal{P}^2_1, \ldots, \mathcal{P}^2_s$ denote the non-singleton equivalence classes of $[q]$ with respect to $\sim_1$ and $\sim_2$, respectively. Further, for $i \in [r]$ and $j \in [s]$, let $\mathsf{n}_i = |\mathcal{P}^1_i|$ and $\mathsf{n}'_j = |\mathcal{P}^2_j|$. Then, we have

$$\begin{aligned} \mathsf{Ex}\left(\sum_{i=1}^{c_2+c_3} \eta^2_{c_1+i}\right) &\leq \mathsf{Ex}\left(\sum_{j=1}^{r} {\mathsf{n}_j}^2\right) + \mathsf{Ex}\left(\sum_{k=1}^{s} {\mathsf{n}'_k}^2\right) \\ &\leq 4q^2\epsilon_1. \end{aligned} \tag{25}$$

where the first inequality follows from linearity, and the second inequality follows from Lemma 4. Theorem 5 then follows from (17), (24), and (25). □

6 Instantiating LRW+

In this section, we show that any cascaded LRW construction with $r \geq 2$ rounds can be viewed as an instance of LRW+. Thus, they can be proven secure up to $2^{3n/4}$ queries provided the derived hash functions are 2^{-n}-universal. Note that, it would be sufficient to define $\widetilde{\mathsf{H}}_1$, $\widetilde{\mathsf{H}}_2$, H, $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ for each construction. In the following discussion, let $\boldsymbol{\pi}'^r \leftarrow\!\$\ \mathsf{Perm}(n)$ and $\mathsf{H}'^r \leftarrow\!\$\ \mathcal{H}^r$, where $\mathcal{H}$ is an ϵ-AUHF.

6.1 Cascaded LRW1

For $r \geq 2$, the r-$\mathsf{LRW1}[\boldsymbol{\pi}^r]$ construction takes as input $(t, m) \in \{0,1\}^n \times \{0,1\}^n$ and returns $c \in \{0,1\}^n$, which is defined as follows:
Let $y_0 = m$ and for all $i \in [r]$:

$$\begin{aligned} x_i &:= t \oplus y_{i-1}, \\ y_i &:= \boldsymbol{\pi}'_i(x_i), \end{aligned}$$

and finally $c := y_r$. The inverse of r-LRW1 is analogously defined.

Cascaded $\mathsf{LRW1}$ as an Instance of $\mathsf{LRW+}$. For some $r \geq 2$, $r' = \lfloor r/2 \rfloor$, and any (t, k, m) such that r-$\mathsf{LRW1}(t, m) = c$, let

$$\widetilde{\mathsf{H}}_1(t, m) := x_{r'} \qquad \mathsf{H}(t) := t \qquad \widetilde{\mathsf{H}}_2(t, c) := y_{r'+1},$$

and

$$\boldsymbol{\pi}_1 := \boldsymbol{\pi}'_{r'} \qquad \boldsymbol{\pi}_2 := \boldsymbol{\pi}'^{-1}_{r'+1}.$$

Clearly, the $\mathsf{LRW+}$ instance so defined is same as r-$\mathsf{LRW1}$. Furthermore, assuming $r \geq 4$, $\boldsymbol{\pi}'^r \leftarrow\!\!\$\ \mathsf{Perm}(n)$, KG is a PISM, $\widetilde{\mathsf{H}}_1$ and $\widetilde{\mathsf{H}}_2$ are $(2^n-1)^{-1}$-AUTPF, and H is 0-AUHF. Thus, using Theorem 5, we have the following corollary on the security of cascaded $\mathsf{LRW1}$.

Corollary 3. *For $r \geq 4$, we have*

$$\mathbf{Adv}^{\mathsf{tsprp}}_{r\text{-}\mathsf{LRW1}}(q) \leq \frac{2q^2}{(2^n-1)^{1.5n}} + \frac{54q^4}{(2^n-1)^3} + \frac{3q^2}{(2^n-1)^2}.$$

In particular, for $r = 4$, we have proved CCA security for 4-$\mathsf{LRW1}$ up to $2^{3n/4}$ queries.

6.2 Cascaded $\mathsf{LRW2}$

For $r \geq 1$, the r-$\mathsf{LRW2}[\boldsymbol{\pi}^r, \mathsf{H}'^r]$ construction takes as input $(t, m) \in \{0,1\}^\tau \times \{0,1\}^n$ and returns $c \in \{0,1\}^n$, which is defined as follows:
Let $y_0 = m$, H'_0 be a constant function that returns 0^n, and for all $i \in [r]$:

$$\begin{aligned} x_i &:= \mathsf{H}'_{i-1}(t) \oplus \mathsf{H}'_i(t) \oplus y_{i-1}, \\ y_i &:= \boldsymbol{\pi}'_i(x_i), \end{aligned}$$

and finally $c := \mathsf{H}'_r(t) \oplus y_r$. The inverse of r-$\mathsf{LRW2}$ is analogously defined.

Cascaded $\mathsf{LRW2}$ as an Instance of $\mathsf{LRW+}$. For some $r \geq 2$, $r' = \lfloor r/2 \rfloor$, and any (t, k, m) such that r-$\mathsf{LRW2}(t, m) = c$, let

$$\widetilde{\mathsf{H}}_1(t, m) := x_{r'} \qquad \mathsf{H}(t) := \mathsf{H}'_{r'}(t) \oplus \mathsf{H}'_{r'+1}(t) \qquad \widetilde{\mathsf{H}}_2(t, c) := y_{r'+1},$$

and

$$\boldsymbol{\pi}_1 := \boldsymbol{\pi}'_{r'} \qquad \boldsymbol{\pi}_2 := \boldsymbol{\pi}'^{-1}_{r'+1}.$$

Clearly, the $\mathsf{LRW+}$ instance so defined is same as r-$\mathsf{LRW2}$. Furthermore, assuming $\boldsymbol{\pi}'^r \leftarrow\!\!\$\ \mathsf{Perm}(n)$ and $\mathsf{H}'^r \leftarrow\!\!\$\ \mathcal{H}^r$, KG is a PISM, $\widetilde{\mathsf{H}}_1$ and $\widetilde{\mathsf{H}}_2$ are ϵ-AUTPF, and H is ϵ-AUHF Thus, using Theorem 5, we have the following corollary on the security of cascaded $\mathsf{LRW2}$.

Corollary 4. *For $r \geq 2$, we have*

$$\mathbf{Adv}^{\mathsf{tsprp}}_{r\text{-}\mathsf{LRW2}}(q) \leq 2q^2\epsilon^{1.5} + \frac{9q^4\epsilon^2}{2^n} + \frac{32q^4\epsilon}{2^{2n}} + \frac{13q^4}{2^{3n}} + 2q^2\epsilon^2 + \frac{2q^2}{2^{2n}}.$$

In particular, for $r = 2$, assuming $\epsilon = O\left(2^{-n}\right)$, we have reproved the CCA security for 2-$\mathsf{LRW2}$ up to $2^{3n/4}$ queries.

7 Conclusion

In this paper, we gave a birthday-bound CCA distinguisher on TNT, thereby completely invalidating its beyond-the-birthday bound security claims. Further, we showed that our attack is tight by reestablishing a birthday bound security for TNT and its single-keyed variant.

In addition, we showed that by adding just one more block cipher call, the security can be amplified to $3n/4$-bit even in the CCA setting. We note that our generalization of the cascaded LRW constructions could be of independent interest.

Acknowledgments: The authors would like to thank Chun Guo for his comments on the attacks presented on TNT. Ashwin Jha carried out this work in the framework of the French-German-Center for Cybersecurity, a collaboration of CISPA and LORIA. The work done by Mridul Nandi is supported by Defence Research and Development Organisation (DRDO, India)'s cryptanalysis project.

References

1. Roberto Avanzi. The qarma block cipher family. almost mds matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency s-boxes. *IACR Transactions on Symmetric Cryptology*, pages 4–44, 2017.
2. 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*, pages 641–673, 2020.
3. Zhenzhen Bao, Chun Guo, Jian Guo, and Ling Song. Tnt: how to tweak a block cipher. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 641–673. Springer, 2020.
4. Augustin Bariant and Gaëtan Leurent. Truncated boomerang attacks and application to aes-based ciphers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 3–35. Springer, 2023.
5. 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: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II 36*, pages 123–153. Springer, 2016.
6. Srimanta Bhattacharya and Mridul Nandi. A note on the chi-square method: A tool for proving cryptographic security. *Cryptography and Communications*, 10:935–957, 2018.
7. 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.
8. 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.

9. Benoît Cogliati, Avijit Dutta, Mridul Nandi, Jacques Patarin, and Abishanka Saha. Proof of Mirror Theory for a Wide Range of $\xi _{\max }$. In *Advances in Cryptology - EUROCRYPT 2023, Proceedings, Part IV*, pages 470–501, 2023.
10. Paul Crowley. Mercy: A Fast Large Block Cipher for Disk Sector Encryption. In *Fast Software Encryption - FSE 2000, Proceedings*, pages 49–63, 2000.
11. Joan Daemen and Vincent Rijmen. Probability distributions of correlation and differentials in block ciphers. *Journal of Mathematical Cryptology*, 1(3):221–242, 2007.
12. Wei Dai, Viet Tung Hoang, and Stefano Tessaro. Information-theoretic indistinguishability via the chi-squared method. In *Annual International Cryptology Conference*, pages 497–523. Springer, 2017.
13. Nilanjan Datta, Shreya Dey, Avijit Dutta, and Sougata Mondal. Cascading Four Round LRW1 is Beyond Birthday Bound Secure. *IACR Cryptol. ePrint Arch.*, page 1242, 2023.
14. 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.
15. Chun Guo, Jian Guo, Eik List, and Ling Song. Towards Closing the Security Gap of Tweak-aNd-Tweak (TNT). In *Advances in Cryptology - ASIACRYPT 2020, Proceedings, Part I*, pages 567–597, 2020.
16. Chun Guo, Jian Guo, Eik List, and Ling Song. Towards closing the security gap of tweak-and-tweak (tnt). In *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26*, pages 567–597. Springer, 2020.
17. Zhenzhen Guo, Gaoli Wang, Orr Dunkelman, Yinxue Pan, and Shengyuan Liu. Tweakable sm4: How to tweak sm4 into tweakable block ciphers? *Journal of Information Security and Applications*, 72:103406, 2023.
18. 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.
19. 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.
20. 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.
21. Jérémy Jean, Ivica Nikolić, Thomas Peyrin, and Yannick Seurin. The deoxys aead family. *Journal of Cryptology*, 34(3):31, 2021.
22. Ashwin Jha and Mridul Nandi. Tight Security of Cascaded LRW2. *J. Cryptol.*, 33(3):1272–1317, 2020.
23. Ted Krovetz and Phillip Rogaway. The Software Performance of Authenticated-Encryption Modes. In *Fast Software Encryption - FSE 2011. Revised Selected Papers*, pages 306–327, 2011.
24. Rodolphe Lampe and Yannick Seurin. Tweakable Blockciphers with Asymptotically Optimal Security. In *Fast Software Encryption - FSE 2013, Revised Selected Papers*, pages 133–151, 2013.

25. 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.
26. Gregor Leander. Small scale variants of the block cipher present. *Cryptology ePrint Archive*, 2010.
27. Moses D. Liskov, Ronald L. Rivest, and David A. Wagner. Tweakable Block Ciphers. In *Advances in Cryptology - CRYPTO 2002, Proceedings*, pages 31–46, 2002.
28. Bart Mennink. Towards Tight Security of Cascaded LRW2. In *Theory of Cryptography - TCC 2018, Proceedings, Part II*, pages 192–222, 2018.
29. Bart Mennink and Samuel Neves. Encrypted Davies-Meyer and Its Dual: Towards Optimal Security Using Mirror Theory. In *Advances in Cryptology - CRYPTO 2017, Proceedings, Part III*, pages 556–583, 2017.
30. Kazuhiko Minematsu. Improved Security Analysis of XEX and LRW Modes. In *Selected Areas in Cryptography - SAC 2006, Revised Selected Papers*, pages 96–113, 2006.
31. Alexander Moch and Eik List. Parallelizable MACs Based on the Sum of PRPs with Security Beyond the Birthday Bound. In *Applied Cryptography and Network Security - ACNS 2019, Proceedings*, pages 131–151, 2019.
32. Luke O'Connor. On the distribution of characteristics in bijective mappings. In *Advances in Cryptology—EUROCRYPT'93: Workshop on the Theory and Application of Cryptographic Techniques Lofthus, Norway, May 23–27, 1993 Proceedings 12*, pages 360–370. Springer, 1994.
33. Jacques Patarin. The "Coefficients H" Technique. In *Selected Areas in Cryptography - SAC 2008, Revised Selected Papers*, pages 328–345, 2008.
34. Jacques Patarin. Introduction to Mirror Theory: Analysis of Systems of Linear Equalities and Linear Non Equalities for Cryptography. *IACR Cryptol. ePrint Arch.*, page 287, 2010.
35. 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.
36. Gordon Procter. A Note on the CLRW2 Tweakable Block Cipher Construction. *IACR Cryptology ePrint Archive*, 2014:111, 2014.
37. 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.
38. Rich Schroeppel and Hilarie Orman. The Hasty Pudding Cipher. AES candidate submitted to NIST, 1998.
39. Yaobin Shen, Thomas Peters, François-Xavier Standaert, Gaëtan Cassiers, and Corentin Verhamme. Triplex: an Efficient and One-Pass Leakage-Resistant Mode of Operation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(4):135–162, 2022.
40. Zhongliang Zhang, Zhen Qin, and Chun Guo. Just tweak! Asymptotically optimal security for the cascaded LRW1 tweakable blockcipher. *Des. Codes Cryptogr.*, 91(3):1035–1052, 2023.
41. Zhongliang Zhang, Zhen Qin, and Chun Guo. Just tweak! asymptotically optimal security for the cascaded lrw1 tweakable blockcipher. *Designs, Codes and Cryptography*, 91(3):1035–1052, 2023.

A On the Security Proof of BaoGGS20

The authors of [2] presented a CCA security proof of TNT that clearly contradicts our attack. Assuming our attack is correct, given it is supported by practical verification, theoretical analysis and practical estimations, the contradiction must stem from a bug in the proof. The proof follows the χ^2 method proposed by Dai *et al.* [12]. Compared to other proof methods, this method is quite recent. After carefully studying the security proof, we identified an issue that involves a fundamental, yet subtle, case analysis. The main technique of the proof, from a high level point of view, works as follows:

- A deterministic distinguisher observes the first $l-1$ queries and selects whether the next query is a forward or inverse query as well as the tweak T_l and the plaintext M_l or ciphertext C_l.
- Find the probability distribution of all the internal values of the construction given the first $l-1$ query. We call a set of possible vectors of internal values **Inter**.
- For each possible **Inter**, estimate the probability distribution of each possible response to query l.

The authors then analyze different possible cases and apply the χ^2 method on the resulting distribution.

In order to better understand the issue, we analyze our distinguisher in the flow of the security proof. The distinguisher in Algorithm 1 works as follows:

- If l is odd, it makes a forward query $(X_0, T_{l-2}+1)$.
- If l is even, it makes a backward query $(Y_{l-1}, T_{l-1} \oplus \Delta)$.

Let (S_o, U_o, V_o) are the output of π_1, input of π_2 and output of π_2 in the last (odd) query $l-1$, and we estimate the probability

$$\Pr[X_l = X_i | X_i \in \mathcal{Q}_l \texttt{ and } i \text{ is odd}].$$

Let (S_i, U_i, V_i) and (S_e, U_e, V_e) are the corresponding internal values of X_i and X_l, respectively. Then, we know that

$$V_o \oplus V_e = \Delta$$

and

$$\Pr[X_l = X_i | X_i \in \mathcal{Q}_l \texttt{ and } i \text{ is odd}] =$$

$$\Pr[S_e = S^{'} | X^{'} \in \mathcal{Q}_l \texttt{ and } i \text{ is odd}] =$$

$$\Pr[U_e \oplus T_{l-1} \oplus \Delta = U_i \oplus T_{i-1} \oplus \Delta | X^{'} \in \mathcal{Q}_l \texttt{ and } i \text{ is odd}] =$$

$$\Pr[U_e \oplus U_i = T_{l-1} \oplus T_{i-1} | X^{'} \in \mathcal{Q}_l \texttt{ and } i \text{ is odd}] =$$

Since X_0 is fixed for all odd queries, so is S_o. Thus, $U_o \oplus T_{l-1} = U_{i-1} \oplus T_{i-1}$. Therefore,

$$\Pr[U_e \oplus U_i = U_o \oplus U_{i-1} | X^{'} \in \mathcal{Q}_l \texttt{ and } i \text{ is odd}] =$$

$$\Pr[U_e \oplus U_o = U_i \oplus U_{i-1} | X^{'} \in \mathcal{Q}_l \texttt{ and } i \text{ is odd}] \approx \frac{|\mathcal{S}_{\delta,\Delta}| - 1}{2^n}$$

where $\delta = U_o \oplus U_e$. As discussed in the analysis of the distinguisher, this probability depends on the DDT of π_2 and is not the same for every permutation. Thus, it deviates from the distribution assumed in [2]. In terms of the proof presented in [2], the event we are discussing belongs to case 5 (case 1 if we swap all the forward and backward queries). In this case, the authors claim

$$\Pr[X_l = X_i | X_i \in \mathcal{Q}_l \texttt{ and } i \text{ is odd}] \leq \frac{4l}{2^{2n}} + \frac{1}{2^n - l}$$

(Equation (9) of [2]). It is easy to see that our analysis/distinguisher violates this bound. We argue that the distribution assumed for case 5/case 1 - class $\mathcal{B}$ erroneously underestimates the probability of certain bad events, and by changing the distribution to account for these bad events, the proof argumentation falls apart. Besides, it is not clear how to do so in the existing proof framework using the χ^2 method.

In particular, we look at the term $4l/2^{2n}$. The term stems from the following argument in [2]:

"It remains to bound $\Pr[\mathbf{Inter} \in \mathcal{A} | \mathcal{Q}_{l-1}]$*. For this, note that once the values in* **Inter** *except for* (S_l, W_l) *have been fixed, the number of choices for* (S_l, W_l) *is at least* $(2^n - \alpha(\mathcal{Q}_{l-1}))(2^n - \gamma(\mathcal{Q}_{l-1})) \geq 2^{2n}/4$*, where* $\alpha(\mathcal{Q}_{l-1}) \geq q \geq 2^n/2$ *and* $\gamma(\mathcal{Q}_{l-1}) \geq q \geq 2^n/2$ *are the number of distinct values in* $(S_1, \ldots S_{l-1})$ *and* $(W_1, \ldots W_{l-1})$*. Out of these* $\geq 2^{2n}/4$ *choices, the number of choices that ensure the desired property* $TNT(T_l, X_l) = Y_l$ *is at most* $l-1$*, which results from the following selection process: we first pick a pair of input-oput* (U_i, V_i) *with* $i \leq l-1$*, and then set* $S_l = T_l \oplus U_i$ *and* $W_l = T_l \oplus V_i$*. Therefore,* $\Pr[\mathbf{Inter} \in \mathcal{A} | \mathcal{Q}_{l-1}] \leq 4l/2^{2n}$*,and thus the upper bound in this case is*

$$\frac{4l}{2^{2n}} + \frac{1}{2^n - l}\text{''}.$$

Consider the first case of the 4-way multi-collision in Figure 6, which we recall in Figure 13. We note that if the triplet (δ, S_o, U_o) is known, then the collision happens with probability 1, which puts it in class $\mathcal{A}$. Then, what remains is to calculate what is the probability that the adversary can force this collision, *i.e.,*

$$\Pr[\mathbf{Inter} \in \mathcal{A} | \mathcal{Q}_{l-1}] = \Pr[U_e \oplus U_o = T_1 \oplus T_2 | \mathcal{Q}_{l-1}],$$

where T_1 and T_2 are determined by the adversary during previous queries. This means than once U_o in **Inter** is fixed (both U_o and U_e belong to a queries $i, j < l$), U_e has at most $2^n - 1 - \alpha(\mathcal{Q}_{l-1})$ choices[5], where $\alpha(\mathcal{Q}_{l-1}) \leq q \leq 2^{n-1}$ is the number of distinct values in $\{U_1, \ldots U_l\} \setminus \{U_o, U_e\}$ only 1 of them enforces the collision. In other words,

$$\Pr[\mathbf{Inter} \in \mathcal{A} | \mathcal{Q}_{l-1}] =$$

[5] We use the notation of [2] in this part.

$$
\begin{aligned}
&\Pr[U_e \oplus U_o = T_1 \oplus T_2 | \mathcal{Q}_{l-1}] \\
&\geq \frac{1}{2^n - 1 - \alpha(\mathcal{Q}_{l-1})} \\
&\geq \frac{1}{2^n - 1} \gg \frac{4l}{2^{2n}},
\end{aligned}
$$

when $l \ll q$, contradicting Equation (9) of [2].

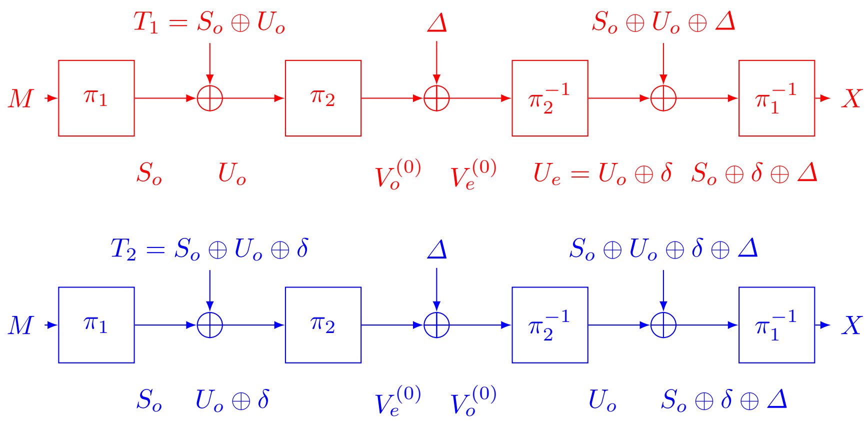

Fig. 13: A class $\mathcal{A}$ Collision.

Note that the values of V_i and W_i for $i < l$ did not affect the behaviour of the collision or the probability that **Inter** is in class $\mathcal{A}$. It seems the ambiguity may stem from applying the χ^2 method to a primitive with two dependent functions ($\tilde{E}$ and its inverse). By cascading forward and backward queries, we managed to eliminate W_i for all $1 \leq i \leq q$ and the values of W_l do not matter for the attack. Similarly, by fixing the difference between V_o and V_e to a constant $\varDelta$, we minimize the effect of their exact values on the attack.

A potential fix of this issue could be to add a tweak dependent operation after π_3, to prevent π_3 and π_3^{-1} from cancelling each other out. However, such solution may introduce new issues and is beyond out scope of study. On the other hand, we argue that fixing the proof using the exact same method is neither required nor needed, since [41] already provides a birthday bound proof and our distinguisher shows its tightness.