BBB Secure Arbitrary Length Tweak TBC from n-bit Block Ciphers

Arghya Bhattacharjee[1], Ritam Bhaumik[2,3] and Nilanjan Datta[4], Avijit Dutta[4], Shibam Ghosh[5] and Sougata Mandal[4,6]

[1] University of Luxembourg, Luxembourg
[2] EPFL, Lausanne, Switzerland
[3] Technology Innovation Institute, Abu Dhabi, United Arab Emirates
[4] Institute for Advancing Intelligence, TCG-CREST, Kolkata, India
[5] INRIA, Paris, France
[6] Ramakrishna Mission Vivekananda Educational and Research Institute, India
bhattacharjeearghya29@gmail.com,bhaumik.ritam@gmail.com,nilanjan.datta@tcgcrest.org,avirocks.dutta13@gmail.com,shibam.ghosh@inria.fr,sougatamandal2014@gmail.com

Abstract. At FSE'15, Mennink introduced the concept of designing beyond-the-birthday bound secure tweakable block cipher from an ideal block cipher. They proposed two tweakable block ciphers $\widetilde{F}[1]$ and $\widetilde{F}[2]$ that accepts n-bit tweak using a block cipher of n-bit key and n-bit data. Mennink proved that the constructions achieve security up to $2^{2n/3}$ and 2^n queries, respectively, assuming the underlying block cipher is ideal. Later, at ASIACRYPT'16, Wang et al. proposed a class of 32 new tweakable block ciphers derived from n-bit ideal block ciphers that achieve optimal security, i.e., security up to 2^n queries. The proposed designs by both Mennink and Wang et al. admit only n-bit tweaks. In FSE'23, Shen and Standaert proposed a tweakable block cipher $\widetilde{G}2$ that accepts $2n$-bit tweaks and achieves security up to 2^n queries. Their construction uses three block cipher calls, which was shown to be optimal for beyond-birthday-bound secure tweakable block ciphers accepting $2n$-bit tweaks. In this paper, we extend this line of research and consider designing tweakable block cipher supporting $3n$-bit tweaks from ideal block cipher. First, we show that there is a generic birthday-bound distinguishing attack on any such design with three block cipher calls if any of the block cipher keys are tweak-independent. We then propose a tweakable block cipher $\widetilde{\mathsf{G}}3^*$, which leverages three block cipher calls with each key being dependent on tweak. We demonstrate that $\widetilde{\mathsf{G}}3^*$ achieve security up to $2^{2n/3}$ queries. Furthermore, we extend this result and propose an optimally secure construction, dubbed $\widetilde{\mathsf{G}}3$, that uses four ideal block cipher calls with only one tweak-dependent key. Finally, we generalize this and propose an optimally secure tweakable block cipher $\widetilde{\mathsf{G}}r$ that processes rn-bit tweaks using $(r+1)$ block cipher invocations with only one tweak-dependent block cipher key. Our experimental evaluation asserts that ZMAC instantiated with $\widetilde{\mathsf{G}}3$ and $\widetilde{\mathsf{G}}4$ (i.e., $\widetilde{\mathsf{G}}r$ with $r = 4$) performs better than all the existing ideal cipher based TBC candidates.

Keywords: Tweakable Block Cipher · Ideal Cipher Model · H-Coefficient Technique · Beyond Birthday Bound · Sum Capture Lemma

1 Introduction

A block cipher is a family of permutations that is indexed by a secret key. Over time, block ciphers have gained widespread acceptance as a fundamental cryptographic object. However, their applicability is somewhat constrained due to the specific utilization of block ciphers within various modes of operation. Tweakable block cipher (TBC), as an additional fundamental cryptographic building block, serves to introduce variability within the cipher's structure. It is defined as a family of permutations $\widetilde{\mathsf{E}} : \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ indexed by secret key $k \in \mathcal{K}$ and public tweak $t \in \mathcal{T}$. The prototype of a TBC first appeared in Schroeppel's Hasty Pudding Cipher [Sch], a submission to the NIST competition for Advanced Encryption Standard [NIS00]. In this design, an additional input, called "spice", was introduced alongside the key and the plaintext in a block cipher. The purpose of this extra input was to randomize the selection of the permutation family, meaning that different values of the spice correspond to different and independent permutation families. This design concept was later formalized as a TBC by Liskov, Rivest and Wagner [LRW02, LRW11].

Tweakable block ciphers have several diverse applications, notably in designing of authenticated encryption schemes like Deoxys [JNPS21], Romulus [IKMP20], and several other candidates of NIST and CAESAR competetions [GLS+, JNPa, JNPb, Wan, HKR15, JNPS21]. Apart from designing AE schemes, TBCs have found diverse application in designing wide block encryption modes [BLN18, NI22], message authentication codes [CS08a, Nai15, CLS17, IMPS17a, Nai19, CIL+20], and hash functions [FLS+, GIK+, Hir22].

The first provably secure design of TBC was proposed by Liskov et al. [LRW02] in LRW1 and LRW2 constructions. A close contender of LRW2 construction, called XEX was proposed by Rogaway [Rog04], which was later extended by Chakraborty and Sarkar [CS06] and Minematsu [Min06] with reduced key space. The security of all these constructions are limited up to the birthday bound of the input size of the block cipher, assuming that the underlying block cipher is secure in strong pseudorandom permutation sense. A series of later works [LST12, LS13, JN20, DDDM23, JKNS23, ZQG23] have showed that cascading independent instances of LRW1 or LRW2 constructions achieves beyond the birthday bound security of the input size of the block cipher.

1.1 Tweakable Block Ciphers from Ideal Block Ciphers

In [Men15a, Men15b], Mennink initiated the study of designing tweakable block ciphers from ideal block ciphers. He demonstrated that any tweakable block cipher for n-bit tweak with a single primitive call and arbitrary linear pre- and post-processing functions cannot achieve more than birthday bound security. He then proposed the $\widetilde{F}[1]$ and $\widetilde{F}[2]$ constructions, both built from an n-bit block cipher with an n-bit key. The $\widetilde{F}[1]$ construction consists of one multiplication and a single block cipher call with a tweak-dependent key, achieving security up to $2^{n/3}$ queries. On the other hand, $\widetilde{F}[2]$ makes two block cipher calls, with one of them involving a tweak-dependent key, achieving optimal security[1] under the assumption that the block cipher behaves like an ideal cipher.

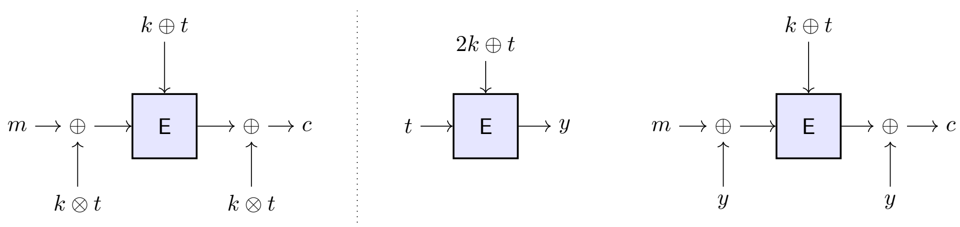


Figure 1: Construction $\widetilde{F}[1]$ (left) and $\widetilde{F}[2]$ (right) by Mennink

In [WGZ+16], Wang et al. proposed 32 additional block cipher based tweakable block

[1] A TBC with n-bit data and n-bit key is optimally secure if it is secure up to 2^n queries. Note that an exhaustive key search requires 2^n queries in this case.

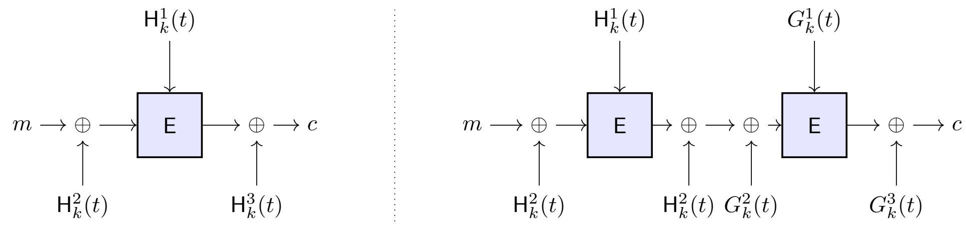


Figure 2: Construction XHX (left) and XHX2 (right)

ciphers and showed each one of them achieves optimal security based on the assumption that the underlying block cipher is an n-bit ideal cipher with n-bit key. Note that, all these constructions admit tweak of length n bits.

To incorporate variable length tweak, Jha et al. [JLM+17] proposed XHX construction that employs a block cipher and a keyed hash function to process arbitrary length tweak. XHX was proven to be secured up to $2^{(n+\kappa)/2}$ queries under the assumption that the underlying block cipher is an n-bit ideal cipher with κ-bit key. Later in [LL18], Lee et al. have extended XHX to XHX2 and proved its security up to $\min\{2^{2(n+\kappa)/3}, 2^{n+\kappa/2}\}$. Recently, Shen and Standaert [SS23] studied how to design tweakable block ciphers that admits $2n$-bit tweak from an n-bit ideal block cipher with n-bit key. It was shown that one cannot get more than $n/2$-bit security by making only two block cipher calls with $2n$-bit tweak. They have shown that by making three block cipher calls, one can realize an optimally secure tweakable block cipher ($\widetilde{G}2$) that admits $2n$ bit tweak. Moreover, they have conjectured that to construct an n-bit secure TBC with rn-bit tweaks, where $r > 2$, one may require at least $(r+1)$ block cipher calls. We pursue this line of research, aiming to design tweakable block ciphers that support large tweaks. However, before delving into this approach, let us first explore whether TBCs with large tweaks have any practical motivation.

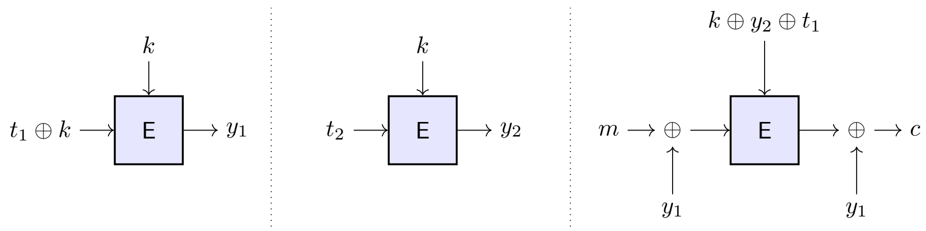


Figure 3: Construction $\widetilde{\mathsf{G2}}$ by Shen et al.

1.2 Importance of Digesting Long Tweak

Having a flexible tweak length is an interesting design goal for a TBC. Some dedicated TBCs like SKINNY [BJK+16], QARMAv2 [ABD+23] and Deoxys-TBC [JNPS21] allow $2n$-bit tweaks for n-bit blocks and n-bit keys. In fact some of their recent variants allow larger tweak lengths. For example, SKINNYe-64-256 [NSS20] and SKINNYee [NSS22] allows tweak length up to $3n$-bits and $(5n+3)$-bits, respectively. Similarly, Deoxys-TBC-512 and Deoxys-TBC-640 [CJPS22] allow tweaks of length up to $3n$-bit and $4n$-bit, respectively. In general, the tweak of a TBC can be used to contain additional information associated with a plaintext block [MI15, ABD+23]. Hence, it is desirable to make the tweak longer than the block length for more flexible designs. Recent trends show that a TBC with a large tweak is helpful as a building block for various modes of operation. For example MACs [IMPS17b] based on large tweak TBC provides higher efficiency. Similarly, authenticated encryption schemes [NSS20, NSS22, CJPS22] based on large tweak TBC provides a higher security bound. Large tweak TBCs are also useful in designing various leakage resilience almost rate-1 authenticated encryption schemes, For example, Triplex [SPS+22] uses $2n$-bit tweaks and achieves n-bit security with rate 2/3. Multiplex [PSS23] achieves n-bit security and uses dn-bit tweaks to achieve rate $d/(d+1)$. Tweplex [DDLM23] uses $dn/2$-bit tweaks to achieve birthday bound security with a rate $d/(d+1)$. Large tweak TBC is also helpful in designing various tweakable enciphering schemes [HR03, Hal04, HR04, WFW05, MM07, CS08b, Sar09, Dwo10, Sar11, BN15, DN18, CEL+21, CDK23], where a large tweak can support more modular designs.

1.3 Our Contributions

This paper focuses on designing block cipher based tweakable block cipher that supports large tweaks with a strong security guarrantee. Specifically, we aim to extend the work of Shen et al. to construct n-bit TBCs that accept n-bit key and rn-bits tweak for $r \geq 3$. Our contributions are fourfold, as outlined below:

1. We investigate the minimum number of block cipher calls required to construct a TBC that ensures beyond-birthday-bound security while processing $3n$-bit tweaks with n-bit key and n-bit data. We demonstrate that constructions using three block cipher calls cannot achieve the desired security bound unless all three block cipher keys are tweak-dependent. We prove this by exhaustively characterizing all possible TBC constructions with 0, 1, or 2 tweak-dependent keys and showing birthday-bound (or constant) attacks against each of them. Details can be found in Sect. 3.

2. Motivated by this result, we observed that it is possible to construct a beyond-birthday-bound secure TBC that accepts $3n$ bit tweaks, if all the three block cipher keys are tweak-dependent. In other words, we propose $\widetilde{\mathsf{G}}3^*$, a block cipher based TBC that processes $3n$-bit tweaks using three block cipher calls such that each of their keys are tweak dependent. We have shown in Sect. 4 that $\widetilde{\mathsf{G}}3^*$ is secured up to $2^{2n/3}$ queries in the ideal cipher model.

3. We prove that at least one tweak-dependent key is a necessary and sufficient condition for constructing an n-bit secure TBC using four block cipher calls that admits $3n$ bit tweak. We support this assertion by showing a generic birthday-bound attack in Sect. 5.1 against all possible four block cipher call constructions where all the block cipher keys are tweak-independent. We then propose $\widetilde{\mathsf{G}}3$ in Sect. 5.2 that processes $3n$-bit tweaks using four block cipher calls, with only one tweak-dependent block cipher key and prove its security up to 2^n queries in the ideal cipher model.

4. Finally, we extend the idea of $\widetilde{\mathsf{G}}3$ in Sect. 6 to yield a generic construction $\widetilde{\mathsf{G}}r$ ($r > 3$) that processes rn-bit tweaks using $(r+1)$ block cipher calls with only one tweak-dependent block cipher key and prove its security up to 2^n queries in the ideal cipher model.

We compare our proposals with the state-of-the-art tweakable block cipher schemes in Table 1. This comparison includes key size, tweak size, number of primitive calls, and respective security bounds.

Table 1: Comparison of $\widetilde{\mathsf{G}}3^*$, $\widetilde{\mathsf{G}}3$ and $\widetilde{\mathsf{G}}r$ with existing TBCs in the ideal cipher model. The tweak size refers to the size of tweak supported by the design. For XHX and XHX2, the key size is κ bits. For the remaining constructions, the key size is n bits.

Construction	#BC	#Hash	Tweak size	Security (in bits)
XHX	1	2	arbitrary	$(n+\kappa)/2$ [JLM+17]
XHX2	2	4	arbitrary	$\min\{2(n+\kappa)/3, n+\kappa/2\}$ [LL18]
$\widetilde{F}[1]$	1	0	n	$2n/3$ [Men15a]
$\widetilde{F}[2]$	2	0	n	n [Men15a]
$\widetilde{E}_1, \ldots, \widetilde{E}_{32}$	2	0	n	n [WGZ+16]
$\widetilde{G}2$	3	0	$2n$	n [SS23]
$\widetilde{\mathsf{G}}3^*$	3	0	$3n$	$2n/3$ [Sec. 4]
$\widetilde{\mathsf{G}}3$	4	0	$3n$	n [Sec. 5.2]
$\widetilde{\mathsf{G}}r$	$r+1$	0	rn	n [Sec. 6]

1.4 Performance Evaluation

This paper provides a complete characterization of the design landscape of large tweak tweakable block ciphers from ideal block ciphers without using hash functions. Along with the theoretic merit of our study, our proposals are particularly suited for TBC-based constructions that supports large tweaks. For example, our designs can be used to instantiate ZMAC, Deoxys-AE1, Deoxys-AE2, Multiplex, Tweplex that allow or require processing large tweaks. To demonstrate the efficiency of our proposed constructions, we implement them using AES-128 [DR02] and use them as primitives in the ZMAC construction. We report the number of cycles per byte required to process a given message by ZMAC. The software implementation details and benchmarking setup are presented in Section 7, and the source code is publicly available at `https://github.com/ShibamCrS/BBB_SecureLargeTweakTBC.git`. The results are presented in Table 2. A graphical presentation of the data is provided in Fig 4.
The performance result depicts that ZMAC instantiated with $\widetilde{\mathsf{G}}3$ and $\widetilde{\mathsf{G}}4$[2] performs better than XHX with polynomial-based universal hashing of $3n$-bit and $4n$-bit tweaks, respectively (see Table. 2 and Figure. 4). In addition, the performance results are also in line with the fact that we obtain performance improvement with large tweak sizes (see XHX with 16, 32, 48 and 64 B tweaks). Another important observation from the performance result of $\widetilde{\mathsf{G}}3^*$ suggests that it is important to minimize tweak-dependent keys of the underlying block ciphers even if the number of block cipher invocations is more. This is because each tweak-dependent key requires a fresh key-scheduling which is costly, particularly for long messages. As $\widetilde{F}[2]$ and $\widetilde{E}_1, \ldots, \widetilde{E}_{32}$ share the same design principle of having one tweak dependent key, we have chosen not to use later constructions in our experiment.

Remark 1. We would like to point out that despite having a lesser number of block-cipher invocation, $\widetilde{\mathsf{G}}3^*$ performs worse than $\widetilde{\mathsf{G}}3$. This is due to the use of all tweak-dependent keys that requires a fresh key-scheduling while processing each block. Thus, the significance of $\widetilde{\mathsf{G}}3^*$ from the application point of view might seem to be limited. However, the existence of $\widetilde{\mathsf{G}}3^*$ shows the possibility of designing beyond birthday bound secure TBC with $3n$-

[2]Note that $\widetilde{\mathsf{G}}4$ is the instantiation of $\widetilde{\mathsf{G}}r$ with $r = 4$.

Table 2: Software performance of ZMAC with different TBCs for various message lengths. All the TBCs used here are built from AES-128. In XHX we use PolyHash as the underlying hash function. The values are express cycles per byte (cpb).

TBC	Tweak size	1 KB	2 KB	4 KB	8 KB	16 KB	32 KB	64 KB
XHX	16B	7.60	7.01	6.72	6.56	6.51	6.48	6.46
XHX	32B	6.93	6.05	5.67	5.45	5.36	5.30	5.26
XHX	48B	6.58	5.68	5.23	5.00	4.89	4.83	4.80
XHX	64B	6.84	5.73	5.18	4.86	4.70	4.63	4.59
$\widetilde{F}[1]$	16B	6.32	5.83	5.58	5.46	5.41	5.38	5.36
$\widetilde{F}[2]$	16B	6.12	5.65	5.42	5.30	5.25	5.21	5.20
$\widetilde{G}2$	32B	4.61	4.04	3.81	3.66	3.60	3.57	3.55
$\widetilde{\mathsf{G}}3^*$	48B	10.30	8.90	8.20	7.83	7.66	7.57	7.53
$\widetilde{\mathsf{G}}3$	48B	3.72	3.23	2.99	2.87	2.81	2.77	2.75
$\widetilde{\mathsf{G}}4$	64B	3.38	2.86	2.61	2.46	2.39	2.35	2.33

bit tweak using three block cipher calls. This is in contrast to the work of Shen and Standaert [SS23] that shows an impossibility result in designing beyond birthday bound secure TBC processing n-bit and $2n$-bit tweaks using one and two block cipher calls.

2 Preliminaries

Notation: For a finite set $\mathcal{X}$, we write $X \xleftarrow{\$} \mathcal{X}$ to denote that X is uniformly sampled from $\mathcal{X}$. We write $(X_1, X_2, \ldots, X_q) \xleftarrow{\$} \mathcal{X}$ to denote that each X_i is sampled uniformly at random from $\mathcal{X}$. For a set $\mathcal{X}$, we write $\mathcal{X} \xleftarrow{\cup} X$ to denote that $\mathcal{X} \leftarrow \mathcal{X} \cup \{X\}$. For a fixed n, we write the set of all n-bit binary strings as $\{0,1\}^n$, and $\{0,1\}^*$ denote the set of all binary strings of arbitrary length. ε is used to denote the empty string. $|x|$ denotes the length of the bit string x. $\mathsf{msb}_c(Z)$ and $\mathsf{lsb}_c(Z)$ return the c most and least significant bits of a bit string Z, respectively. $x[i,j]$ denotes the substring from i-th bit to j-th bit of x. The concatenation of two strings x and y is denoted as $x\|y$. We also often write it as (x, y). For two elements $x, y \in \{0,1\}^n$, $x.y$ denotes the usual field multiplication in $\mathsf{GF}(2^n)$. We say a function $f : \{0,1\}^{dn} \to \{0,1\}^{d'n}$ is a linear if for every $x, y \in \{0,1\}^{dn}$, $f(x \oplus y) = f(x) \oplus f(y)$, and for any constant $c \in \{0,1\}^{dn}$, $f(c \cdot x) = c \cdot f(x)$, where $\cdot$ is the usual field multiplication. We say a function $g : \{0,1\}^{dn} \to \{0,1\}^{d'n}$ is affine if there is a linear function $f : \{0,1\}^{dn} \to \{0,1\}^{d'n}$ and an element $b \in \{0,1\}^{d'n}$ such that $g(x) = f(x) \oplus b$ for all $x \in \{0,1\}^{dn}$. We write $(a)_q$ to denote the number of ways q distinct objects have been chosen from a set of a elements, which is $a(a-1)(a-2)\ldots(a-q+1)$. For a natural number q, $(x_1, x_2, \ldots, x_q) \in (\{0,1\}^n)^q$ denotes a tuple of q elements, where each element is an n-bit binary string. We write $(\{0,1\}^n)^{\underline{q}} := \{(x_1, x_2, \ldots, x_q) \in (\{0,1\}^n)^q : \forall i \neq j, x_i \neq x_j\}$ to denote the set of tuples of q distinct n-bit binary strings. Thus, we have $|(\{0,1\}^n)^{\underline{q}}| = (2^n)_q$.

Block Cipher: A block cipher is a function $\mathsf{E} : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ such that for each $k \in \mathcal{K}$, $\mathsf{E}(k, \cdot)$ is a permutation over $\{0,1\}^n$. A block cipher is said to be (q, t, ϵ)-secure pseudo random permutation if for any polynomial time adversary $\mathcal{A}$ with running time at most t that makes at most q queries to either the block cipher E_k for a randomly

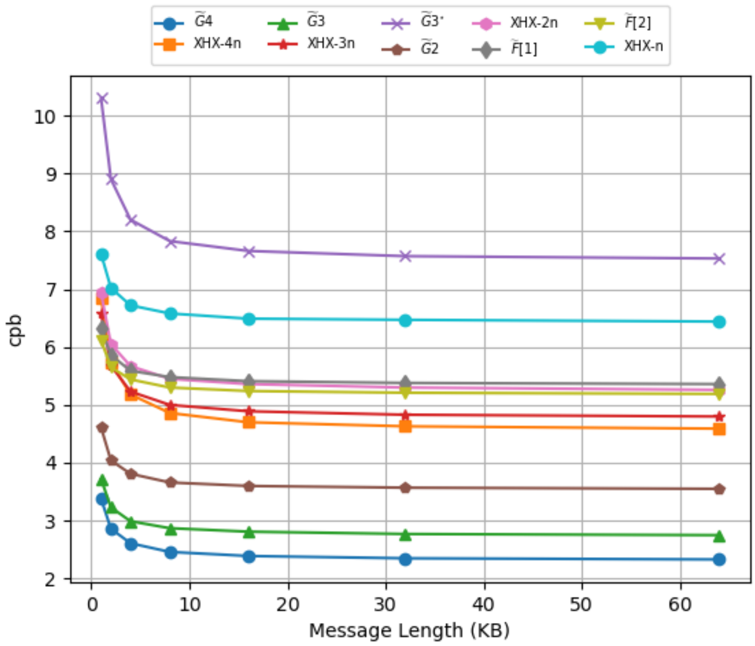


Figure 4: Software performance of ZMAC with different TBCs for various message lengths. XHX-τ denotes XHX with τ-bit tweak

chosen secret key k or an n-bit random permutation P, cannot distinguish the output distribution of the two random systems but with probability at most ϵ. Formally, we define the distinguishing advantage of the adversary $\mathcal{A}$ in distinguishing E_k from P as follows:

$$\mathbf{Adv}^{\mathrm{prp}}_{\mathsf{E}}(\mathcal{A}) := \Pr[k \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathsf{E}_k(\cdot)} = 1] - \Pr[\mathsf{P} \xleftarrow{\$} \mathsf{Perm}(n) : \mathcal{A}^{\mathsf{P}(\cdot)} = 1].$$

We call a block cipher (q, t, ϵ)-secure strong pseudo random permutation if for any polynomial time adversary $\mathcal{A}$ with running time at most t that makes at most q queries to either the block cipher E_k and its inverse E_k^{-1} for a randomly chosen secret key k or an n-bit random permutation P and its inverse P^{-1}, cannot distinguish the output distribution of the two random systems but with probability at most ϵ. In other words, we define the strong pseudo random permutation advantage of the adversary $\mathcal{A}$ in distinguishing $(\mathsf{E}_k, \mathsf{E}_k^{-1})$ from $(\mathsf{P}, \mathsf{P}^{-1})$ as follows:

$$\mathbf{Adv}^{\mathrm{sprp}}_{\mathsf{E}}(\mathcal{A}) := \Pr[k \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathsf{E}_k(\cdot), \mathsf{E}_k^{-1}(\cdot)} = 1] - \Pr[\mathsf{P} \xleftarrow{\$} \mathsf{Perm}(n) : \mathcal{A}^{\mathsf{P}(\cdot), \mathsf{P}^{-1}(\cdot)} = 1].$$

TSPRP Security in the Ideal-Cipher Model: A tweakable block cipher $\widetilde{\mathsf{E}} : \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ is a function such that for each key $k \in \mathcal{K}$ and each tweak $t \in \mathcal{T}$, $\widetilde{\mathsf{E}}(k, t, \cdot)$ is a permutation over $\{0,1\}^n$. We define the tweakable strong pseudorandom security of $\widetilde{\mathsf{E}}$ under the ideal-cipher model. We assume that $\widetilde{\mathsf{E}}$ makes internal calls to a publicly evaluated block cipher E with more than one key. Typically, $\widetilde{\mathsf{E}}$ would be keyed with some key k and derive block cipher keys $k_1, k_2, \ldots, k_m$ as a function of k and other inputs ($\widetilde{\mathsf{E}}$ can make internal calls to multiple block ciphers when all of them are independently and uniformly distributed over the set $\mathsf{BC}(\mathcal{K}, \{0,1\}^n)$). For simplicity, we write $\widetilde{\mathsf{E}}^{\mathsf{E}}_k$ to denote $\widetilde{\mathsf{E}}$ with a uniformly sampled block cipher $\mathsf{E} \xleftarrow{\$} \mathsf{BC}(\mathcal{K}, \{0,1\}^n)$, which is keyed by a randomly sampled key k from $\mathcal{K}$.

The distinguisher $\mathcal{A}$ is given access to either $(\widetilde{\mathsf{E}}^{\mathsf{E}}_k, (\widetilde{\mathsf{E}}^{-1})^{\mathsf{E}}_k, \mathsf{E}^{\pm})$ for a randomly sampled key k or $(\widetilde{\mathsf{P}}, \widetilde{\mathsf{P}}^{-1}, \mathsf{E}^{\pm})$ for $\widetilde{\mathsf{P}} \xleftarrow{\$} \mathsf{TP}(\mathcal{T}, \{0,1\}^n)$, where $\mathsf{E} \xleftarrow{\$} \mathsf{BC}(\mathcal{K}, \{0,1\}^n)$ is a uniformly sampled n-bit block cipher such that $\mathcal{A}$ can make forward or inverse queries to E, which is denoted as $\mathsf{E}^{\pm}$. We define the tsprp-advantage of $\mathcal{A}$ against the tweakable block cipher $\widetilde{\mathsf{E}}$ in the ideal cipher model as

$$\mathbf{Adv}^{\text{tsprp-icm}}_{\widetilde{\mathsf{E}}}(\mathcal{A}) := \mathbf{Adv}^{(\widetilde{\mathsf{E}}^{\mathsf{E}}_{k},(\widetilde{\mathsf{E}}^{-1})^{\mathsf{E}}_{k},\mathsf{E}^{\pm})}_{(\widetilde{\mathsf{P}},\widetilde{\mathsf{P}}^{-1},\mathsf{E}^{\pm})}(\mathcal{A}),$$

for $k \xleftarrow{\$} \mathcal{K}, \widetilde{P} \xleftarrow{\$} \mathsf{TP}(\mathcal{T},\{0,1\}^n), \mathsf{E} \xleftarrow{\$} \mathsf{BC}(\mathcal{K},\{0,1\}^n)$ and the randomness of the adversary $\mathcal{A}$. We say that $\widetilde{\mathsf{E}}$ is a (q,p,ϵ)-tsprp in the ideal cipher model if

$$\mathbf{Adv}^{\text{tsprp-icm}}_{\widetilde{\mathsf{E}}}(\mathcal{A}) \leq \epsilon,$$

for all adversaries $\mathcal{A}$ that make q queries to $\widetilde{\mathsf{E}}, \widetilde{\mathsf{E}}^{-1}$, p forward and inverse offline ideal-cipher queries to E.

2.1 H-Coefficient Technique

Let $\mathcal{A}$ be a computationally unbounded deterministic distinguisher that interacts with either the oracles in the real world, or in the ideal world. The collection of all the queries and responses that $\mathcal{A}$ made and received to and from the oracle, is called the *transcript* of $\mathcal{A}$, denoted as τ. Sometimes, we allow the oracle to release more internal information to $\mathcal{A}$ only after $\mathcal{A}$ completes all its queries and responses, but before it outputs its decision bit.

Let X_{re} and X_{id} denote the probability distributions of the transcript τ induced by the real oracle and the ideal oracle respectively. The probability of realizing a transcript τ in the ideal oracle (i.e., $\Pr[X_{\text{id}} = \tau]$) is called the *ideal interpolation probability.* Similarly, one can define the *real interpolation probability.* A transcript τ is said to be *attainable* with respect to $\mathcal{A}$ if the ideal interpolation probability is non-zero (i.e., $\Pr[X_{\text{id}} = \tau] > 0$). We denote the set of all attainable transcripts by Θ. Following these notations, we state the main theorem of H-Coefficient Technique [Pat08, CS14] as follows:

Theorem 1 (H-Coefficient Technique). *Let $\mathcal{A}$ be a fixed deterministic distinguisher that has access to either the real oracle $\mathcal{O}_{\text{re}}$ or the ideal oracle $\mathcal{O}_{\text{id}}$. Let $\Theta = \Theta_{\text{g}} \sqcup \Theta_{\text{b}}$ (disjoint union) be some partition of the set of all attainable transcripts of $\mathcal{A}$. Suppose there exists $\epsilon_{\text{ratio}} \geq 0$ such that for any $\tau \in \Theta_{\text{g}}$,*

$$\frac{\Pr[X_{\text{re}} = \tau]}{\Pr[X_{\text{id}} = \tau]} \geq 1 - \epsilon_{\text{ratio}},$$

and there exists $\epsilon_{\text{bad}} \geq 0$ such that $\Pr[X_{\text{id}} \in \Theta_{\text{b}}] \leq \epsilon_{\text{bad}}$. Then,

$$\mathbf{Adv}^{\mathcal{O}_{\text{id}}}_{\mathcal{O}_{\text{re}}}(\mathcal{A}) := |\Pr[\mathcal{A}^{\mathcal{O}_{\text{re}}} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_{\text{id}}} = 1]| \leq \epsilon_{\text{ratio}} + \epsilon_{\text{bad}}. \tag{1}$$

2.2 Sum Capture Lemma

In this section, we state a variant of the sum capture lemma [Bab02] used in [CS14]. Informally, the results states that when choosing a random subset $\mathcal{Z}$ of $\mathrm{GF}(2^n)$ (or more generally any abelian group) of size q, the value

$$\mu(\mathcal{Z}) := \max_{\mathcal{X},\mathcal{Y} \subseteq \mathrm{GF}(2^n)} |\{(z,x,y) \in \mathcal{Z} \times \mathcal{X} \times \mathcal{Y} : z = x \oplus y\}|,$$

is at most $q|\mathcal{X}||\mathcal{Y}|/2^n$, except with negligible probabilty. Chen et al. [CS14] proved the result for a different setting where $\mathcal{Z}$ arises from the interaction of an adversary with a random permutation P, namely $\mathcal{Z} = \{x \oplus y : (x,y) \in \mathcal{Q}\}$, where $\mathcal{Q}$ is the transcript of the interaction between the adversary and the permutation. We employ the similar result in our setting which is stated as follows:

Lemma 1. *Let* RF *be a random function that maps elements from* $\{0,1\}^n$ *to* $\{0,1\}^n$*. Let* $\mathcal{A}$ *be some probabilitistic distinguisher that makes* q *adaptive queries to* RF*. Let* $\mathcal{Q} = ((x_1, y_1), \ldots, (x_q, y_q))$ *denotes the transcript of the interaction with* RF *to* $\mathcal{A}$*. For any two subsets* $\mathcal{U}$ *and* $\mathcal{V}$ *of* $\{0,1\}^n$*, let*

$$\mu(\mathcal{Q}, \mathcal{U}, \mathcal{V}) = |\{((x, y), u, v) \in \mathcal{Q} \times \mathcal{U} \times \mathcal{V} : x \oplus u = y \oplus v\}|.$$

Then assuming $9n \leq q \leq 2^{n-1}$*, we have*

$$\Pr_{\mathsf{RF},\omega}\left[\exists\, \mathcal{U}, \mathcal{V} \subseteq \{0,1\}^n : \mu(\mathcal{Q}, \mathcal{U}, \mathcal{V}) \geq \frac{q|\mathcal{U}||\mathcal{V}|}{2^n} + 3\sqrt{nq|\mathcal{U}||\mathcal{V}|}\right] \leq \frac{2}{2^n}, \tag{2}$$

where the probability is taken over the random choices of RF *and the random coins* ω *of* $\mathcal{A}$*.*

2.3 Useful Combinatorial Results

In this section, we state and prove some important combinatorial results that would be required later in the security analysis of different tweakable block cipher constructions.

Lemma 2. *Let* $f = (f_1, f_2, f_3, f_4)$ *be a function, where* $f_s : \{0,1\}^{3n} \to \{0,1\}^n$*,* $\forall s \in \{1, 2, 3, 4\}$ *are affine functions. Then* f *satisfies one of the following conditions:*

1. *There exist* $t^1, t^2 \in \{0,1\}^{3n}$ *such that* $f_s(t^1) = f_s(t^2), \forall\ s \in \{1, 2, 3, 4\}$*.*
2. *There exist* $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ *satisfying* $f_1(t^i) = f_1(t^j)$*,* $f_3(t^i) \neq f_3(t^j)$*, for all distinct* $i, j \in \{1, 2, \ldots, 2^{n/2}\}$*.*
3. *There exist* $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ *satisfying* $f_1(t^i) = f_1(t^j)$*,* $f_2(t^i) \neq f_2(t^j)$*,* $f_4(t^i) \neq f_4(t^j)$*, for all distinct* $i, j \in \{1, 2, \ldots, 2^{n/2}\}$*.*

We defer the proof of the lemma in Supplementary Material A.1.

Lemma 3. *Let* γ *be an element in* $\{0,1\}^n$*. Let* $f = (f_1, f_2, f_3, f_4)$ *be a function, where* $f_s : \{0,1\}^{3n} \to \{0,1\}^n$ *for all* $s \in \{1, 2, 3, 4\}$ *are affine functions. Then* f *satisfies at least one of the following conditions:*

1. *There exist* $t^1, t^2 \in \{0,1\}^{3n}$ *such that* $f_s(t^1) = f_s(t^2)$ *for all* $s \in \{1, 2, 3, 4\}$*.*
2. *There exist* $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ *satisfying* $f_2(t^i) \neq f_2(t^j)$*,* $f_3(t^i) = f_3(t^j)$*,* $f_4(t^i) = f_4(t^j)$*, for all distinct* $i, j \in \{1, 2, \ldots, 2^{n/2}\}$*.*
3. *There exist* $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ *satisfying* $f_1(t^i) \neq f_1(t^j)$*,* $f_3(t^i) \neq f_3(t^j)$*,* $f_4(t^i) = \gamma.f_3(t^i)$*, for all distinct* $i, j \in \{1, 2, \ldots, 2^{n/2}\}$*.*
4. *There exist* $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ *satisfying* $f_2(t^i) \neq f_2(t^j)$*,* $f_4(t^i) = \gamma.f_3(t^i)$*, for all distinct* $i, j \in \{1, 2, \ldots, 2^{n/2}\}$*.*

We defer the proof of the lemma in Supplementary Material A.2.

Lemma 4. *Let* $f = (f_1, f_2, f_3, f_4)$ *be a function, where* $f_s : \{0,1\}^{3n} \to \{0,1\}^n$ *for all* $s \in \{1, 2, 3, 4\}$ *are affine functions. Then* f *satisfies at least one of the following conditions:*

1. *There exist* $t^1, t^2 \in \{0,1\}^{3n}$ *such that* $f_s(t^1) = f_s(t^2)$ *for all* $s \in \{1,2,3,4\}$.
2. *There exist* $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ *satisfying* $f_2(t^i) \neq f_2(t^j)$, $f_4(t^i) = f_4(t^j)$, *for all distinct* $i, j \in \{1, 2, \ldots, 2^{n/2}\}$.
3. *There exist* $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ *satisfying* $f_1(t^i) \neq f_1(t^j)$, $f_3(t^i) \neq f_3(t^j)$, $f_4(t^i) = f_4(t^j)$, *for all distinct* $i, j \in \{1, 2, \ldots, 2^{n/2}\}$.

We defer the proof of the lemma in Supplementary Material A.3.

3 Generic Birthday Attacks on TBCs with $3n$-bit Tweak from Three BCs with Any Tweak-independent Key

In this section, we demonstrate that constructing tweakable block ciphers with $3n$-bit tweaks that are secure beyond the birthday bound using three block ciphers is impossible unless all the block ciphers have tweak-dependent keys. To support our claim, we first illustrate birthday-bound attacks on the generic construction where all three block ciphers use tweak-independent keys. Subsequently, we extend the idea to mount birthday attacks in cases where one or two block ciphers use tweak-independent keys. Note that our search space considers constructions with the following simplified assumptions: (i) the message is fed only at the input of the last block cipher call, (ii) no tweak is fed into the input or the output of the last block cipher call. We will justify the choice of this search space at the end of the subsection.

3.1 Constructions with Three Tweak-independent Keys

In this subsection, we consider TBC constructions with three block ciphers, in which we have all the block cipher calls with tweak-independent keys. The generalized construction for this case, dubbed $\mathcal{C}_1$, is depicted in Fig. 16. Here we assume $f_1,\ f_2 : \{0,1\}^{3n} \to \{0,1\}^n$ are any affine functions and $a_1, a_2, a_3, \alpha_1, \alpha_2, \alpha_3.\beta_1, \beta_2$ are elements from $\{0,1\}^n$. Note that incorporating tweaks into the message does not amplify security. So, we refrain from using such modifications in our constructions. Now, to attack this generic construction,

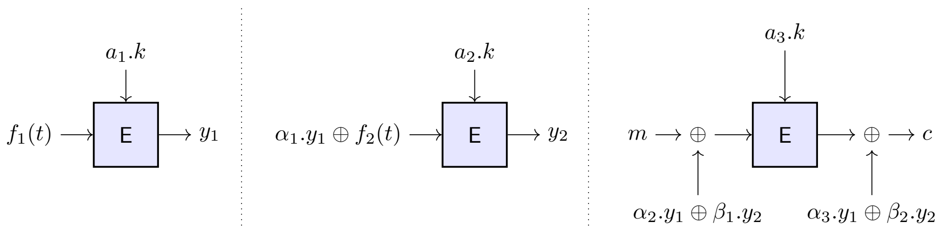


Figure 5: Construction $\mathcal{C}_1$: All the block ciphers use tweak-independent key

our strategy is as follows:

1. Find two tweaks such that t^1, t^2 such that $f_1(t^1) = f_1(t^2)$, $f_2(t^1) = f_2(t^2)$. Note that, with this choice of tweaks, we will have $y_1^1 = y_1^2$ as well as $y_2^1 = y_2^2$.
2. We can use the above observation to distinguish the TBC from a random tweakable permutation by making two oracle queries (m, t^1), (m, t^2), and verifying if the corresponding outputs match. Note that, for the real construction, this matches

with probability 1, while for random tweakable permutations, the probability is only $1/2^n$.

An algorithmic description of the attack is presented in Fig. 18 of the Supplementary Material C.1.

3.2 Constructions with Two Tweak-independent Keys

In this subsection, we consider all the possible TBC constructions with three block ciphers where we have two block cipher calls with tweak-independent keys. By tweak-independent keys, we mean keys are derived only from the master secret key. Use of such keys are efficient as one do not need separate sub-key generation functions to process those block cipher calls. It is straightforward to see that there are three possible cases depending on which of the block cipher invocations uses the tweak-dependent key.

Case 1: First block cipher uses the tweak-dependent key. Here we look at all the possible constructions where the first block cipher uses the tweak-dependent key and the next two block cipher uses tweak-independent keys. The generalized construction, dubbed $\mathcal{C}_2$, is depicted in Fig. 6. Now, to attack this generic construction, let us consider the

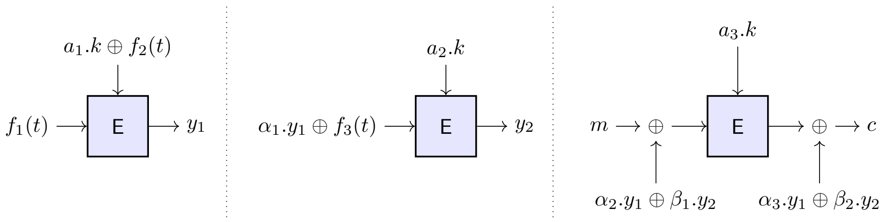


Figure 6: Construction $\mathcal{C}_2$: Only first block cipher uses tweak-dependent key

function $f : \{0,1\}^{3n} \to \{0,1\}^{3n}$ defined as $f(t) := f_1(t)\|f_2(t)\|f_3(t)$ is injective. Otherwise, we can find two tweaks t^1 and t^2 for which $f(t^1) = f(t^2)$. Now if we encrypt (m, t^1) and (m, t^2), we have $y_1^1 = y_1^2$, and $y_2^1 = y_2^2$, which ensures the obtained ciphertexts would be same for the real construction. Thus, we can mount an attack with a constant number of queries. Now we consider the case when f is injective, and in this case, our strategy is as follows:

1. Find $2^{n/2}$ tweaks such that for each pair of tweaks (t^i, t^j), we have $f_2(t^i) \neq f_2(t^j)$, and $f_3(t^i) = f_3(t^j)$. The injectivity of the function f ensures that we will have such tweaks. Now look at the y_1 values - since the keys used in the block cipher for generating these values are distinct, and we have $2^{n/2}$ keys, at least two of them collide, i.e., there exists i, j such that $y_1^i = y_1^j$.

2. Now, let's examine the y_2 values. Given that the same keys are utilized in the block cipher to generate these values, and there exist indices i and j such that $y_1^i = y_1^j$ and $f_3(t^i) = f_3(t^j)$, it follows that $y_2^i = y_2^j$. Now the question is how to detect such a collision. Observe that, in such a case the ciphertext c_i generated for (m, t^i) would be equal to c_j, the ciphertext generated for (m, t^j).

3. Finally, we can distinguish the TBC from a random tweakable permutation by making two additional oracle queries $(m^\star, t^i)$, $(m^\star, t^j)$, where $m^\star \neq m$, and verifying if the corresponding outputs match. Note that, for the real construction, this matches with probability 1, while for random tweakable permutations, the probability is only $1/2^n$.

An algorithmic description of the attack is shown in Fig. 19 (See Supplementary Material C.2).

Case 2: Second block cipher uses the tweak-dependent key. Here we look at all the possible constructions where the second block cipher uses the tweak-dependent key and the other two block cipher uses tweak-independent keys. The generalized construction, dubbed $\mathcal{C}_3$, is depicted in Fig. 7.

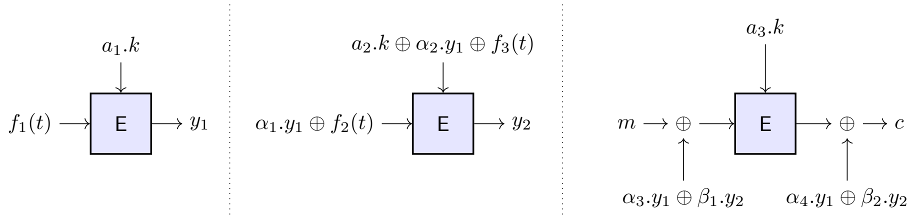


Figure 7: Construction $\mathcal{C}_3$: Only second block cipher uses tweak-independent keys

It is easy to see that a similar birthday attack with $2^{n/2}$ tweaks satisfying $f_1(t^i) = f_1(t^j)$ and $f_3(t^i) \neq f_3(t^j)$, for each (t^i, t^j) pairs, following an adversary as given in Fig.20, Supplementary Material C.3, will hold in this case.

Case 3: Final block cipher uses the tweak-dependent key. Here we look at all the possible constructions where the final block cipher uses the tweak-dependent key and the first two block cipher uses tweak-independent keys. The generalized construction, dubbed $\mathcal{C}_4$, is depicted in Fig. 8.

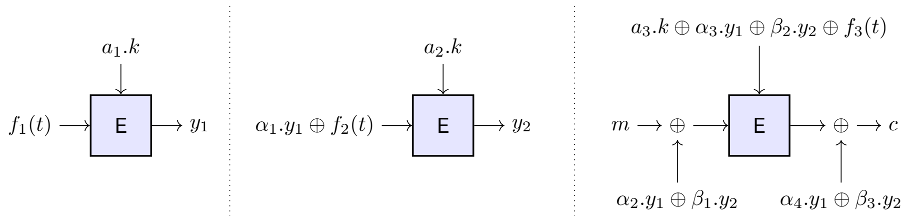


Figure 8: Construction $\mathcal{C}_4$: Only the final block cipher uses tweak-independent key

Now, to mount an attack on this generic construction, we consider the following sub-cases depending on the values of β_1 and β_2:

Sub-case 3.1: $\beta_1 = \beta_2 = 0$. In this case, we mount the birthday attack as follows:

1. Find tweaks t^1 and t^2 such that $f_1(t^1) = f_1(t^2)$, $f_2(t^1) \neq f_2(t^2)$, and $f_3(t^1) = f_3(t^2)$.
2. Make two queries (m, t^1) and (m, t^2). Note that the condition $\beta_1 = \beta_2 = 0$ ensures that there is a (key, input) collision occurs in the final block cipher for both queries. Let us assume that the corresponding ciphertexts are c_1 and c_2. It is easy to see that we have $y_2^1 \oplus y_2^2 = c_1 \oplus c_2$.
3. Finally, we can distinguish the real construction from a random tweakable permutation by making two additional oracle queries $(m \oplus \Delta, t^i)$, $(m \oplus \Delta, t^j)$, where $\Delta \neq 0$, and verifying that the corresponding ciphertexts, say $c_i^\star$ and $c_j^\star$, satisfy the equation $c_1^\star \oplus c_2^\star = c_1 \oplus c_2$.

Sub-case 3.2: $\beta_1 \neq 0,\ \beta_2 = 0$**.** Here we mount the attack as follows:

1. Find $2^{n/2}$ tweaks such that for each pair of tweaks (t^i, t^j), we have $f_1(t^i) = f_1(t^j)$, $f_2(t^i) \neq f_2(t^j)$, and $f_3(t^i) = f_3(t^j)$. Again, the injectivity of the function $f = (f_1, f_2, f_3)$ ensures that we will have such tweaks. Note that, with this choice of tweaks, we will have $y_1^i = y_1^j$, for all (i, j).

2. Now we make $2^{n/2}$ queries in the form (m_i, t^i), such that for all i, j, $m_i \neq m_j$. Note that our choice of messages ensures that a (key, input) collision occurs in the final block cipher if $y_2^i \oplus y_2^j = \beta_1^{-1}(m_i \oplus m_j)$. It is easy to see that by birthday paradox, we expect that at least one such pair, say $((m_i, t^i), (m_j, t^j))$ exists, and in that case, we have $c_i \oplus c_j = \beta_1^{-1}\beta_3(m_i \oplus m_j)$.

3. Finally, we can distinguish the real construction from a random tweakable permutation by making two additional oracle queries $(m_i \oplus \Delta, t^i)$, $(m_j \oplus \Delta, t^j)$, where $\Delta \neq 0$, and verifying that the corresponding ciphertexts, say $c_i^\star$ and $c_j^\star$, satisfy the equation $c_i^\star \oplus c_j^\star = \beta_1^{-1}\beta_3(m_i \oplus m_j)$.

Sub-case 3.3: $\beta_2 \neq 0$**.** Here we proceed as follows:

1. Find $2^{n/2}$ tweaks such that for each pair of tweaks (t^i, t^j), we have $f_1(t^i) = f_1(t^j)$, $f_2(t^i) \neq f_2(t^j)$, and $f_3(t^i) \neq f_3(t^j)$.

2. Now we make $2^{n/2}$ queries in the form $(m_i := \beta_2^{-1}\beta_1 f_3(t^i),\ t^i)$. Note that our choice of messages ensures that a (key, input) collision occurs in the final block cipher if $\beta_1(y_2^i \oplus y_2^j) = m_i \oplus m_j$. Now by birthday paradox, we expect that at least one such pair, say $((m_i, t^i), (m_j, t^j))$ exists. In that case we have $c_i \oplus c_j = \beta_3\beta_2^{-1}(f_3(t^i) \oplus f_3(t^j))$.

3. Finally, we can distinguish the real construction from a random tweakable permutation by making two additional oracle queries $(m_i \oplus \Delta, t^i)$, $(m_j \oplus \Delta, t^j)$, where $\Delta \neq 0$, and verifying that the corresponding ciphertexts, say $c_i^\star$ and $c_j^\star$, satisfy the equation $c_i^\star \oplus c_j^\star = c_i \oplus c_j$.

An algorithmic description of the attack corresponding to the three sub-cases are presented in Fig. 20 of the Supplementary Material C.4.

3.3 Constructions with One Tweak-independent Key

In this subsection, we consider all the possible TBC constructions with three block ciphers where we have a single block cipher call with tweak-independent key. It is straightforward to see that there are three possible cases depending on which of the block cipher invocations uses the tweak-dependent key.

Case 1: First block cipher uses the tweak-independent key. Here we look at all the possible constructions where the first block cipher uses the tweak-independent key and the next two block cipher uses tweak-dependent keys. The generalized construction, dubbed $\mathcal{C}_5$, is depicted in Fig. 9.
Sub-case 1.1: $\beta_1 = \beta_2 = 0$**.** In this case, we mount the constant query attack as follows:

1. Find 2 tweaks t^1, t^2 such that $f_1(t^1) = f_1(t^2)$ and $f_4(t^1) = f_4(t^2)$. Note that, with this choice of tweaks, we will have $y_1^1 = y_1^2$.

2. Now we make 2 queries (m, t^1) and (m, t^2), for some message m. We will have the same (input, key) pair of the final block cipher as the messages and the first block cipher outputs are the same for both queries. Observe corresponding responses say c_1 and c_2.

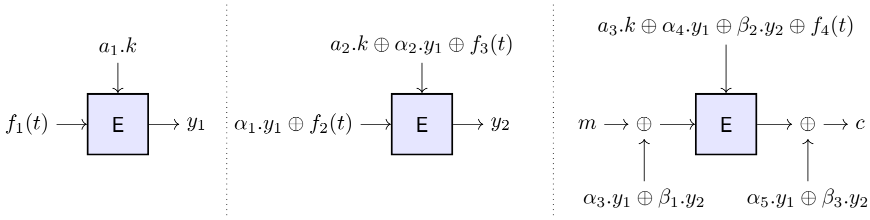


Figure 9: Construction $\mathcal{C}_5$: Only the first block cipher uses tweak-independent key

3. Finally, we can distinguish the TBC from a random tweakable permutation by making two additional oracle queries $(m \oplus \Delta, t^1)$, $(m \oplus \Delta, t^2)$, where $\Delta \neq 0$ and observing if the corresponding ciphertexts, say $c_1^\star$ and $c_2^\star$ satisfies the equation $c_1^\star \oplus c_2^\star = c_1 \oplus c_2$. Note that, for our TBC construction, this equation matches with probability 1, while for random tweakable permutations, the probability is only $1/2^n$.

Sub-case 1.2: $\beta_1 \neq 0$, $\beta_2 = 0$**.** In this case, we will be able to find $2^{n/2}$ tweaks $t^1, \ldots, t^{2^{n/2}}$ such that for each pair of tweaks (t^i, t^j), $f_1(t^i) = f_1(t^j)$, $f_4(t^i) = f_4(t^j)$, and either $f_3(t^i) \neq f_3(t^j)$ or $f_2(t^i) \neq f_2(t^j)$. Based on this observation, our approach is described as follows:

1. Find $2^{n/2}$ tweaks $t^1, \ldots, t^{2^{n/2}}$ such that for each pair of tweaks (t^i, t^j), we have at least of above two condition. Note that, with this choice of tweaks, we will have $y_1^i = y_1^j$, for all (i, j).
2. Now we make $2^{n/2}$ queries (m_i, t^i), where $m_i \neq m_j$ for each pair (i, j). We expect at least one collision in the (input, key) pair of the final block cipher as such a collision occurs when $(y_2^i \oplus y_2^j) = \beta_1^{-1}(m_i \oplus m_j)$. This collision is observable through the equation $\beta_1(c_i \oplus c_j) = \beta_3(m_i \oplus m_j)$.
3. Finally, we can distinguish the real construction from a random tweakable permutation by making two additional oracle queries $(m_i \oplus \Delta, t^i)$, $(m_j \oplus \Delta, t^j)$, where $\Delta \neq 0$ and observing if the corresponding ciphertexts, say $c_i^\star$ and $c_j^\star$ satisfies the equation $c_i^\star \oplus c_j^\star = c_i \oplus c_j$.

Sub-case 1.3: $\beta_2 \neq 0$**.** If we can find $2^{n/2}$ tweaks $t^1, \ldots, t^{2^{n/2}}$ such that for each pair of tweaks (t^i, t^j), we have $f_1(t^i) = f_1(t^j)$, $f_3(t^i) \neq f_3(t^j)$, then we use the following strategy.

1. Find $2^{n/2}$ tweaks $t^1, \ldots, t^{2^{n/2}}$ such that for each pair of tweaks (t^i, t^j), we have $f_1(t^i) = f_1(t^j)$, $f_3(t^i) \neq f_3(t^j)$. Note that, with this choice of tweaks, we will have $y_1^i = y_1^j$, for all (i, j).
2. Now we make $2^{n/2}$ queries $(m_i = \beta_2^{-1}\beta_1(f_4(t^i)), t^i)$. We expect at least one collision in the (input, key) pair of the final block cipher as such a collision occurs when $(y_2^i \oplus y_2^j) = \beta_2^{-1}(f_4(t^i) \oplus f_4(t^j))$. This collision is observable through the equation $(c_i + c_j) = \beta_3\beta_2^{-1}(f_4(t^i) + f_4(t^j))$.
3. Finally, we can distinguish the TBC from a random tweakable permutation by making two additional oracle queries $(m_i \oplus \Delta, t^i)$, $(m_j \oplus \Delta, t^j)$, where $\Delta \neq 0$ and observing if the corresponding ciphertexts, say $c_i^\star$ and $c_j^\star$ satisfies the equation $c_i^\star \oplus c_j^\star = c_i \oplus c_j$.

Otherwise, by virtue of Lemma 2, we can find $2^{n/2}$ tweaks $t^1, \ldots, t^{2^{n/2}}$ such that for each pair of tweaks (t^i, t^j), we have $f_1(t^i) = f_1(t^j)$, $f_2(t^i) \neq f_2(t^j)$, and $f_4(t^i) \neq f_4(t^j)$. In this case, our approach is described as follows:

1. Find $2^{n/2}$ tweaks $t^1, \dots, t^{2^{n/2}}$ such that for each pair of tweaks (t^i, t^j), we have $f_1(t^i) = f_1(t^j)$, $f_2(t^i) \neq f_2(t^j)$, and $f_4(t^i) \neq f_4(t^j)$. Note that, with this choice of tweaks, we will have $y_1^i = y_1^j$, for all (i, j).

2. Now we make $2^{n/2}$ queries $(m_i = \beta_2^{-1}\beta_1(f_4(t^i)), t^i)$. We expect at least one collision in the (input, key) pair of the final block cipher as such a collision occurs when $(y_2^i \oplus y_2^j) = \beta_2^{-1}(f_4(t^i) \oplus f_4(t^j))$. This collision is observable through the equation $\beta_1(c_i \oplus c_j) = \beta_3(m_i \oplus m_j)$.

3. Finally, we can distinguish this construction from a random tweakable permutation by making two additional oracle queries $(m_i \oplus \Delta, t^i)$, $(m_j \oplus \Delta, t^j)$, where $\Delta \neq 0$ and observing if the corresponding ciphertexts, say $c_i^\star$ and $c_j^\star$ satisfies the equation $c_i^\star \oplus c_j^\star = c_i \oplus c_j$. Note that, for our TBC construction, this equation matches with probability 1, while for random tweakable permutations, the probability is only $1/2^n$.

An algorithmic description of the attack corresponding to the three sub-cases are presented in Fig. 21 of the Supplementary Material C.5.

Case 2: Second block cipher uses the tweak-independent key. We will look at all the possible constructions where the second block cipher uses the tweak-independent key and the other two block cipher uses tweak-dependent keys. The generalized construction, dubbed $\mathcal{C}_6$, is depicted in Fig. 10. Note that if $\alpha_1 = 0$, then we can easily mount a birthday attack using similar technique as used in the previous case. Hence, we concentrate only on the constructions with $\alpha_1 \neq 0$. Now, to make an attack on the generic construction, we

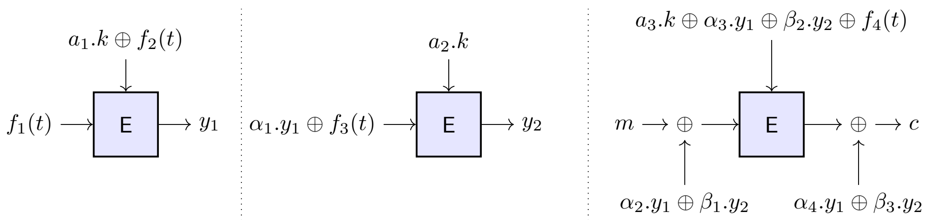


Figure 10: Construction $\mathcal{C}_6$: Only the second block cipher uses tweak-independent key

first make the following observation: If we can find two tweaks $t^1, t^2 \in \{0,1\}^{3n}$ satisfying $f_s(t^1) = f_s(t^2)$. $\forall\ s \in \{1,2,3,4\}$, then we are done. In this case, we use the following strategy:

1. Find $t^1, t^2 \in \{0,1\}^{3n}$ satisfying $f_s(t^1) = f_s(t^2)$. $\forall\ s \in \{1,2,3,4\}$. This ensures $y_1^1 = y_1^2$ and $y_2^1 = y_2^2$.

2. Make two query (m, t^1) and (m, t^2) for any message m, and observe if the respective cipher texts c_1 and c_2 are equal (TBC), or not (random tweakable permutation).

If we do not have such t^1, t^2, then we use following attack strategy:

Sub-case 2.1 $\alpha_3 = 0$. By virtue of lemma 4, we will have $2^{n/2}$ tweaks $t^1, t^2, \dots, t^{2^{n/2}}$ satisfying either (C1) $f_2(t^i) \neq f_2(t^j) \wedge f_4(t^i) = f_4(t^j)$, or (C2) $f_1(t^i) \neq f_1(t^j) \wedge f_3(t^i) \neq f_3(t^j) \wedge f_4(t^i) = f_4(t^j)$. For both cases, we use the following strategy:

1. Find $2^{n/2}$ tweaks $t^1, t^2, \dots, t^{2^{n/2}}$ satisfying at least one of conditions (C1) or (C2).

2. Now we make $2^{n/2}$ queries $(m_i = \alpha_2\alpha_1^{-1}(f_3(t^i)), t^i)$. We expect at least one collision in the (input, key) pair of the final block cipher as such a collision occurs when $(y_1^i \oplus y_1^j) = \alpha_1^{-1}(f_3(t^i) \oplus f_3(t^j))$. This collision is observable through the equation $\alpha_1(c_i \oplus c_j) = \alpha_4(f_3(t^i) \oplus f_3(t^j))$.

3. Finally, we can distinguish the TBC from a random tweakable permutation by making two additional queries $(m_i \oplus \Delta, t^i)$, $(m_j \oplus \Delta, t^j)$, where $\Delta \neq 0$ and observing if the corresponding ciphertexts, say c_i^*, c_j^* satisfies $c_i^* \oplus c_j^* = c_i \oplus c_j$.

Sub-case 2.2 $\alpha_3 \neq 0$. In this case, we apply Lemma 3 and deduce that we can find $2^{n/2}$ tweaks $t^1, t^2, \ldots, t^{2^{n/2}}$ satisfying either (C1) $f_2(t^i) \neq f_2(t^j),\ f_3(t^i) = f_3(t^j),\ f_4(t^i) = f_4(t^j)$, or (C2) $f_1(t^i) \neq f_1(t^j),\ f_3(t^i) \neq f_3(t^j),\ f_4(t^i) = \alpha_3\alpha_1^{-1}f_3(t^i)$, or (C3) $f_2(t^i) \neq f_2(t^j),\ f_4(t^i) = \alpha_3\alpha_1^{-1}f_3(t^i)$. For all the three cases, we use the following strategy:

1. Find $2^{n/2}$ tweaks $t^1, t^2, \ldots, t^{2^{n/2}}$ satisfying at least one of conditions (C1), (C2) and (C3).

2. Now we make $2^{n/2}$ queries $(m_i = \alpha_2\alpha_1^{-1}(f_3(t^i)), t^i)$. We expect at least one collision in the (input, key) pair of the final block cipher as such a collision occurs when $(y_1^i \oplus y_1^j) = \alpha_1^{-1}(f_3(t^i) \oplus f_3(t^j))$. This collision is observable through the equation $\alpha_1(c_i \oplus c_j) = \alpha_4(f_3(t^i) \oplus f_3(t^j))$.

3. Finally, we can distinguish the TBC from a random tweakable permutation by making two additional queries $(m_i \oplus \Delta, t^i)$, $(m_j \oplus \Delta, t^j)$, where $\Delta \neq 0$ and observing if the corresponding ciphertexts, say c_i^*, c_j^* satisfies $c_i^* \oplus c_j^* = c_i \oplus c_j$.

An algorithmic description of the attack corresponding to the two sub-cases are shown in Fig. 22 of the Supplementary Material C.6.

Case 3: Final block cipher uses the tweak-independent key. We will look at all the possible constructions where the final block cipher uses the tweak-independent key and the other two block cipher uses tweak-dependent keys. The generalized construction, dubbed $\mathcal{C}_7$, is depicted in Fig. 11. Now, if there exists $t^1, t^2 \in \{0,1\}^{3n}$ satis-

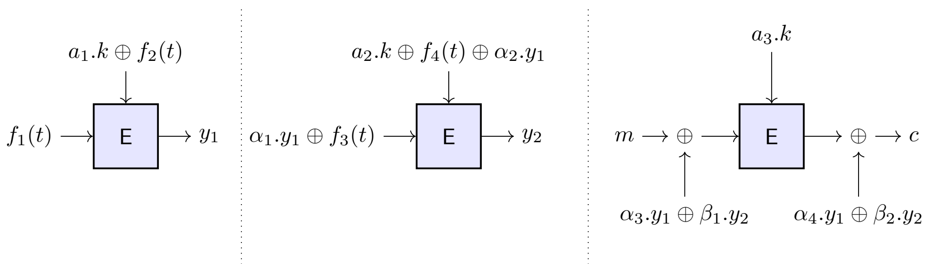


Figure 11: Construction $\mathcal{C}_7$: Only the final block cipher uses tweak-independent key

fying $f_s(t^1) = f_s(t^2),\ \forall\ s \in \{1,2,3,4\}$, we simply use the two-query distinguisher as used in the previous case. Otherwise, there exist $2^{n/2}$ many tweaks say $t^1, t^2, \ldots, t^{2^{n/2}}$ satisfying either (C1) $(f_1(t^i) = f_1(t^j)) \wedge (f_2(t^i) = f_2(t^j)) \wedge (f_3(t^i) \neq f_3(t^j)$, or (C2) $(f_1(t^i) = f_1(t^j)) \wedge (f_2(t^i) = f_2(t^j)) \wedge (f_4(t^i) \neq f_4(t^j)))$, for all (i,j) pair. We use this fact to mount the attack as follows.

Sub-case 3.1: $\beta_1 \neq 0$. In this case we use attack strategy as follows:

1. Find $2^{n/2}$ tweaks satisfying either (C1) or (C2).

2. Make $2^{n/2}$ queries (m_i, t^i), where $m_i \neq m_j$ for all (i, j) pair and observe cipher text c_i's. We expect at least one collision in (input, key) pair of the final block cipher as it occurs if $y_2^i \oplus y_2^j = \beta_1^{-1}(m_i \oplus m_j)$. This is observable by the equation $\beta_1(c_i \oplus c_j) = \beta_2(m_i \oplus m_j)$.

3. Finally, make additional two queries $(m_i \oplus \Delta, t^i)$, $(m_j \oplus \Delta, t^j)$ and get corresponding cipher text c_i^*, c_j^*. Distinguish by observing whether $c_i^* \oplus c_j^* = c_i \oplus c_j$ (real TBC construction), or not (random tweakable permutation).

Sub-case 3.2: $\beta_1 = 0$. We will proceed as follows:

1. Find two tweak t^1, t^2 such that $f_1(t^1) = f_1(t^2) \wedge f_2(t^1) = f_2(t^2)$. It is easy to see that we have $y_1^1 = y_1^2$.

2. Make two queries (m, t^1) and (m, t^2). Note that if $c_1,\ c_2$ be two corresponding cipher text, then $c_1 \oplus c_2 = \beta_2(y_2^1 \oplus y_2^2)$.

3. Finally, make two additional queries $(m \oplus \Delta, t^1)$ and $(m \oplus \Delta, t^2)$ and observe if the corresponding cipher texts c_1^* and c_2^* satisfy the equation $c_1^* \oplus c_2^* = c_1 \oplus c_2$ (for real TBC construction), or not (random tweakable permutation).

The concrete attacks correspond to these two subcases are formally presented in Fig. 23, Supplementary Material C.7.

3.4 Justification of the Search Space

We have already mentioned that our search space considers constructions with assumptions that the message is fed only at the input of the last block cipher call, and no tweak is fed into the input or the output of the last block cipher call. Here we briefly justify our assumptions below:

- *Case 1: Message is fed into keys:* Here the construction won't be invertible, as finding the keys of a block cipher given its (input, output) pairs is not possible.

- *Case 2: Message is fed into several block ciphers:* Suppose the message is fed into the second and the final block cipher. For the invertibility of the construction, the final block cipher must be independent of the output of the second block cipher. This can be exploited to mount a simple PRP attack. A similar argument can be made for all possible combinations.

- *Case 3: A linear combination of the message, tweaks and the block cipher outputs is used to define the ciphertext*: One can easily mount a two-query PRP attack, exploiting the property that under the same tweak, a linear combination of two ciphertexts can be written as a linear combination of the corresponding two plaintexts.

- *Case 4: Tweak is fed into the input/output of the final block cipher:* This does not strengthen the security, and similar attacks will go through. In fact, since the tweaks are controlled by the adversary, this may weaken the security.

- *Case 5: Message is fed into one of the non-final block-ciphers:* There are two cases: the message in XORed before the first block cipher call or the second block cipher call. For each of them, we have several cases when all the keys are not tweak-dependent, and for each of the cases, we show a birthday or constant-time attacks using similar ideas as used in Sect. 3.1-3.3 when the message block is used in the final block-cipher. A detailed analysis of this presented in Supplementary Material B.

4 BBB Secure TBC with $3n$-bit Tweaks Using Three Block Cipher Calls

Let $\mathsf{E} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a n-bit block cipher. The tweakable block cipher $\widetilde{\mathsf{G}}3^* : \{0,1\}^n \times \{0,1\}^{3n} \times \{0,1\}^n \to \{0,1\}^n$ with a $3n$ bit tweak with only three block cipher invocation is constructed as follows: two block cipher calls are first invoked sequentially to produce two masks y_1 and y_2 from the tweaks t_1, t_2, t_3 and the master key k. By using y_1 to mask both the input and output, and using y_2, the master key k, and t_1 to provide variety in the sub-key, a third block cipher call is then invoked to encrypt the message m to the ciphertext c. A pictorial illustration of the construction $\widetilde{\mathsf{G}}3^*$ is given in Fig. 12.

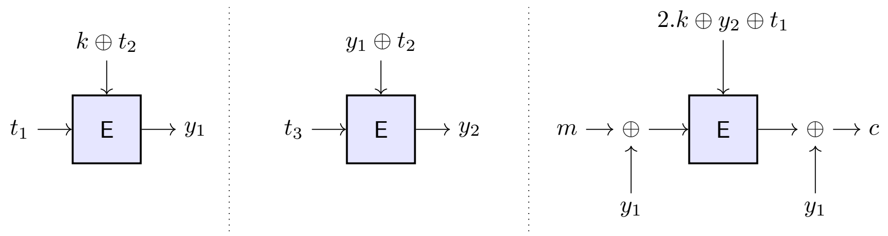


Figure 12: $\widetilde{\mathsf{G}_3^*}$ construction: TBC with $3n$ bit tweaks using three block cipher calls. We impose a natural ordering on the block cipher calls from left to right.

In the following, we demonstrate that $\widetilde{\mathsf{G}}3^*$ is a secure tweakable block cipher with $3n$ bit tweaks against all adversaries that makes roughly $2^{2n/3}$ construction and ideal-cipher queries. Formally, we present the following result:

Theorem 2. *Let $\mathcal{A}$ be an adversary making at most q construction queries and p ideal-cipher queries including both forward and backward queries. Then,*

$$\mathbf{Adv}^{\text{tsprp-icm}}_{\widetilde{\mathsf{G}}3^*}(\mathcal{A}) \leq \frac{q}{2^n} + \frac{12q^2}{2^{2n}} + \frac{6q^2p}{2^{2n}} + \frac{4qp}{2^{2n}} + \frac{11qp^2}{2^{2n}}.$$

Proof. Let us assume that $\mathcal{A}$ makes at most q construction queries (to the first oracle) and p ideal-cipher queries (to the second oracle). Let $\tau_c = \{(t^1, m_1, c_1), (t^2, m_2, c_2), \ldots, (t^q, m_q, c_q)\}$ denotes the list of construction query-responses, where each $t^i = t_1^i \| t_2^i \| t_3^i$ is a concatenation of three n-bit strings, and $\tau_p = \{(L_1, u_1, v_1), (L_2, u_2, v_2), \ldots, (L_p, u_p, v_p)\}$ denotes the list of ideal-cipher query-responses, where L_i denotes the ideal-cipher key chosen at the i-th query. For the sake of convenience, we assume that the oracle releases some intermediate values to the distinguisher after the interaction is over, but before $\mathcal{A}$ outputs its decision bit. In the real world, the oracle releases the block cipher key k and the $(y_1^i, y_2^i), i \in [q]$ tuple. On the other hand, the oracle in the ideal world randomly samples n-bit dummy key k and computes $(y_1^i, y_2^i), i \in [q]$ tuple, where y_1^i and y_2^i are computed similar to the real world and finally release them to the distinguisher. Therefore, the extended transcript of the attack is $\tau = (\tau_c, \tau_p, (y_1^i, y_2^i)_{i \in [q]}, k)$.

4.1 Defining the Bad Transcripts

Let Θ denote the set of all attainable transcripts. We call an attainable transcript $\tau \in \Theta$ is bad if it satisfies either of the following:

1. $\mathsf{Bad1}$: $\exists i \in [q], \alpha \neq \beta \in [p] : k \oplus t_2^i = L_\alpha,\ \ y_1^i \oplus t_2^i = L_\beta$
2. $\mathsf{Bad2}$: $\exists i \neq j \in [q] : y_1^i \oplus y_1^j = m_i \oplus m_j,\ y_2^i \oplus y_2^j = t_1^i \oplus t_1^j$
3. $\mathsf{Bad3}$: $\exists i \neq j \in [q] : y_1^i \oplus y_1^j = c_i \oplus c_j,\ y_2^i \oplus y_2^j = t_1^i \oplus t_1^j$
4. $\mathsf{Bad4}$: $\exists i \in [q], \alpha \in [p] : m_i \oplus y_1^i = u_\alpha,\ 2k \oplus y_2^i \oplus t_1^i = L_\alpha$
5. $\mathsf{Bad5}$: $\exists i \in [q], \alpha \in [p] : c_i \oplus y_1^i = v_\alpha,\ 2k \oplus y_2^i \oplus t_1^i = L_\alpha$
6. $\mathsf{Bad6}$: $\exists i \neq j \in [q] : 2k \oplus y_2^i \oplus t_1^i = k \oplus t_2^j,\ m_i \oplus y_1^i = t_1^j$
7. $\mathsf{Bad7}$: $\exists i \neq j \in [q] : 2k \oplus y_2^i \oplus t_1^i = k \oplus t_2^j,\ c_i \oplus y_1^i = y_1^j$
8. $\mathsf{Bad8}$: $\exists i \neq j \in [q] : 2k \oplus y_2^i \oplus t_1^i = y_1^j \oplus t_2^j,\ m_i \oplus y_1^i = t_3^j$
9. $\mathsf{Bad9}$: $\exists i \neq j \in [q] : 2k \oplus y_2^i \oplus t_1^i = y_1^j \oplus t_2^j,\ c_i \oplus y_1^i = y_2^j$
10. $\mathsf{Bad10}$: $\exists i \in [q] : 2k \oplus y_2^i \oplus t_1^i = k \oplus t_2^i$

In the following lemma we state that one of the bad events holds in the ideal world with very low probability.

Lemma 5. *Let Θ_{b} denote the set of all bad transcripts and recall that X_{id} denotes the random variable of transcript τ induced in the ideal world. Then, we have the following:*

$$\Pr[\mathsf{X}_{\mathrm{id}} \in \Theta_{\mathrm{b}}] \leq \frac{q}{2^n} + \frac{12q^2}{2^{2n}} + \frac{6q^2p}{2^{2n}} + \frac{4qp}{2^{2n}} + \frac{11qp^2}{2^{2n}}. \tag{3}$$

Proof Let us denote $\mathsf{Bad} = \mathsf{Bad1} \vee (\vee_{i=2}^{9} \mathsf{Badi} \mid \overline{\mathsf{Bad1}}) \vee \mathsf{Bad10}$. Therefore, by applying the union bound, we have

$$\Pr[\mathsf{Bad}] \leq \Pr[\mathsf{Bad1}] + \sum_{i=2}^{9} \Pr[\mathsf{Badi} \mid \overline{\mathsf{Bad1}}] + \Pr[\mathsf{Bad10}].$$

Therefore, to bound the probability of the event Bad, we individually bound the probability of the event $\mathsf{Bad1}, \mathsf{Bad10}$ and Badi for $2 \leq i \leq 10$ conditioned on the complement of the event $\mathsf{Bad1}$ and then we apply the union bound to obtain the final result.

Bounding $\mathsf{Bad1}$: We bound the event in two cases: (i) when $t_1^i \neq u_\alpha$ for all $\alpha \in [p]$ and (ii) when $\exists \alpha \in [p]$ such that $t_1^i = u_\alpha$. To bound the first case, we note that if $t_1^i \neq u_\alpha$, then y_1^i is fresh and thus, we use the randomness of y_1^i to bound the event $y_1^i \oplus t_2^i = L_\beta$ to at most $1/(2^n - p) \leq 2/2^n$ assuming $p \leq 2^{n-1}$. Moreover, due to the randomness of k, we bound the event $k \oplus t_2^i = L_\alpha$ to $1/2^n$. Therefore, by varying over all possible choices of indices, we have

$$\Pr[\mathsf{Bad1}] \leq 2qp^2/2^{2n}. \tag{4}$$

To bound the second case, we consider three following sub-cases: (a) when $\alpha > i$ and the α-th ideal-cipher query is forward one. In that case, for a fixed choice of indices $i \in [q]$, $\alpha, \beta \in [p]$, the probability of the event $k \oplus t_2^i = L_\alpha, t_1^i = u_\alpha, v_\alpha \oplus t_2^i = L_\beta$ is upper bounded by $1/2^n \cdot 1/(2^n - p)$ due to the randomness of the key k and the randomness of the ideal-cipher query output v_α. By varying over all possible choices of indices $i \in [q], \alpha \neq \beta \in [p]$ and by assuming $p \leq 2^{n-1}$, we have

$$\Pr[\mathsf{Bad1}] \leq 2qp^2/2^{2n}. \tag{5}$$

(b) when $\alpha > i$ and the α-th ideal-cipher query is inverse one. In that case, for a fixed choice of indices $i \in [q]$, $\alpha, \beta \in [p]$, the probability of the event $k \oplus t_2^i = L_\alpha, t_1^i = u_\alpha, v_\alpha \oplus t_2^i = L_\beta$ is upper bounded by $1/2^n \cdot 1/(2^n - p)$ due to the randomness of the key k and the randomness of the ideal-cipher query output u_α. By varying over all possible choices of indices $i \in [q], \alpha \neq \beta \in [p]$ and by assuming $p \leq 2^{n-1}$, we have

$$\Pr[\mathsf{Bad1}] \leq 2qp^2/2^{2n}. \tag{6}$$

(c) On the other hand, if $\alpha < i$, then we cannot use the randomness of v_α. In that case, for a fixed choices of i, α and β, the probability that $k \oplus t_2^i = L_\alpha, t_1^i = u_\alpha, v_\alpha \oplus t_2^i = L_\beta$ holds is upper bounded by $1/2^n$ due to the randomness of the key k. However, the number of choices of i, α and β such that $v_\alpha \oplus t_2^i = L_\beta$ holds is at most $qp^2/2^n$ by the virtue of the Sum-Capture Lemma. Therefore, in both the cases, we have

$$\Pr[\mathsf{Bad1}] \leq qp^2/2^{2n}. \tag{7}$$

Therefore, by combining Eqn. (4), Eqn. (5), Eqn. (6), and Eqn. (7), we have

$$\Pr[\mathsf{Bad1}] \leq 7qp^2/2^{2n}. \tag{8}$$

Bounding $\mathsf{Bad2} \mid \overline{\mathsf{Bad1}}$: To bound the event, we need to bound the probability of the following two equations hold:

$$\mathcal{E} = \begin{cases} (1): \ y_1^i \oplus y_1^j = m_i \oplus m_j \\ (2): \ y_2^i \oplus y_2^j = t_1^i \oplus t_1^j \end{cases}$$

Now, we bound the probability of this event in several cases as follows:

Case I. $(t_1^i, t_2^i) = (t_1^j, t_2^j)$**:** If the condition happens, then it implies that $y_1^i = y_1^j$ and thus from Eqn. (1), we have $m_i = m_j$. Since the distinguisher is non-trivial, therefore, it implies that $t_3^i \neq t_3^j$. But then it implies that $y_2^i \neq y_2^j$. However, from Eqn. (2), we have $y_2^i = y_2^j$ which is a contradiction and hence the probability of the event would be zero.

Case-II. y variables are determined by ideal-cipher query: Without loss of generality, we assume that y_1^i is determined by an ideal-cipher query. Then by the virtue of $\overline{\mathsf{Bad}_1}$, y_2^i fresh, i.e., it is not determined by any ideal-cipher query. Hence, the above equations are boiled down to the following:

$$\begin{cases} k \oplus t_2^i = L_\alpha \\ t_1^i = u_\alpha \\ y_1^j = m_i \oplus m_j \oplus v_\alpha \\ y_2^i \oplus y_2^j = t_1^i \oplus t_1^j \end{cases}$$

Using the randomness of y_2^i and the randomness of the key k, the probability of the above event is bounded by $1/2^n \cdot 1/(2^n - p)$. However, the number of choices of i, j, α is $\binom{q}{2}p$. By assuming $p \leq 2^{n-1}$, we have

$$\Pr[\mathsf{Bad2} \mid \overline{\mathsf{Bad1}}] \leq q^2p/2^{2n}. \tag{9}$$

Case-III. none of the y variables are determined by ideal-cipher query: We consider this case in several sub-cases as follows:

1. We consider the case when $t_1^i = t_3^i, t_1^j = t_3^j, k = y_1^i$ and $y_1^i = y_2^i$. This event implies $y_1^i = y_2^i$ and $y_1^j = y_2^j$. Hence, the rank of the system of equations $\mathcal{E}$ is 1 and hence $\mathcal{E}$ holds with probability at most $1/(2^n - p)$. However, we also have the randomness from the equation $k = y_1^i$ which additionally contributes to 2^{-n} in the probability. Therefore, for a fixed choice of indices, the probability that $\mathcal{E}$ holds is at most $1/2^n \cdot 1/(2^n - p)$. By varying over the all possible choices of indices and by assuming $p \leq 2^{n-1}$, we have
$$\Pr[\mathsf{Bad2}] \leq q^2/2^{2n}. \tag{10}$$
2. We consider a case when $t_1^i = t_3^j, t_3^i = t_1^j, k \oplus y_1^j = t_2^i \oplus t_2^j$ and $y_1^i = y_1^j$. This event implies $y_1^i = y_2^j$ and $y_1^j = y_2^i$. Hence, the rank of the system of equations $\mathcal{E}$ is 1 and hence $\mathcal{E}$ holds with probability at most $1/(2^n - p)$. However, we also have the randomness from the equation $k \oplus y_1^j = t_2^i \oplus t_2^j$ which additionally contributes to 2^{-n} in the probability. Therefore, for a fixed choice of indices, the probability that $\mathcal{E}$ holds is at most $1/2^n \cdot 1/(2^n - p)$. By varying over the all possible choices of indices, we have
$$\Pr[\mathsf{Bad2}] \leq q^2/2^{2n}. \tag{11}$$
3. If the above two cases do not happen, then the rank of the system of equations $\mathcal{E}$ is 2 and in that case, we obtain two fresh random variables y_1^i and y_2^i which jointly contributes $1/(2^n - p)^2$ to the probability of the above system of equations $\mathcal{E}$. Hence, for a fixed choice of indices, the probability that $\mathcal{E}$ holds is at most $1/(2^n - p)^2$. By varying over the all possible choices of indices and by assuming $p \leq 2^{n-1}$, we have
$$\Pr[\mathsf{Bad2}] \leq 2q^2/2^{2n}. \tag{12}$$

Therefore, by combining Eqn. (9)-Eqn. (12), we have
$$\Pr[\mathsf{Bad2} \mid \overline{\mathsf{Bad1}}] \leq 4q^2/2^{2n} + q^2p/2^{2n}. \tag{13}$$

Bounding $\mathsf{Bad3} \mid \overline{\mathsf{Bad1}}$: Bounding $\mathsf{Bad3} \mid \overline{\mathsf{Bad1}}$ is identical to that of $\mathsf{Bad2} \mid \overline{\mathsf{Bad1}}$ and hence we have,
$$\Pr[\mathsf{Bad3} \mid \overline{\mathsf{Bad1}}] \leq 4q^2/2^{2n} + q^2p/2^{2n}. \tag{14}$$

Bounding $\mathsf{Bad4} \mid \overline{\mathsf{Bad1}}$: We bound this event in several sub-cases as follows: (a) If y_1^i is not determined by any ideal-cipher query, then for a fixed choices of indices, using the randomness of y_1^i and key k, we bound this event up to $1/2^n \cdot 1/(2^n - p)$. By varying the choices of indices and by assuming $p \leq 2^{n-1}$, we have
$$\Pr[\mathsf{Bad4}] \leq 2qp/2^{2n} \tag{15}$$

(b) On the other hand, if y_1^i is determined by ideal-cipher query, let $t_1^i = u_\beta, k \oplus t_2^i = L_\beta$ for some $\beta \in [p]$ which implies $y_1^i = v_\beta$, then by the virtue of $\overline{\mathsf{Bad1}}$, y_2^i must be fresh. In that case, the event gets boils down to the following system of equations:
$$\begin{cases} m_i \oplus y_1^i = u_\alpha \\ 2k \oplus y_2^i \oplus t_1^i = L_\alpha \\ t_1^i = u_\beta \\ k \oplus t_2^i = L_\beta \end{cases}$$

For a fixed choices of indices, using the randomness of y_2^i and k, the probability of the above system of equation holds is $1/2^n \cdot 1/(2^n - p)$. Moreover, the number of choices of indices is qp^2. Thus, we have
$$\Pr[\mathsf{Bad4} \mid \overline{\mathsf{Bad1}}] \leq 2qp^2/2^{2n}. \tag{16}$$

Therefore, by combining Eqn. (15), and Eqn. (16), we have

$$\Pr[\mathsf{Bad4} \mid \overline{\mathsf{Bad1}}] \leq 2qp/2^{2n} + 2qp^2/2^{2n}. \tag{17}$$

Bounding $\mathsf{Bad5} \mid \overline{\mathsf{Bad1}}$: Bounding $\mathsf{Bad5} \mid \overline{\mathsf{Bad1}}$ is identical to that of $\mathsf{Bad4} \mid \overline{\mathsf{Bad1}}$ and hence, we have

$$\Pr[\mathsf{Bad5} \mid \overline{\mathsf{Bad1}}] \leq 2qp/2^{2n} + 2qp^2/2^{2n}. \tag{18}$$

Bounding $\mathsf{Bad6} \mid \overline{\mathsf{Bad1}}$: For a fixed choice of indices, the above event boils down to bounding the following system of equations hold:

$$\begin{cases} 3k = y_2^i \oplus t_1^i \oplus t_2^j \\ y_1^i = m_i \oplus t_1^j \end{cases}$$

Now, we analyze the probability of the above event in the following two sub-cases: (a) when y_1^i is fresh, then we use the randomness of the key k and y_1^i to bound the probability to at most $1/2^n \cdot 1/(2^n - p)$. However, by varying the all possible choices of indices and by assuming $p \leq 2^{n-1}$, we have

$$\Pr[\mathsf{Bad6} \mid \overline{\mathsf{Bad1}}] \leq q^2/2^{2n}. \tag{19}$$

(b) On the other hand, if y_1^i is not fresh, i.e., y_1^i is determined from ideal-cipher query, then, the event boils down to the following system of equations hold:

$$\begin{cases} 3k = y_2^i \oplus t_1^i \oplus t_2^j \\ k \oplus t_2^i = L_\alpha \end{cases}$$

where $\alpha \in [p]$. Since y_1^i is determined from ideal-cipher query, by the virtue of $\overline{\mathsf{Bad1}}$, y_2^i is fresh. Now, we use the randomness of the key k and y_2^i to bound the probability to at most $1/2^n \cdot 1/(2^n - p)$. However, by varying the all possible choices of indices and by assuming $p \leq 2^{n-1}$, we have

$$\Pr[\mathsf{Bad6} \mid \overline{\mathsf{Bad1}}] \leq q^2 p/2^{2n}. \tag{20}$$

By combining Eqn. (19) and Eqn. (20), we have

$$\Pr[\mathsf{Bad6} \mid \overline{\mathsf{Bad1}}] \leq q^2/2^{2n} + q^2 p/2^{2n}. \tag{21}$$

Bounding $\mathsf{Bad7} \mid \overline{\mathsf{Bad1}}$: Bounding this event is identical to that of bounding $\mathsf{Bad6} \mid \overline{\mathsf{Bad1}}$ and hence we have

$$\Pr[\mathsf{Bad7} \mid \overline{\mathsf{Bad1}}] \leq q^2/2^{2n} + q^2 p/2^{2n}. \tag{22}$$

Bounding $\mathsf{Bad8}$: For a fixed choice of indices, the above event boils down to bounding the following system of equations hold:

$$\begin{cases} 2k = y_2^i \oplus t_1^i \oplus y_1^j \oplus t_2^j \\ y_1^i = m_i \oplus t_3^j \end{cases}$$

Now, we analyze the probability of the above event in the following two sub-cases: (a) when y_1^i is fresh, then we use the randomness of the key k and y_1^i to bound the probability to at most $1/2^n \cdot 1/(2^n - p)$. However, by varying the all possible choices of indices and by assuming $p \leq 2^{n-1}$, we have

$$\Pr[\mathsf{Bad8}] \leq q^2/2^{2n}. \tag{23}$$

(b) On the other hand, if y_1^i is not fresh, i.e., y_1^i is determined from ideal-cipher query, then, the event boils down to the following system of equations hold:

$$\begin{cases} 2k = y_2^i \oplus t_1^i \oplus y_1^j \oplus t_2^j \\ k \oplus t_2^i = L_\alpha \end{cases}$$

where $\alpha \in [p]$. Since y_1^i is determined from ideal-cipher query, by the virtue of $\overline{\mathsf{Bad1}}$, y_2^i is fresh. Now, we use the randomness of the key k and y_2^i to bound the probability to at most $1/2^n \cdot 1/(2^n - p)$. However, by varying the all possible choices of indices and by assuming $p \leq 2^{n-1}$, we have

$$\Pr[\mathsf{Bad8} \mid \overline{\mathsf{Bad1}}] \leq q^2 p/2^{2n}. \tag{24}$$

By combining Eqn. (23) and Eqn. (24), we have

$$\Pr[\mathsf{Bad8} \mid \overline{\mathsf{Bad1}}] \leq q^2/2^{2n} + q^2 p/2^{2n}. \tag{25}$$

Bounding $\mathsf{Bad9} \mid \overline{\mathsf{Bad1}}$: Bounding this event is identical to that of bounding $\mathsf{Bad8} \mid \overline{\mathsf{Bad1}}$ and hence we have

$$\Pr[\mathsf{Bad9} \mid \overline{\mathsf{Bad1}}] \leq q^2/2^{2n} + q^2 p/2^{2n}. \tag{26}$$

Bounding $\mathsf{Bad10}$: For a fixed choice of index i, the probability of the event $2k \oplus y_2^i \oplus t_1^i = k \oplus t_2^i$ is upper bounded by 2^{-n} due to the randomness of the n-bit key k. By varying over all possible choices of indices, we have

$$\Pr[\mathsf{Bad10}] \leq q/2^n \tag{27}$$

We derive the bound of Lemma 5 by combining Eqn. (8)-Eqn. (27). □

4.2 Good Transcript Analysis

Let $\tau = (\tau_c, \tau_p, (y_1^i, y_2^i)_{i \in [q]}, k)$ be a good transcript. We consider a set

$$\mathcal{S} = \{\big((k \oplus t_2^1, t_1^1, y_1^1), (y_1^1 \oplus t_2^1, t_3^1, y_2^1)\big), \ldots, \big((k \oplus t_2^q, t_1^q, y_1^q), (y_1^q \oplus t_2^q, t_3^q, y_2^q)\big)\}$$

that records the (key, input, output) triplet of the first and second block cipher call of the construction across all q construction queries. For each n-bit string $K \in \{0,1\}^n$, we define a list $\mathsf{IC}(K) = \{(L, u, v) \in \tau_p : L = K\}$ that records the (ideal-cipher key, input, output) triplet across all p ideal-cipher queries such that the ideal-cipher key is K. We maintain a list of integers $\mathcal{L}_1$, where we include an index $i \in [q]$ in $\mathcal{L}_1$, if $\exists \alpha \in [p]$ such that $k \oplus t_2^i = L_\alpha$. Similarly, we maintain a list of integers $\mathcal{L}_2$, where we include an index $i \in [q]$ in $\mathcal{L}_2$, if $\exists \alpha \in [p]$ such that $y_1^i \oplus t_2^i = L_\alpha$. Note that, $\mathcal{L}_1 \cap \mathcal{L}_2 = \phi$, otherwise the event $\mathsf{Bad1}$ would have been hold. Now, we define a set

$$\mathsf{H}_1 := \{(k \oplus t_2^i, t_1^i, y_1^i), (y_1^i \oplus t_2^i, t_3^i, y_3^i) : i \notin \mathcal{L}_1 \cup \mathcal{L}_2\}.$$

Note that, H_1 records the (key, input, output) triplet of the first and second block cipher call of the construction across all q construction queries such that both keys of the block cipher have not collided with any ideal-cipher key. Moreover, $\mathsf{H}_1 \subseteq \mathcal{S}$. Finally, for each $i \in \mathcal{L}_1$, we include the element $(k \oplus t_2^i, t_1^i, y_1^i)$ into the list $\mathsf{IC}(k \oplus t_2^i)$, i.e., $\mathsf{IC}(k \oplus t_2^i) \leftarrow \mathsf{IC}(k \oplus t_2^i) \cup \{(k \oplus t_2^i, t_1^i, y_1^i)\}$, and, for each $i \in \mathcal{L}_2$, we include the element $(y_1^i \oplus t_2^i, t_3^i, y_3^i)$ into the list $\mathsf{IC}(y_1^i \oplus t_2^i)$, i.e., $\mathsf{IC}(y_1^i \oplus t_2^i) \leftarrow \mathsf{IC}(y_1^i \oplus t_2^i) \cup \{(y_1^i \oplus t_2^i, t_3^i, y_3^i)\}$. For each key $K \in \{0,1\}^n$, we define the set

$$\mathsf{H}_2(K) := \{(2k \oplus y_2 \oplus t_1, m \oplus y_1, c \oplus y_1) : (t_1 \| t_2 \| t_3, m, c) \in \tau_c, 2k \oplus y_2 \oplus t_1 = K\}$$

that records the (key, input, output) triplet of the third block cipher call such that the key is K. Similarly, for each tweak $t \in \{0,1\}^{3n}$, we define the set

$$\mathsf{H}(t) := \{(t_1\|t_2\|t_3, m, c) \in \tau_c : t_1\|t_2\|t_3 = t\}$$

which records all q construction queries and response excluding the block cipher key such that the tweak of the construction query is t. Finally, for each key $K \in \{0,1\}^n$, we define the set

$$\mathsf{Z}(K) := \{(t_1\|t_2\|t_3) : (t_1\|t_2\|t_3, m, c) \in \tau_c, 2k \oplus y_2 \oplus t_1 = K\}.$$

It is to be noted that as the transcript is good, for each key $K \in \{0,1\}^n$, we have $\mathsf{H}_2(K) \cap \mathsf{H}_1 = \phi$, otherwise either of the event Bad6-Bad10 would have been hold. Similarly, for each key $K \in \{0,1\}^n$, we have $\mathsf{H}_2(K) \cap \mathsf{IC}(K) = \phi$, otherwise either of the events Bad4-Bad9 would have been hold. Finally, by the virtue of the definition, we have $\mathsf{H}_1 \cap \mathsf{IC}(K) = \phi$ for each key $K \in \{0,1\}^n$.

Let us fix a key $K \in \{0,1\}^n$. For each $t \in \mathsf{Z}(K)$, $|\mathsf{H}(t)|$ denotes the number of construction queries with tweak t. Then, we have for each key $K \in \{0,1\}^n$,

$$\sum_{t \in \mathsf{Z}(K)} |\mathsf{H}(t)| = |\mathsf{H}_2(K)|.$$

For the sake of simplicity, let us denote $|\mathsf{H}_1| = \alpha_1$. For each $K \in \{0,1\}^n$, we denote $|\mathsf{IC}(K)| = \alpha_{\mathsf{ic}}(K)$, $|\mathsf{H}_2(K)| = \alpha_2(K)$ and for each tweak $t \in \{0,1\}^{3n}$, we denote $|\mathsf{H}(t)| = \alpha(t)$. Therefore, for a fixed good transcript τ, the ideal interpolation probability becomes

$$\begin{aligned}
\Pr[\mathsf{X}_{\mathrm{id}} = \tau] &= \frac{1}{2^n} \cdot \prod_{K \in \{0,1\}^n} \cdot \prod_{j=0}^{\alpha_2(K)-1} \frac{1}{2^n - j} \cdot \Bigg(\prod_{K \in \{0,1\}^n} \prod_{j=0}^{\alpha_{\mathsf{ic}}(K)-1} \frac{1}{2^n - j} \Bigg) \\
&\qquad \cdot \Bigg(\prod_{K \in \{0,1\}^n} \prod_{t \in \mathsf{Z}(K)} \prod_{p=0}^{\alpha(t)-1} \frac{1}{2^n - p} \Bigg) \\
&= \frac{1}{2^n} \cdot \prod_{K \in \{0,1\}^n} \cdot \prod_{j=0}^{\alpha_2(K)-1} \frac{1}{2^n - j} \cdot \Bigg(\prod_{K \in \{0,1\}^n} \prod_{j=0}^{\alpha_{\mathsf{ic}}(K)-1} \frac{1}{2^n - j} \prod_{p=0}^{\alpha_2(K)-1} \frac{1}{2^n - p} \Bigg).
\end{aligned}$$

To bound the real interpolation probability, the number of times block cipher is called for deriving sub-keys is α_1. However, number of times block cipher is called for ideal-cipher queries and construction queries is $\alpha_{\mathsf{ic}}(K) + \alpha_2(K)$ for each key $K \in \{0,1\}^n$. Therefore, we have

$$\Pr[\mathsf{X}_{\mathrm{re}} = \tau] = \frac{1}{2^n} \cdot \prod_{K \in \{0,1\}^n} \cdot \prod_{j=0}^{\alpha_2(K)-1} \frac{1}{2^n - j} \cdot \Bigg(\prod_{K \in \{0,1\}^n} \prod_{j=0}^{\alpha_{\mathsf{ic}}(K)+\alpha_2(K)-1} \frac{1}{2^n - j} \Bigg).$$

Since for each key $K \in \{0,1\}^n$, we have

$$\prod_{j=0}^{\alpha_{\mathsf{ic}}(K)-1} \frac{1}{2^n - j} \prod_{p=0}^{\alpha_2(K)-1} \frac{1}{2^n - p} \leq \prod_{j=0}^{\alpha_{\mathsf{ic}}(K)+\alpha_2(K)-1} \frac{1}{2^n - j},$$

the ratio of the real to ideal interpolation probability becomes ≥ 1, which proves the result. $\square$

Remark 2. The security of our construction holds as long as the online query complexity $q \leq 2^{2n/3}$ and the offline query complexity $p \leq 2^{2n/3}$. Ideally, the offline query complexity should go up to 2^n as they are cheaper than the online queries, but we believe that it is challenging to improve the security bound of our construction that tolerates offline query complexity up to 2^n.

5 Optimally Secure TBC with $3n$-bit Tweaks Using Four Block Cipher Calls

In this section, we show that to process $3n$ bit tweaks using four block cipher calls, having one tweak-dependent block cipher key is both necessary and sufficient condition for achieving security up to $O(2^n)$ queries. In the following, we first show that at least one tweak-dependent key is necessary to construct TBCs with $3n$-bit tweak from four block ciphers calls. Followed by, we show that at least one tweak-dependent key is sufficient to construct TBCs with $3n$-bit tweak from four block ciphers calls.

5.1 Generic Birthday Attacks on TBCs with $3n$-bit tweak from Four BC with All Tweak-independent Keys

In this subsection, we will show that at least one tweak-dependent key is necessary to construct TBCs with $3n$-bit tweak from four block ciphers. In other words, we exhibit birthday bound attacks on all TBC constructions with four block cipher calls that process $3n$ bit tweaks with no tweak dependency key. More precisely, we consider the generic construction using four block ciphers where no block cipher keys are tweak-dependent, dubbed $\mathcal{C}_8$ as depicted in Fig.13, and present a birthday attack on the construction.

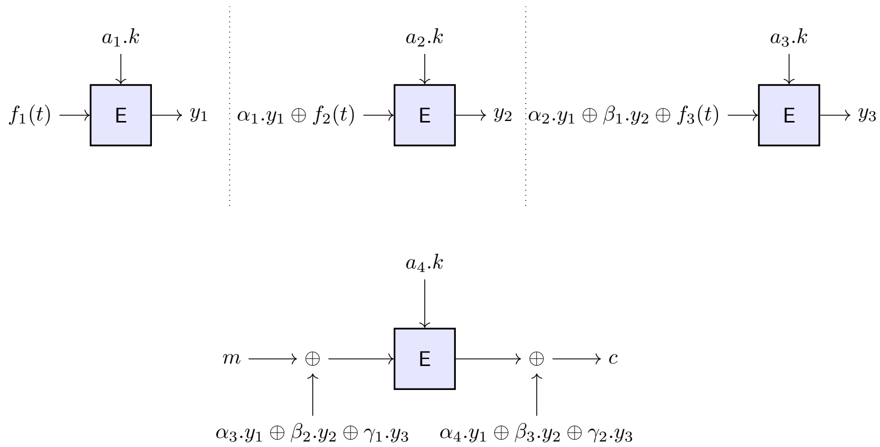


Figure 13: Construction $\mathcal{C}_8$: All the four block cipher uses tweak-independent keys.

To mount an attack on this generic construction, our strategy is as follows:

Case 1: $\gamma_1 = 0$. In this case we find a constant query attack as follows:

1. Find t^1, t^2 satisfying $f_1(t^1) = f_1(t^2)$ and $f_2(t^1) = f_2(t^2)$. This choice makes $y_1^1 = y_1^2$ and $y_2^1 = y_2^2$.
2. Make two query with (m, t^1) and (m, t^2). We have the same (input, key) for both queries. Note that, corresponding cipher texts c_1, c_2 will satisfy $c_1 \oplus c_2 = \gamma_2(y_3^1 \oplus y_3^2)$.
3. Finally, make two additional queries $(m \oplus \Delta, t^1)$ and $(m \oplus \Delta, t^2)$, where $\Delta \neq 0$. Let $c_1^\star$ and $c_2^\star$ be two cipher texts. Return 1 if $c_1^\star \oplus c_2^\star = c_1 \oplus c_2$. Note that, this equation happens for TBC construction is 1, while for random tweakable permutation, the probability is only $1/2^n$.

Case 2: $\gamma_1 \neq 0$. Here we find a birthday attack as follows:

1. Find $2^{n/2}$ many tweaks such that for each pair of tweaks (t^i, t^j), we have $f_1(t^i) = f_1(t^j)$, $f_2(t^i) = f_2(t^j)$, and $f_3(t^i) \neq f_3(t^j)$. Note that, with this choice of tweaks, we will have $y_1^i = y_1^j$, and $y_2^i = y_2^j$, for all (i, j).

2. Now we make $2^{n/2}$ queries (m_i, t^i) such that all the m_i-values are distinct. It is easy to see that a collision in the input of the final block cipher happens when $\gamma_1(y_3^i \oplus y_3^j) = m_i \oplus m_j$. Now due to birthday paradox, we expect one such collision. Moreover, this collision is detectable as in this case, we have $\gamma_1(c_i \oplus c_j) = \gamma_2(m_i \oplus m_j)$.

3. Finally, we can distinguish the TBC from a random tweakable permutation by making two additional oracle queries $(m_i \oplus \Delta, t^i)$, $(m_j \oplus \Delta, t^j)$, where $\Delta \neq 0$, and verifying if the corresponding outputs, say $c_i^\star$ and $c_j^\star$ satisfies the following equation: $c_i^\star \oplus c_j^\star = c_i \oplus c_j$.

An algorithmic description of the attack is shown in Fig. 24 (See Supplementary Material C.8).

Remark 3. In general, we can mount a similar generic birthday attack on TBCs with rn-bit tweak from $(r + 1)$ Block ciphers if all the block cipher keys are Tweak-independent.

5.2 Optimal Secure TBC with $3n$-bit tweak from Four BC with one Tweak-dependent Key

Let $\mathsf{E} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a n-bit block cipher. The tweakable block cipher $\widetilde{\mathsf{G3}} : \{0,1\}^n \times \{0,1\}^{3n} \times \{0,1\}^n \to \{0,1\}^n$ with a $3n$ bit tweak using four block cipher calls is constructed as follows: three block cipher calls are first invoked in parallel to produce three masks y_1, y_2 and y_3 from the tweaks t_1, t_2, t_3 and the master key k. By using $y_1 \oplus y_2$ to mask the input and by using $y_1 \oplus y_3$ to mask the output, and using $y_2 \oplus y_3$, the master key k, and t_1 to provide variety in the sub-key, a fourth block cipher call is then invoked to encrypt the message m to the ciphertext c. A pictorial illustration of the construction $\widetilde{\mathsf{G3}}$ is given in Fig. 14.

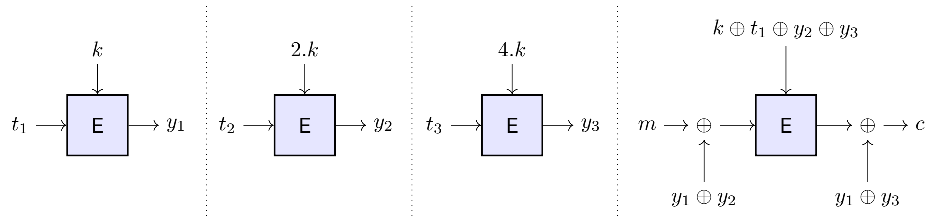


Figure 14: $\widetilde{\mathsf{G3}}$ construction: TBC with 3n-bit tweaks using four block cipher calls. We impose a natural ordering on the block cipher calls from left to right.

In the following we show that $\widetilde{\mathsf{G3}}$ is a secure tweakable block cipher with $3n$ bit tweaks against all adversaries that makes roughly 2^n construction and ideal-cipher queries. Formally, we have the following result:

Theorem 3. *Let $\mathcal{A}$ be an adversary making at most q construction queries and p ideal-cipher queries including both forward and backward queries. Then,*

$$\mathbf{Adv}^{\text{tsprp-icm}}_{\widetilde{\mathsf{G3}}}(\mathcal{A}) \leq \frac{4q(p+q)}{2^{2n}} + \frac{4q + 3p + 1}{2^n}.$$

Proof. We consider $\mathcal{A}$ to be a computationally unbounded deterministic distinguisher that interacts with a pair of oracles in either the real world $(\widetilde{\mathsf{G3}}^{\mathsf{E}}, \mathsf{E}^{\pm})$ or in the ideal world $(\widetilde{\mathsf{P}}, \mathsf{E}^{\pm})$. Let us assume that $\mathcal{A}$ makes at most q construction queries and p ideal-cipher queries. Let $\tau_c = \{(t_1^1 \| t_2^1 \| t_3^1, m_1, c_1), \ldots, (t_1^q \| t_2^q \| t_3^q, m_q, c_q)\}$ denote the list of construction query-responses and $\tau_p = \{(L_1, u_1, v_1), (L_2, u_2, v_2), \ldots, (L_p, u_p, v_p)\}$ denote the list of ideal-cipher query-responses. For the sake of proof, let the oracle release some additional value after all of adversary $\mathcal{A}$'s query responses are finished. Note that these additional released values can only increase the adversary's advantage. We assume that the oracle in the real world releases the block cipher key k and the tuple $(y_1^i, y_2^i, y_3^i), i \in [q]$ tuple. On the other hand, the oracle in the ideal world randomly samples n-bit dummy key k and computes $(y_1^i, y_2^i, y_3^i), i \in [q]$ tuple, where y_1^i, y_2^i, and y_3^i are computed similar to the real world and finally released them to the distinguisher. Therefore, the extended transcript of the attack is $\tau = (\tau_c, \tau_p, (y_1^i, y_2^i, y_3^i)_{i \in [q]}, k)$. Let Θ denote the set of all attainable transcripts. We call an attainable transcript $\tau \in \Theta$ is bad if it satisfies either of the following:

1. $\mathsf{Bad1}$: $k = 0$.
2. $\mathsf{Bad2}$: $\exists \alpha \in [p] : L_\alpha \in \{k, 2k, 4k\}$.
3. $\mathsf{Bad3}$: $\exists i \in [q] : k \oplus t_1^i \oplus y_2^i \oplus y_3^i \in \{k, 2k, 4k\}$.
4. $\mathsf{Bad4}$: $\exists i \in [q], \alpha \in [p] : m_i \oplus y_1^i \oplus y_2^i = u_\alpha,\ k \oplus t_1^i \oplus y_2^i \oplus y_3^i = L_\alpha$.
5. $\mathsf{Bad5}$: $\exists i \in [q], \alpha \in [p] : c_i \oplus y_1^i \oplus y_3^i = v_\alpha,\ k \oplus t_1^i \oplus y_2^i \oplus y_3^i = L_\alpha$.
6. $\mathsf{Bad6}$: $\exists i \neq j \in [q] : y_1^i \oplus y_2^i \oplus y_1^j \oplus y_2^j = m_i \oplus m_j,\ y_2^i \oplus y_3^i \oplus y_2^j \oplus y_3^j = t_1^i \oplus t_1^j$.
7. $\mathsf{Bad7}$: $\exists i \neq j \in [q] : y_1^i \oplus y_3^i \oplus y_1^j \oplus y_3^j = c_i \oplus c_j,\ y_2^i \oplus y_3^i \oplus y_2^j \oplus y_3^j = t_1^i \oplus t_1^j$.

In the following lemma we state that one of the bad events holds in the ideal world with very low probability.

Lemma 6. *Let Θ_{b} denote the set of all bad transcripts and recall that X_{id} denotes the random variable of transcript τ induced in the ideal world. Then, we have the following:*

$$\Pr[\mathsf{X}_{\mathrm{id}} \in \Theta_{\mathrm{b}}] \leq \frac{4q(p+q)}{2^{2n}} + \frac{4q + 3p + 1}{2^n}. \tag{28}$$

Proof Let us denote $\mathsf{Bad} = \mathsf{Bad1} \vee \mathsf{Bad2} \vee (\vee_{i=3}^{7} \mathsf{Badi} \mid \overline{\mathsf{Bad2}})$. Therefore, by applying the union bound, we have

$$\Pr[\mathsf{Bad}] \leq \Pr[\mathsf{Bad1}] + \Pr[\mathsf{Bad2}] + \sum_{i=3}^{7} \Pr[\mathsf{Badi} \mid \overline{\mathsf{Bad2}}].$$

Therefore, to bound the probability of the event Bad, we individually bound the probability of the event $\mathsf{Bad1}, \mathsf{Bad2}$ and Badi for $3 \leq i \leq 7$ conditioned on the complement of the event $\mathsf{Bad2}$. Then we apply the union bound to obtain the final result.

Bounding $\mathsf{Bad1}$: It is easy to see the randomness of the key k ensures that

$$\Pr[\mathsf{Bad1}] \leq 1/2^n. \tag{29}$$

Bounding $\mathsf{Bad2}$: For a fixed index $\alpha \in [p]$, the probability of the event $k = L_\alpha$ is upper bounded by $1/2^n$ due to the randomness of the key k. Similarly, the probability of the event $2k = L_\alpha$ is upper bounded by $1/2^n$ and the probability of the event $4k = L_\alpha$ is upper bounded by $1/2^n$ due to the randomness of the key k. By varying over all possible choices of indices $\alpha \in [p]$, we have

$$\Pr[\mathsf{Bad2}] \leq 3p/2^n. \tag{30}$$

Bounding $\mathsf{Bad3} \mid \overline{\mathsf{Bad2}}$: For a fixed choice of index i, we bound the event $k \oplus t_1^i \oplus y_2^i \oplus y_3^i = k$, which boils down to the event $y_2^i \oplus y_3^i = t_1^i$. Due to the event $\overline{\mathsf{Bad2}}$, y_2^i variable is fresh and hence we bound the probability of the event to at most $1/(2^n - p)$. Similarly, for a fixed choice of index i, we bound the event $k \oplus t_1^i \oplus y_2^i \oplus y_3^i = 2k$, which boils down to the event

$$k = (2 \oplus 1)^{-1}(y_2^i \oplus y_3^i \oplus t_1^i).$$

Using the entropy of the random variable k, we bound the probability of the event at most $1/2^n$. Similarly, for a fixed choice of index i, we bound the event $k \oplus t_1^i \oplus y_2^i \oplus y_3^i = 4k$, which boils down to the event

$$k = (2^2 \oplus 1)^{-1}(y_2^i \oplus y_3^i \oplus t_1^i).$$

Using the entropy of the random variable k, we bound the probability of the event at most $1/2^n$. Therefore, by varying over all possible choices of indices and by assuming $p \leq 2^{n-1}$, we have

$$\Pr[\mathsf{Bad3} \mid \overline{\mathsf{Bad2}}] \leq 4q/2^n. \tag{31}$$

Bounding $\mathsf{Bad4} \mid \overline{\mathsf{Bad2}}$: For a fixed choice of indices $i \in [q], \alpha \in [p]$, the probability of the event

$$\begin{cases} y_1^i \oplus y_2^i = m_i \oplus u_\alpha \\ k \oplus y_2^i \oplus y_3^i = t_1^i \oplus L_\alpha \end{cases}$$

is upper bounded by $1/2^n \cdot 1/(2^n - p)$ due to the randomness of the key k and y_1^i as y_1^i is not determined by ideal-cipher query due to the virtue of $\neg\mathsf{Bad2}$. By varying over all possible choices of indices $i \in [q], \alpha \in [p]$ and by assuming $p \leq 2^{n-1}$, we have

$$\Pr[\mathsf{Bad4} \mid \overline{\mathsf{Bad2}}] \leq 2qp/2^{2n}. \tag{32}$$

Bounding $\mathsf{Bad5} \mid \overline{\mathsf{Bad2}}$: Bounding this event is identical to that of $\mathsf{Bad4} \mid \overline{\mathsf{Bad2}}$ and hence we have

$$\Pr[\mathsf{Bad5} \mid \overline{\mathsf{Bad2}}] \leq 2qp/2^{2n}. \tag{33}$$

Bounding $\mathsf{Bad6} \mid \overline{\mathsf{Bad2}}$: We bound the probability of this event in several sub-cases as follows:

1. if $(t_1^i, t_2^i) = (t_1^j, t_2^j)$, then it implies that $y_1^i = y_1^j$ and $y_2^i = y_2^j$ which follows from the construction. Hence, it implies from the above equation

   $$(y_1^i \oplus y_2^i) \oplus (y_1^j \oplus y_2^j) = m_i \oplus m_j$$

   that $m_i = m_j$. On the other hand, the above condition also implies from the equation

   $$(y_2^i \oplus y_3^i) \oplus (y_2^j \oplus y_3^j) = t_1^i \oplus t_1^j$$

   that $y_3^i = y_3^j$ that follows from the construction which in turn implies that $t_3^i = t_3^j$. However, if the tweaks are same for i-th and j-th query, then the corresponding message should be different as we assume non-trivial distinguisher which is violated from the above sequence of logical events. Therefore, in this case, the probability of the event is zero.

2. if $(t_1^i, t_3^i) = (t_1^j, t_3^j)$, then it implies that $y_1^i = y_1^j$ and $y_3^i = y_3^j$ which follows from the construction. Hence, it implies from the above equation

$$(y_2^i \oplus y_3^i) \oplus (y_2^j \oplus y_3^j) = t_1^i \oplus t_1^j$$

that $y_2^i = y_2^j$ which in turn implies that $t_2^i = t_3^i$. On the other hand, the above condition implies from the above equation

$$(y_1^i \oplus y_2^i) \oplus (y_1^j \oplus y_2^j) = m_i \oplus m_j$$

that $m_i = m_j$. However, if the tweaks are same for i-th and j-th query, then the corresponding message should be different as we assume non-trivial distinguisher which is violated from the above sequence of logical events. Therefore, in this case, the probability of the event is zero.

3. if $(t_2^i, t_3^i) = (t_2^j, t_3^j)$, then it implies that $y_2^i = y_2^j$ and $y_3^i = y_3^j$ which follows from the construction. Hence, it implies from the above equation

$$(y_2^i \oplus y_3^i) \oplus (y_2^j \oplus y_3^j) = t_1^i \oplus t_1^j$$

that $t_1^i = t_1^j$, which again implies that $y_1^i = y_1^j$ that follows from the construction. But again it implies that $m_i = m_j$ which follows from the equation

$$(y_1^i \oplus y_2^i) \oplus (y_1^j \oplus y_2^j) = m_i \oplus m_j.$$

However, if the tweaks are same for i-th and j-th query, then the corresponding message should be different as we assume non-trivial distinguisher which is violated from the above sequence of logical events. Therefore, in this case, the probability of the event is zero.

4. In all the other cases, at most one of t_1^i, t_2^i, t_3^i will collide with the corresponding t_1^j, t_2^j, t_3^j respectively. In that case we obtain two fresh random variables from each of the two equations

$$\begin{cases} (y_2^i \oplus y_3^i) \oplus (y_2^j \oplus y_3^j) = t_1^i \oplus t_1^j \\ (y_1^i \oplus y_2^i) \oplus (y_1^j \oplus y_2^j) = m_i \oplus m_j \end{cases}$$

Without loss of generality, we assume that $i < j$ and in that case we choose y_1^j as fresh random variable from the first equation and choose y_3^j as fresh random variable from the second equation. Note that we can utilize the randomness of both y_1^j and y_3^j together due to $\overline{\mathsf{Bad1}}$. Using the randomness of y_1^j and y_3^j, we bound the probability of the above event for a fixed choices of indices to at most $1/(2^n - p)^2$. By varying over all possible choices of indices and by assuming $p \le 2^{n-1}$, we have

$$\Pr[\mathsf{Bad6} \mid \overline{\mathsf{Bad2}}] \le 2q^2/2^{2n}. \tag{34}$$

Bounding $\mathsf{Bad7} \mid \overline{\mathsf{Bad2}}$: Bounding this event is exactly identical to that of $\mathsf{Bad6} \mid \overline{\mathsf{Bad2}}$ and hence, we have

$$\Pr[\mathsf{Bad7} | \overline{\mathsf{Bad2}}] \le 2q^2/2^{2n}. \tag{35}$$

We derive the bound of Lemma 6 by combining the bounds from Eqn. (29)-Eqn. (35). $\square$

We lower bound the ratio of the real to ideal interpolation probability for a good transcript. Formally, we prove the following lemma.

Lemma 7. *Let $\tau = (\tau_c, \tau_p, (y_1^i, y_2^i, y_3^i)_{i\in[q]}, k)$ be a good transcript. Let X_{re} and X_{id} be two random variables defined as above. Then, we have*

$$\frac{\Pr[\mathsf{X}_{\text{re}} = \tau]}{\Pr[\mathsf{X}_{\text{id}} = \tau]} \geq 1. \tag{36}$$

Proof Let $\tau = (\tau_c, \tau_p, (y_1^i, y_2^i, y_3^i)_{i\in[q]}, k)$ be a good transcript. Let us consider the following set:

$$\mathsf{H}_1 = \{(k, t_1^1, y_1^1), (2k, t_2^1, y_2^1), (4k, t_3^1, y_3^1), \ldots, (k, t_1^q, y_1^q), (2k, t_2^q, y_2^q), (4k, t_3^q, y_3^q)\}$$

which records the (key, input, output) triplet of the first, second and the third block cipher call of the construction across all q construction queries. For each key $K \in \{0,1\}^n$, we define the sets $\mathsf{H}_2(K) = \{(L, u, v) \in \tau_p : L = K\}$ and $\mathsf{H}_3(K) = \{(k \oplus t_1^i \oplus y_2^i \oplus y_3^i, m_i \oplus y_1^i \oplus y_2^i, c_i \oplus y_1^i \oplus y_2^i) : k \oplus t_1^i \oplus y_2^i \oplus y_3^i = K\}$, where $(t_1^i \| t_2^i \| t_3^i, m_i, c_i) \in \tau_c$. Note that, $\mathsf{H}_2(K)$ denotes the set of (key, input, output) triplet across all p ideal-cipher queries such that the key is K. Similarly, $\mathsf{H}_3(K)$ denotes the set of all triplet of (key, input, output) of the third block cipher call of the construction across all q construction queries such that the key of the third block cipher call is K. For each tweak $t \in \{0,1\}^{3n}$, we define the set

$$\mathsf{H}(t) = \{(t_1^i \| t_2^i \| t_3^i, m_i, c_i) \in \tau_c : t_1^i \| t_2^i \| t_3^i = t\}$$

which records all q triplet of tweak, queries and response excluding the block cipher key such that the tweak of the construction query is t. Finally, for each key $K \in \{0,1\}^n$, we define the set

$$\mathsf{Z}(K) = \{(t_1^i \| t_2^i \| t_3^i) : (t_1^i \| t_2^i \| t_3^i, m_i, c_i) \in \tau_c \wedge k \oplus t_1^i \oplus y_2^i \oplus y_3^i = K\}.$$

Since the transcript is good, we have $\mathsf{H}_1 \cap \mathsf{H}_2(K) = \emptyset,\ \mathsf{H}_1 \cap \mathsf{H}_3(K) = \emptyset,\ \mathsf{H}_2(K) \cap \mathsf{H}_3(K) = \emptyset$, for each $K \in \{0,1\}^n$. These follow directly from $\overline{\mathsf{Bad2}}$, $\overline{\mathsf{Bad3}}$, and $\overline{\mathsf{Bad4}} \wedge \overline{\mathsf{Bad5}}$.
Let us fix a key $K \in \{0,1\}^n$. For each $t \in \mathsf{Z}(K)$, $|\mathsf{H}(t)|$ denotes the number of construction queries with tweak t. Due to $\overline{\mathsf{Bad6}} \wedge \overline{\mathsf{Bad7}}$, we have for each key $K \in \{0,1\}^n$,

$$\sum_{t \in \mathsf{Z}(K)} |\mathsf{H}(t)| = |\mathsf{H}_3(K)|.$$

For the sake of simplicity, let us denote $|\mathsf{H}_1| = \alpha_1$. For each $K \in \{0,1\}^n$, we denote $|\mathsf{H}_2(K)| = \alpha_2(K)$, $|\mathsf{H}_3(K)| = \alpha_3(K)$ and for each tweak $t \in \{0,1\}^{3n}$, we denote $|\mathsf{H}(t)| = \alpha(t)$. Therefore, for a fixed good transcript τ, the ideal interpolation probability becomes

$$\Pr[\mathsf{X}_{\text{id}} = \tau] \quad \leq \quad \frac{1}{2^n} \cdot \prod_{i=0}^{\alpha_1 - 1} \frac{1}{2^n - i} \cdot \Bigg(\prod_{K \in \{0,1\}^n} \prod_{j=0}^{\alpha_2(K)-1} \frac{1}{2^n - j} \prod_{p=0}^{\alpha_3(K)-1} \frac{1}{2^n - p} \Bigg).$$

To bound the real interpolation probability, the number of times block cipher is called for deriving sub-keys is α_1. However, the number of times block cipher is called for ideal-cipher queries and construction queries is $\alpha_2(K) + \alpha_3(K)$ for each key $K \in \{0,1\}^n$. Therefore, we have

$$\Pr[\mathsf{X}_{\text{re}} = \tau] = \frac{1}{2^n} \cdot \prod_{i=0}^{\alpha_1 - 1} \frac{1}{2^n - i} \cdot \Bigg(\prod_{K \in \{0,1\}^n} \prod_{j=0}^{\alpha_2(K)+\alpha_3(K)-1} \frac{1}{2^n - j} \Bigg).$$

Since for each key $K \in \{0,1\}^n$, we have

$$\prod_{j=0}^{\alpha_2(K)-1} \frac{1}{2^n - j} \prod_{p=0}^{\alpha_3(K)-1} \frac{1}{2^n - p} \leq \prod_{j=0}^{\alpha_2(K)+\alpha_3(K)-1} \frac{1}{2^n - j},$$

the ratio of the real to ideal interpolation probability becomes ≥ 1, which proves the result. □
Finally, Theorem 3 follows by combining Lemma 6 and Lemma 7. □

6 Optimally Secure TBC with rn-bit Tweaks Using $(r+1)$ Block Cipher Calls

Let $\mathsf{E} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a n-bit block cipher and $r \in \mathbb{N}$ be a given parameter. The tweakable block cipher $\widetilde{\mathsf{G}}r : \{0,1\}^n \times \{0,1\}^{rn} \times \{0,1\}^n \to \{0,1\}^n$ with a rn-bit tweak is constructed using $(r+1)$ block ciphers as follows: r block cipher calls are first invoked in parallel to produce r sub-keys $y_1, y_2, \ldots, y_r$ from the tweaks $t_1, t_2, \ldots, t_r$ and the master key k as shown in Fig. 15. Then, the subkeys are linearly combined to generate two n-bit strings Y and Z which are used to compute the ciphertext for a given message m as shown in Fig. 15. In the following we show that $\widetilde{\mathsf{G}_r}$ is a secure tweakable block cipher with rn bit

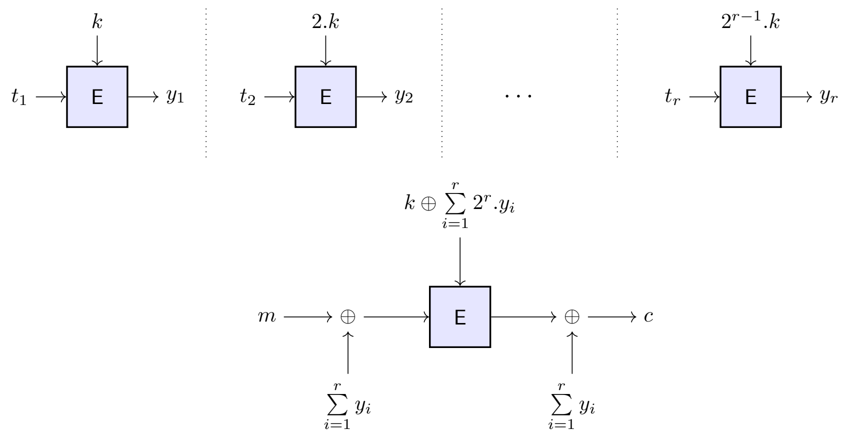


Figure 15: $\widetilde{\mathsf{G}}r$ construction: TBC with rn-bit tweaks using $(r+1)$ block cipher calls.

tweaks against all adversaries that makes roughly 2^n construction and ideal-cipher queries. Formally, we have the following result:

Theorem 4. *Let $\mathcal{A}$ be an adversary making at most q construction queries and p ideal-cipher queries including both forward and backward queries. Then,*

$$\mathbf{Adv}^{\text{tsprp-icm}}_{\widetilde{\mathsf{G}}r}(\mathcal{A}) \leq \frac{4q(p+q)}{2^{2n}} + \frac{2rq + rp + 1}{2^n}.$$

Proof. Let $\tau_c = \{(t_1^1 \| t_2^1 \| \ldots \| t_r^1, m_1, c_1), \ldots, (t_1^q \| t_2^q \| \ldots \| t_r^q, m_q, c_q)\}$ and $\tau_p = \{(L_1, u_1, v_1), (L_2, u_2, v_2), \ldots, (L_p, u_p, v_p)\}$ denotes the list of construction query-responses and ideal-cipher query-responses of $\mathcal{A}$ respectively. After the interaction, the real world oracle releases the block cipher key k and the tuple of sub-keys $(y_1^i, y_2^i, \ldots, y_r^i), i \in [q]$ tuple, whereas the ideal world oracle randomly samples n-bit dummy key k and computes the sub-key tuple $(y_1^i, y_2^i, \ldots, y_r^i), i \in [q]$, where $y_1^i, y_2^i, \ldots, y_r^i$ are computed similar to the real world and finally released them to the distinguisher. Therefore, the extended transcript of the attack is $\tau = (\tau_c, \tau_p, (y_1^i, y_2^i, \ldots, y_r^i)_{i \in [q]}, k)$.

6.1 Definition of Bad Transcript and Bounding its Probability

Let Θ denote the set of all attainable transcripts. We call an attainable transcript $\tau \in \Theta$ is bad if it satisfies either of the following:

1. $\mathsf{Bad1}$: $k = 0$.
2. $\mathsf{Bad2}$: $\exists \alpha \in [p] : L_\alpha \in \{k, 2k, 2^2k, \ldots, 2^{r-1}k\}$.
3. $\mathsf{Bad3}$: $\exists i \in [q] : k \oplus Z \in \{k, 2k, 2^2k, \ldots, 2^{r-1}k\}$.
4. $\mathsf{Bad4}$: $\exists i \in [q], \alpha \in [p] : m_i \oplus Y_i = u_\alpha,\ k \oplus Z_i = L_\alpha$.
5. $\mathsf{Bad5}$: $\exists i \in [q], \alpha \in [p] : c_i \oplus Y_i = v_\alpha,\ k \oplus Z_i = L_\alpha$.
6. $\mathsf{Bad6}$: $\exists i \neq j \in [q] : Y_i \oplus Y_j = m_i \oplus m_j,\ Z_i = Z_j$.
7. $\mathsf{Bad7}$: $\exists i \neq j \in [q] : Y_i \oplus Y_j = c_i \oplus c_j,\ Z_i = Z_j$.

Lemma 8. *Let Θ_{b} denote the set of all bad transcripts and recall that X_{id} denotes the random variable of transcript τ induced in the ideal world. Then, we have the following:*

$$\Pr[\mathsf{X}_{\mathrm{id}} \in \Theta_{\mathrm{b}}] \leq \frac{4q(p+q)}{2^{2n}} + \frac{2rq + rp + 1}{2^n}. \tag{37}$$

Proof Let us denote $\mathsf{Bad} = \mathsf{Bad1} \vee \mathsf{Bad2} \vee (\vee_{i=3}^{7} \mathsf{Badi} \mid \overline{\mathsf{Bad2}})$. Therefore, by applying the union bound, we have

$$\Pr[\mathsf{Bad}] \leq \Pr[\mathsf{Bad1}] + \Pr[\mathsf{Bad2}] + \sum_{i=3}^{7} \Pr[\mathsf{Badi} \mid \overline{\mathsf{Bad2}}].$$

Therefore, to bound the probability of the event Bad, we individually bound the probability of the event $\mathsf{Bad1}, \mathsf{Bad2}$ and Badi for $3 \leq i \leq 7$ conditioned on the complement of the event $\mathsf{Bad2}$. Then we apply the union bound to obtain the final result.

Bounding $\mathsf{Bad1}$: Bounding this event is exactly identical to that of bounding $\mathsf{Bad1}$ in Lemma 6. Thus, we have

$$\Pr[\mathsf{Bad1}] \leq 1/2^n. \tag{38}$$

Bounding $\mathsf{Bad2}$: Bounding this event is again very similar to that of bounding $\mathsf{Bad2}$ in Lemma 6. For a fixed index $\alpha \in [p]$, and for a fixed $i \in [r]$ the probability of the event $2^{i-1}k = L_\alpha$ is upper bounded by $1/2^n$ due to the randomness of the key k. By varying over all possible choices of indices $\alpha \in [p]$ and $i \in [r]$, we have

$$\Pr[\mathsf{Bad2}] \leq rp/2^n. \tag{39}$$

Bounding $\mathsf{Bad3} \mid \overline{\mathsf{Bad2}}$: For a fixed choice of index $i \in [q]$, and for a fixed $\alpha \in [r]$, we bound the event $k \oplus 2^r y_1^i \oplus 2^{r-1} y_2^i \oplus \ldots \oplus 2y_r^i = 2^{\alpha-1}k$, which boils down to the event

$$2^r y_1^i \oplus 2^{r-1} y_2^i \oplus \ldots \oplus 2y_r^i = k(1 \oplus 2^{\alpha-1}).$$

By the virtue of $\overline{\mathsf{Bad2}}$ event, the random variable y_1^i is fresh. Hence, we bound the probability of the event to at most $1/2^n - p$ using the randomness of y_1^i. Therefore, by varying over all possible choices of indices and by assuming $p \leq 2^{n-1}$, we have

$$\Pr[\mathsf{Bad3} \mid \overline{\mathsf{Bad2}}] \leq 2rq/2^n. \tag{40}$$

Bounding $\mathsf{Bad4} \mid \overline{\mathsf{Bad2}}$: For a fixed choice of indices $i \in [q], \alpha \in [p]$, the probability of the event

$$\begin{cases} y_1^i \oplus y_2^i \oplus \ldots \oplus y_r^i = m_i \oplus u_\alpha \\ k \oplus 2^r y_1^i \oplus 2^{r-1} y_2^i \oplus \ldots \oplus 2y_r^i = L_\alpha \end{cases}$$

is upper bounded by $1/2^n \cdot 1/(2^n - p)$ due to the randomness of the key k and y_1^i as y_1^i is not determined by ideal-cipher query due to the virtue of $\overline{\mathsf{Bad2}}$. By varying over all possible choices of indices $i \in [q], \alpha \in [p]$ and by assuming $p \leq 2^{n-1}$, we have

$$\Pr[\mathsf{Bad4} \mid \overline{\mathsf{Bad2}}] \leq 2qp/2^{2n}. \tag{41}$$

Bounding $\mathsf{Bad5} \mid \overline{\mathsf{Bad2}}$: Bounding this event is identical to that of $\mathsf{Bad4} \mid \overline{\mathsf{Bad2}}$ and hence we have

$$\Pr[\mathsf{Bad5} \mid \overline{\mathsf{Bad2}}] \leq 2qp/2^{2n}. \tag{42}$$

Bounding $\mathsf{Bad6} \mid \overline{\mathsf{Bad2}}$: For a fixed choice of indices $i \neq j \in [q]$, the probability of the event

$$\begin{cases} (y_1^i \oplus y_2^i \oplus \ldots \oplus y_r^i) \oplus (y_1^j \oplus y_2^j \oplus y_r^j) = m_i \oplus m_j \\ 2^r(y_1^i \oplus y_1^j) \oplus 2^{r-1}(y_2^i \oplus y_2^j) \oplus \ldots \oplus 2(y_r^i \oplus y_r^j) = 0^n \end{cases}$$

Let $\mathsf{EQ} = \{\alpha_1, \alpha_2, \ldots, \alpha_s\} \subseteq [r]$ such that $t_{\alpha_1}^i = t_{\alpha_1}^j$, $t_{\alpha_2}^i = t_{\alpha_2}^j$, $\ldots t_{\alpha_s}^i = t_{\alpha_s}^j$. Therefore, we have

$$\begin{aligned} m^i \oplus m^j &= \oplus_{\alpha \in [r] \setminus \mathsf{EQ}} (y_\alpha^i \oplus y_\alpha^j) \\ 0^n &= \oplus_{\alpha \in [r] \setminus \mathsf{EQ}} 2^{r-\alpha+1} (y_\alpha^i \oplus y_\alpha^j) \end{aligned}$$

It is easy to see that $|\mathsf{EQ}| < r - 1$, otherwise the probability of the above events would have been zero. Therefore, we assume that $|\mathsf{EQ}| \leq r - 2$. Hence, we get at least two fresh random variables $y_{\alpha_1}^i, y_{\alpha_2}^i$, where $\alpha_1, \alpha_2 \in [r] \setminus \mathsf{EQ}$ by the virtue of $\overline{\mathsf{Bad2}}$. Since, the above system of equations is of rank 2, therefore, by using the randomness of $y_{\alpha_1}^i$ and $y_{\alpha_2}^i$, we upper bound the probability of the above event to $1/(2^n - p)^2$. By varying all possible choices of indices $i \neq j \in [q]$ and by assuming $p \leq 2^{n-1}$, we have

$$\Pr[\mathsf{Bad6} \mid \overline{\mathsf{Bad2}}] \leq 2q^2/2^{2n}. \tag{43}$$

Bounding $\mathsf{Bad7} \mid \overline{\mathsf{Bad2}}$: Bounding this event is exactly identical to that of $\mathsf{Bad6} \mid \overline{\mathsf{Bad2}}$ and hence, we have

$$\Pr[\mathsf{Bad7} | \overline{\mathsf{Bad2}}] \leq 2q^2/2^{2n}. \tag{44}$$

We derive the bound of Lemma 8 by combining the bounds from Eqn. (38)-Eqn. (44). □

6.2 Good Transcript Analysis

In this section, we lower bound the ratio of the real to ideal interpolation probability for a good transcript. Formally, we prove the following lemma.

Lemma 9. *Let $\tau = (\tau_c, \tau_p, (y_1^i, y_2^i, \ldots, y_r^i)_{i \in [q]}, k)$ be a good transcript. Let X_{re} and X_{id} be two random variables defined as above. Then, we have*

$$\frac{\Pr[\mathsf{X}_{\mathrm{re}} = \tau]}{\Pr[\mathsf{X}_{\mathrm{id}} = \tau]} \geq 1. \tag{45}$$

Proof Let $\tau = (\tau_c, \tau_p, (y_1^i, y_2^i, \ldots, y_r^i)_{i\in[q]}, k)$ be a good transcript. Let us consider the following set:

$\mathsf{H}_1 = \{(k, t_1^1, y_1^1), (2k, t_2^1, y_2^1), \ldots, (2^{r-1}k, t_r^1, y_r^1), \ldots, (k, t_1^q, y_1^q), (2k, t_2^q, y_2^q), \ldots, (2^{r-1}k, t_r^q, y_r^q))$

which records the (key, input, output) triplet of the r many block cipher call in the sub-key derivation phase of the construction across all q construction queries. For each key $K \in \{0,1\}^n$, we define the sets $\mathsf{H}_2(K) = \{(L, u, v) \in \tau_p : L = K\}$ and $\mathsf{H}_3(K) = \{(k \oplus 2^r y_1^i \oplus 2^{r-1} y_2^i \oplus \ldots \oplus 2y_r^i, m_i \oplus y_1^i \oplus y_2^i \oplus \ldots \oplus y_r^i, c_i \oplus y_1^i \oplus y_2^i \oplus \ldots \oplus y_r^i) : k \oplus 2^r y_1^i \oplus 2^{r-1} y_2^i \oplus \ldots \oplus 2y_r^i = K\}$, where $(t_1^i \| t_2^i \| \ldots \| t_r^i, m_i, c_i) \in \tau_c$. For each tweak $t \in \{0,1\}^{rn}$, we define the set

$$\mathsf{H}(t) = \{(t_1^i \| t_2^i \| \ldots \| t_r^i, m_i, c_i) \in \tau_c : t_1^i \| t_2^i \| \ldots \| t_r^i = t\}$$

which records all q triplet of tweak, queries and response excluding the block cipher key such that the tweak of the construction query is t. Finally, for each key $K \in \{0,1\}^n$, we define the set

$$\mathsf{Z}(K) = \{(t_1^i \| t_2^i \| \ldots \| t_r^i) : (t_1^i \| t_2^i \| \ldots \| t_r^i, m_i, c_i) \in \tau_c \wedge k \oplus 2^r y_1^i \oplus 2^{r-1} y_2^i \oplus \ldots \oplus 2y_r^i = K\}.$$

Since the transcript is good, we have $\mathsf{H}_1 \cap \mathsf{H}_2(K) = \emptyset,\ \mathsf{H}_1 \cap \mathsf{H}_3(K) = \emptyset,\ \mathsf{H}_2(K) \cap \mathsf{H}_3(K) = \emptyset$, for each $K \in \{0,1\}^n$. These follow directly from $\overline{\mathsf{Bad2}}$, $\overline{\mathsf{Bad3}}$, and $\overline{\mathsf{Bad4}} \wedge \overline{\mathsf{Bad5}}$.

Let us fix a key $K \in \{0,1\}^n$. For each $t \in \mathsf{Z}(K)$, $|\mathsf{H}(t)|$ denotes the number of construction queries with tweak t. Due to $\overline{\mathsf{Bad6}} \wedge \overline{\mathsf{Bad7}}$, we have for each key $K \in \{0,1\}^n$,

$$\sum_{t \in \mathsf{Z}(K)} |\mathsf{H}(t)| = |\mathsf{H}_3(K)|.$$

For the sake of simplicity, let us denote $|\mathsf{H}_1| = \alpha_1$. For each $K \in \{0,1\}^n$, we denote $|\mathsf{H}_2(K)| = \alpha_2(K)$, $|\mathsf{H}_3(K)| = \alpha_3(K)$ and for each tweak $t \in \{0,1\}^{rn}$, we denote $|\mathsf{H}(t)| = \alpha(t)$. Therefore, for a fixed good transcript τ, the ideal interpolation probability becomes

$$\begin{aligned}
\Pr[\mathsf{X}_{\mathrm{id}} = \tau] &= \frac{1}{2^n} \cdot \prod_{i=0}^{\alpha_1 - 1} \frac{1}{2^n - i} \cdot \Big(\prod_{K \in \{0,1\}^n} \prod_{j=0}^{\alpha_2(K)-1} \frac{1}{2^n - j} \Big) \\
&\qquad \cdot \Big(\prod_{K \in \{0,1\}^n} \prod_{t \in \mathsf{Z}(K)} \prod_{p=0}^{\alpha(t)-1} \frac{1}{2^n - p} \Big) \\
&\leq \frac{1}{2^n} \cdot \prod_{i=0}^{\alpha_1 - 1} \frac{1}{2^n - i} \cdot \Big(\prod_{K \in \{0,1\}^n} \prod_{j=0}^{\alpha_2(K)-1} \frac{1}{2^n - j} \prod_{p=0}^{\alpha_3(K)-1} \frac{1}{2^n - p} \Big).
\end{aligned}$$

To bound the real interpolation probability, the number of times block cipher is called for deriving sub-keys is α_1. However, the number of times block cipher is called for ideal-cipher queries and construction queries is $\alpha_2(K) + \alpha_3(K)$ for each key $K \in \{0,1\}^n$. Therefore, we have

$$\Pr[\mathsf{X}_{\mathrm{re}} = \tau] = \frac{1}{2^n} \cdot \prod_{i=0}^{\alpha_1 - 1} \frac{1}{2^n - i} \cdot \Big(\prod_{K \in \{0,1\}^n} \prod_{j=0}^{\alpha_2(K)+\alpha_3(K)-1} \frac{1}{2^n - j} \Big).$$

Since for each key $K \in \{0,1\}^n$, we have

$$\prod_{j=0}^{\alpha_2(K)-1} \frac{1}{2^n - j} \prod_{p=0}^{\alpha_3(K)-1} \frac{1}{2^n - p} \leq \prod_{j=0}^{\alpha_2(K)+\alpha_3(K)-1} \frac{1}{2^n - j},$$

the ratio of the real to ideal interpolation probability becomes ≥ 1, which proves the result. $\square$

Finally, Theorem 4 follows by combining Lemma 8 and Lemma 9. $\square$

7 Software Implementation and Benchmarking Setup

We now describe the setup used for our software implementation and performance evaluation. The benchmarking is conducted on an 11th Gen Intel(R) Core(TM) i7-1165G7 processor, equipped with a 2.80GHz CPU, 32 GB of DDR5 RAM, and hardware support for AES-NI instructions. The underlying block cipher used in all experiments is AES-128 [DR02], which benefits from efficient implementation using AES-NI.

To ensure consistency in our performance measurements, Intel TurboBoost and HyperThreading are disabled during testing. The implementation is compiled using GCC 12.2.0 with the `-O3` optimization flag enabled and `-march` set for architecture-specific tuning. The testing environment consists of Debian 12.6 running on Linux kernel 6.1.0. Performance measurements are taken for various message sizes, with the cost of key setup explicitly excluded. Each encryption experiment is repeated 128 times to obtain accurate and reliable results, and cache warming is performed before starting the measurements.

We also discuss how domain separation is handled within the ZMAC framework. As specified in the ZMAC construction [IMPS17b], domain separation requires a total of 10 distinct instances of the TBC. These instances of the tweakable block ciphers (TBC) are denoted as $\widetilde{\mathsf{E}}^i(K,T)$ for $i = 0, \ldots, 9$.

To derive each instance, the i-th domain separation constant is computed by encrypting i encoded as a 128-bit value, using the master key and an all-zero tweak, yielding the value D_i. The domain-separated instances are then defined as:

$$\widetilde{\mathsf{E}}^i(K,T,x) = \widetilde{\mathsf{E}}(K,T,x \oplus D_i) \text{ for } i = 0, \ldots, 9.$$

This approach follows the domain separation strategy outlined in [IMPS17b], ensuring that each instance operates independently while maintaining efficiency and security.

8 Conclusion

In this paper, we have studied the design of tweakable block ciphers with $3n$-bit tweaks from n-bit block ciphers. Although $\widetilde{\mathsf{G}}3^*$ achieves $2n/3$-bit security, it is not known if the bound is tight. Hence, it appears to be a challenging problem to explore tightness of the bound. In fact, it remains an open to determine if the construction achieves n-bit security. A more general and pertinent question would be to find the minimum number of block cipher calls required to design a tweakable block cipher with dn-bit tweaks to achieve n-bit security.

References

[ABD$^+$23] Roberto Avanzi, Subhadeep Banik, Orr Dunkelman, Maria Eichlseder, Shibam Ghosh, Marcel Nageler, and Francesco Regazzoni. The qarmav2 family of tweakable block ciphers. *IACR Trans. Symmetric Cryptol.*, 2023(3):25–73, 2023.

[Bab02] László Babai. The fourier transform and equations over finite abelian groups (lec- ture notes, version 1.3). 2002.

[BJK$^+$16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA,* *USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016.

[BLN18] Ritam Bhaumik, Eik List, and Mridul Nandi. Zcz – achieving n-bit sprp security with a minimal number of tweakable-block-cipher calls. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018*, pages 336–366, Cham, 2018. Springer International Publishing.

[BN15] Ritam Bhaumik and Mridul Nandi. An inverse-free single-keyed tweakable enciphering scheme. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 159–180. Springer, 2015.

[CDK23] Debrup Chakraborty, Avijit Dutta, and Samir Kundu. Designing tweakable enciphering schemes using public permutations. *Adv. Math. Commun.*, 17(4):771–798, 2023.

[CEL+21] Benoît Cogliati, Jordan Ethan, Virginie Lallemand, ByeongHak Lee, Jooyoung Lee, and Marine Minier. CTET+: A beyond-birthday-bound secure tweakable enciphering scheme using a single pseudorandom permutation. *IACR Trans. Symmetric Cryptol.*, 2021(4):1–35, 2021.

[CIL+20] Wonseok Choi, Akiko Inoue, ByeongHak Lee, Jooyoung Lee, Eik List, Kazuhiko Minematsu, and Yusuke Naito. Highly secure nonce-based macs from the sum of tweakable block ciphers. *IACR Trans. Symmetric Cryptol.*, 2020(4):39–70, 2020.

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

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

[CS06] Debrup Chakraborty and Palash Sarkar. A general construction of tweakable block ciphers and different modes of operations. In Helger Lipmaa, Moti Yung, and Dongdai Lin, editors, *Information Security and Cryptology, Second SKLOIS Conference, Inscrypt 2006, Beijing, China, November 29 - December 1, 2006, Proceedings*, volume 4318 of *Lecture Notes in Computer Science*, pages 88–102. Springer, 2006.

[CS08a] Debrup Chakraborty and Palash Sarkar. A general construction of tweakable block ciphers and different modes of operations. *IEEE Trans. Inf. Theory*, 54(5):1991–2006, 2008.

[CS08b] Debrup Chakraborty and Palash Sarkar. HCH: A new tweakable enciphering scheme using the hash-counter-hash approach. *IEEE Trans. Inf. Theory*, 54(4):1683–1699, 2008.

[CS14] Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 327–350. Springer, 2014.

[DDDM23] Nilanjan Datta, Shreya Dey, Avijit Dutta, and Sougata Mandal. Cascading four round LRW1 is beyond birthday bound secure. *IACR Trans. Symmetric Cryptol.*, 2023(4):365–390, 2023.

[DDLM23] Nilanjan Datta, Avijit Dutta, Eik List, and Sougata Mandal. On the security of triplex- and multiplex-type constructions with smaller tweaks. In Anupam Chattopadhyay, Shivam Bhasin, Stjepan Picek, and Chester Rebeiro, editors, *Progress in Cryptology - INDOCRYPT 2023 - 24th International Conference on Cryptology in India, Goa, India, December 10-13, 2023, Proceedings, Part I*, volume 14459 of *Lecture Notes in Computer Science*, pages 25–47. Springer, 2023.

[DN18] Avijit Dutta and Mridul Nandi. Tweakable HCTR: A BBB secure tweakable enciphering scheme. In Debrup Chakraborty and Tetsu Iwata, editors, *Progress in Cryptology - INDOCRYPT 2018 - 19th International Conference on Cryptology in India, New Delhi, India, December 9-12, 2018, Proceedings*, volume 11356 of *Lecture Notes in Computer Science*, pages 47–69. Springer, 2018.

[DR02] Joan Daemen and Vincent Rijmen. AES and the Wide Trail Design Strategy. In *Advances in Cryptology – Proceedings of EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 108–109. Springer, 2002.

[Dwo10] Morris Dworkin. Recommendation for block cipher modes of operation: the xts-aes mode for confidentiality on storage devices. *NIST SP 800-38E*, January 2010.

[FLS+] Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. Skein. *SHA-3 submission to NIST*, 2006 - 2012.

[GIK+] Chun Guo, Tetsu Iwata, Mustafa Khairallah, Kazuhiko Minematsu, and Thomas Peyrin. Romulus v1.3. *Submission to NIST Lightweight Cryptography Standardization Process*, 2018 - 2023. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/romulus-spec-final.pdf.

[GLS+] Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, Kerem Varici, Anthony Journault, François Durvaux, Lubos Gaspar, and Stéphanie Kerckhof. Scream. *Submission to CAESAR Competition*, 2013 - 2019. https://competitions.cr.yp.to/round2/screamv3.pdf.

[Hal04] Shai Halevi. EME*: Extending EME to handle arbitrary-length messages with associated data. In Anne Canteaut and Kapalee Viswanathan, editors, *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, volume 3348 of *Lecture Notes in Computer Science*, pages 315–327. Springer, 2004.

[Hir22] Shoichi Hirose. Collision-resistant and pseudorandom hash function using tweakable block cipher. In Ilsun You and Taek-Young Youn, editors, *Information Security Applications - 23rd International Conference, WISA 2022, Jeju Island, South Korea, August 24-26, 2022, Revised Selected Papers*, volume 13720 of *Lecture Notes in Computer Science*, pages 3–15. Springer, 2022.

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 15–44. Springer, 2015.

[HR03] Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 482–499. Springer, 2003.

[HR04] Shai Halevi and Phillip Rogaway. A parallelizable enciphering mode. In Tatsuaki Okamoto, editor, *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, volume 2964 of *Lecture Notes in Computer Science*, pages 292–304. Springer, 2004.

[IKMP20] Tetsu Iwata, Mustafa Khairallah, Kazuhiko Minematsu, and Thomas Peyrin. Duel of the titans: The romulus and remus families of lightweight AEAD algorithms. *IACR Trans. Symmetric Cryptol.*, 2020(1):43–120, 2020.

[IMPS17a] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A fast tweakable block cipher mode for highly secure message authentication. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 34–65. Springer, 2017.

[IMPS17b] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A fast tweakable block cipher mode for highly secure message authentication. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 34–65. Springer, 2017.

[JKNS23] Ashwin Jha, Mustafa Khairallah, Mridul Nandi, and Abishanka Saha. Tight security of TNT and beyond: Attacks, proofs and possibilities for the cascaded LRW paradigm. *IACR Cryptol. ePrint Arch.*, page 1272, 2023.

[JLM+17] Ashwin Jha, Eik List, Kazuhiko Minematsu, Sweta Mishra, and Mridul Nandi. XHX - A framework for optimally secure tweakable block ciphers from classical block ciphers and universal hashing. In Tanja Lange and Orr Dunkelman, editors, *Progress in Cryptology - LATINCRYPT 2017 - 5th International Conference on Cryptology and Information Security in Latin America, Havana, Cuba, September 20-22, 2017, Revised Selected Papers*, volume 11368 of *Lecture Notes in Computer Science*, pages 207–227. Springer, 2017.

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

[JNPa] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Joltik v1.3. *Submission to CAESAR Competition*, 2013 - 2019. https://competitions.cr.yp.to/round2/joltikv13.pdf.

[JNPb] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Kiasu v1. *Submission to CAESAR Competition*, 2013 - 2019. https://competitions.cr.yp.to/round1/kiasuv1.pdf.

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

[LL18] ByeongHak Lee and Jooyoung Lee. Tweakable block ciphers secure beyond the birthday bound in the ideal cipher model. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 305–335. Springer, 2018.

[LRW02] Moses D. Liskov, Ronald L. Rivest, and David A. Wagner. Tweakable block ciphers. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2002.

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

[LS13] Rodolphe Lampe and Yannick Seurin. Tweakable blockciphers with asymptotically optimal security. In Shiho Moriai, editor, *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, volume 8424 of *Lecture Notes in Computer Science*, pages 133–151. Springer, 2013.

[LST12] Will Landecker, Thomas Shrimpton, and R. Seth Terashima. Tweakable blockciphers with beyond birthday-bound security. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 14–30. Springer, 2012.

[Men15a] Bart Mennink. Optimally secure tweakable blockciphers. In Gregor Leander, editor, *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, volume 9054 of *Lecture Notes in Computer Science*, pages 428–448. Springer, 2015.

[Men15b] Bart Mennink. Optimally secure tweakable blockciphers. Cryptology ePrint Archive, Paper 2015/363, 2015.

[MI15] Kazuhiko Minematsu and Tetsu Iwata. Tweak-length extension for tweakable blockciphers. In Jens Groth, editor, *Cryptography and Coding - 15th IMA International Conference, IMACC 2015, Oxford, UK, December 15-17, 2015. Proceedings*, volume 9496 of *Lecture Notes in Computer Science*, pages 77–93. Springer, 2015.

[Min06] Kazuhiko Minematsu. Improved security analysis of XEX and LRW modes. In Eli Biham and Amr M. Youssef, editors, *Selected Areas in Cryptography, 13th International Workshop, SAC 2006, Montreal, Canada, August 17-18, 2006 Revised Selected Papers*, volume 4356 of *Lecture Notes in Computer Science*, pages 96–113. Springer, 2006.

[MM07] Kazuhiko Minematsu and Toshiyasu Matsushima. Tweakable enciphering schemes from hash-sum-expansion. In K. Srinathan, C. Pandu Rangan, and Moti Yung, editors, *Progress in Cryptology - INDOCRYPT 2007, 8th International Conference on Cryptology in India, Chennai, India, December 9-13, 2007, Proceedings*, volume 4859 of *Lecture Notes in Computer Science*, pages 252–267. Springer, 2007.

[Nai15] Yusuke Naito. Full prf-secure message authentication code based on tweakable block cipher. In Man Ho Au and Atsuko Miyaji, editors, *Provable Security - 9th International Conference, ProvSec 2015, Kanazawa, Japan, November 24-26, 2015, Proceedings*, volume 9451 of *Lecture Notes in Computer Science*, pages 167–182. Springer, 2015.

[Nai19] Yusuke Naito. A highly secure MAC from tweakable blockciphers with support for short tweaks. In Julian Jang-Jaccard and Fuchun Guo, editors, *Information Security and Privacy - 24th Australasian Conference, ACISP 2019, Christchurch, New Zealand, July 3-5, 2019, Proceedings*, volume 11547 of *Lecture Notes in Computer Science*, pages 588–606. Springer, 2019.

[NI22] Kazuki Nakaya and Tetsu Iwata. Generalized feistel structures based on tweakable block ciphers. *IACR Transactions on Symmetric Cryptology*, 2022(4):24–91, Dec. 2022.

[NIS00] NIST Competition for Advanced Encryption Standard (AES), 1997 - 2000. https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development.

[NSS20] Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Lightweight authenticated encryption mode suitable for threshold implementation. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 705–735. Springer, 2020.

[NSS22] Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Secret can be public: Low-memory AEAD mode for high-order masking. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 315–345. Springer, 2022.

[Pat08] Jacques Patarin. The "coefficients h" technique. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, volume 5381 of *Lecture Notes in Computer Science*, pages 328–345. Springer, 2008.

[PSS23] Thomas Peters, Yaobin Shen, and François-Xavier Standaert. Multiplex: TBC-based Authenticated Encryption with Sponge-Like Rate. http://hdl.handle.net/2078.1/273131, 2023.

[Rog04] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory* *and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, volume 3329 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2004.

[Sar09] Palash Sarkar. Efficient tweakable enciphering schemes from (block-wise) universal hash functions. *IEEE Trans. Inf. Theory*, 55(10):4749–4760, 2009.

[Sar11] Palash Sarkar. Tweakable enciphering schemes using only the encryption function of a block cipher. *Inf. Process. Lett.*, 111(19):945–955, 2011.

[Sch] Richard Schroeppel. The hasty pudding cipher. *AES submission to NIST*, 1997 - 2000.

[SPS+22] 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.

[SS23] Yaobin Shen and François-Xavier Standaert. Optimally secure tweakable block ciphers with a large tweak from n-bit block ciphers. *IACR Trans. Symmetric Cryptol.*, 2023(2):47–68, 2023.

[Wan] Lei Wang. Shell v2.0. *Submission to CAESAR Competition*, 2013 - 2019. https://competitions.cr.yp.to/round2/shellv20.pdf.

[WFW05] Peng Wang, Dengguo Feng, and Wenling Wu. HCTR: A variable-input-length enciphering mode. In Dengguo Feng, Dongdai Lin, and Moti Yung, editors, *Information Security and Cryptology, First SKLOIS Conference, CISC 2005, Beijing, China, December 15-17, 2005, Proceedings*, volume 3822 of *Lecture Notes in Computer Science*, pages 175–188. Springer, 2005.

[WGZ+16] Lei Wang, Jian Guo, Guoyan Zhang, Jingyuan Zhao, and Dawu Gu. How to build fully secure tweakable blockciphers from classical blockciphers. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 455–483, 2016.

[ZQG23] 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.

Appendix

A Proof of Combinatorial Results

A.1 Proof of Lemma 2

We need to show that at least one of the following conditions is true:

1. There exists $t^1, t^2 \in \{0,1\}^{3n}$ such that $f_s(t^1) = f_s(t^2), \forall\ s \in \{1,2,3,4\}$.

2. There exists $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ satisfying $f_1(t^i) = f_1(t^j),\ f_3(t^i) \neq f_3(t^j)$, for all $i, j \in \{1, 2, \ldots, 2^{n/2}\}$.

3. There exists $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ satisfying $f_1(t^i) = f_1(t^j),\ f_2(t^i) \neq f_2(t^j),\ f_4(t^i) \neq f_4(t^j)$, for all $i, j \in \{1, 2, \ldots, 2^{n/2}\}$.

Let us consider the function $f_1 : \{0,1\}^{3n} \to \{0,1\}^n$. By the Pigeonhole Principle, there exist at least 2^{2n} distinct inputs that map the function f_1 to some fixed value, say a. Define the set $A_a := \{t \in \{0,1\}^{3n} : f_1(t) = a\}$. By definition, $|A_a| \geq 2^{2n}$. Now, we consider the two cases:

Case 1: f_3 is solely dependent on f_1. In this case, we have $f_3(t) = c, \forall t \in A_a$, for some $c \in \{0,1\}^n$. Consider the sets $B_b := \{t \in A_a : f_2(t) = b\}, \forall b \in \{0,1\}^n$. If there exists some b such that $|B_b| \geq 2^n + 1$, then we can apply the Pigeonhole Principle to conclude there exists t^1 and t^2 with $f_4(t^1) = f_4(t^2)$, and hence satisfies condition 1. Otherwise, for each $b \in \{0,1\}^n$, the set B_b has exactly 2^n elements. If for some b, B_b has two elements t^1, t^2 with $f_4(t^1) = f_4(t^2)$, then we are done as this satisfies condition 1. If not, for all b, the set $f_4(B_b)$ is equal to $\{0,1\}^n$. Now take the set $S = \bigcup_{i=1}^{2^{n/2}} \{t^i \in B_i :\ f_4(t^i) = i\}$. It is easy to see that $|S| = 2^{n/2}$, and the tweaks in set S satisfy condition 3.

Case 2: f_3 does not solely depend on f_1. Let us consider the sets $C_c := \{t \in A_a : f_3(t) = c\}, \forall c \in \{0,1\}^n$. If there exists $2^{n/2}$ indices $c_1, \ldots, c_{2^{n/2}}$ such that $C_{c_i} \neq \emptyset,\ \forall i \leq 2^{n/2}$, then the set $S = \{t^i \in C_{c_i} : i = 1, 2, \ldots, 2^{n/2}\}$, containing $2^{n/2}$ elements, satisfies condition 2. Otherwise, there exists at most $2^{n/2} - 1$ non-empty C_c's. By the Pigeonhole principle, at least one set, say $C_{c'}$ contains at least $2^{3n/2} + 1$ elements. Now look at the sets $B_b = \{t \in C_{c'} : f_2(t) = b\}, \forall b \in \{0,1\}^n$. If there are at most $2^{n/2} - 1$ non-empty B_b's, then there exists some b', for which $|B_{b'}| \geq 2^n + 1$, and hence, we would have $t^1, t^2 \in B_{b'}$ such that $f_4(t^1) = f_4(t^2)$, satisfying condition 1. Otherwise, we have at least $2^{n/2}$ many non-empty sets B_{b_i} for $i = 1, 2, \ldots, 2^{n/2}$. Now, if all the B_{b_i} sets are injective on f_4, we can construct a set $S := \{t^i \in B_{b_i} : i = 1, 2, \ldots, 2^{n/2}\}$ such that $f_4(t^i) \neq f_4(t^j)$ for all $i, j \in \{1, 2, \ldots, 2^{n/2}\}$ that provides the necessary values to satisfy condition 3. Otherwise, we will have some i for which $t^1, t^2 \in B_{b_i} : f_4(t^1) = f_4(t^2)$, and hence, t^1, t^2 will satisfy condition 1.

A.2 Proof of Lemma 3

Here we fix $\gamma \in \{0,1\}^n$, and our goal is to show that one of the following holds for any affine functions f_1, f_2, f_3, f_4.

1. There exist $t^1, t^2 \in \{0,1\}^{3n}$ such that $f_s(t^1) = f_s(t^2)$ for all $s \in \{1, 2, 3, 4\}$. cannot
2. There exist $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ satisfying $f_2(t^i) \neq f_2(t^j),\ f_3(t^i) = f_3(t^j),\ f_4(t^i) = f_4(t^j)$, for all $i, j \in \{1, 2, \ldots, 2^{n/2}\}$.
3. There exist $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ satisfying $f_1(t^i) \neq f_1(t^j),\ f_3(t^i) \neq f_3(t^j),\ f_4(t^i) = \gamma f_3(t^i)$, for all $i, j \in \{1, 2, \ldots, 2^{n/2}\}$.
4. There exist $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ satisfying $f_2(t^i) \neq f_2(t^j),\ f_4(t^i) = \gamma f_3(t^i)$, for all $i, j \in \{1, 2, \ldots, 2^{n/2}\}$.

We will consider the following two possible cases.

Case 1: f_4 depends solely on f_3. In this case, $f_4(t^i) = f_4(t^j)$ implies $f_3(t^i) = f_3(t^j)$, for all $t^i, t^j \in \{0,1\}^{3n}$. In addition, we assume that both f_1 and f_2 are mutually independent and also independent from f_3 and f_4. Otherwise, we will have t^1, t^2 satisfying condition 1. Note that, we have at least 2^{2n} many t^i's satisfying $f_3(t^i) = a,\ i \in \{1, 2, \ldots, 2^{2n}\}$, for some $a \in \{0,1\}^n$. Let $A_a := \{t^i \in \{0,1\}^{3n} :\ f_3(t^i) = a\}$. Now, define 2^n many sets depending on f_2: $\forall\ b \in \{0,1\}^n,\ B_b := \{t \in A_a : f_2(t) = b\}$. Suppose there exist $b_1, b_2, \ldots, b_{2^{n/2}}$ such that B_{b_i}'s are non-empty for each $i \in \{1, 2, \ldots, 2^{n/2}\}$. Then, the set S having one element from each of B_{b_i}, for all $i \in \{1, 2, \ldots, 2^{n/2}\}$, satisfies condition 2. If such b_i's do not exist, then we have at most $2^{n/2} - 1$ many non-empty B_b's. Hence, by the pigeonhole principle, there exists b' with $|B_{b'}| \geq 2^{3n/2} + 1$. Hence, there exist $t^1, t^2 \in B_{b'}$ such that $f_1(t^1) = f_1(t^2)$, and t^1, t^2 satisfies condition 1.

Case 2: f_4 does not solely depend on f_3. As all the f_s's are affine functions, we can express them as $f_s(t_1, t_2, t_3) = a_s.t_1 \oplus b_s.t_2 \oplus c_s.t_3 \oplus d_s$, where all $a_s, b_s, c_s, d_s \in \{0,1\}^n$. Consider the set $A := \{(t_1, t_2, t_3) \in \{0,1\}^{3n} : (a_4 \oplus \gamma a_3).t_1 \oplus (b_4 \oplus \gamma b_3).t_2 \oplus (c_4 \oplus \gamma c_3).t_3 \oplus (d_4 \oplus \gamma d_3) = 0\}$. It is easy to see that, by definition, $t \in A$ if and only if $f_4(t) = \gamma.f_3(t)$ and $|A| \geq 2^{2n}$. Now, define the sets $B_b := \{t \in A : f_2(t) = b\}$, for all $b \in \{0,1\}^n$. If there exist $2^{n/2}$ many non-empty B_b's, then we will pick up $2^{n/2}$ many values, one from each of those non-empty sets, satisfying condition 4. Otherwise, we will have at least one $b' \in 0, 1^n$ such that $|B_{b'}| \geq 2^{3n/2} + 1$. In this case, we consider the sets $C_c := \{t \in B_{b'} : f_1(t) = c\}$ for all $c \in \{0,1\}^n$. If we do not have at least $2^{n/2}$ many non-empty C_c's, then we will have a $c' \in \{0,1\}^n$ satisfying $|C_{c'}| \geq 2^n + 1$. Hence, we will have $t^1, t^2 \in C_{c'}$ satisfying condition 1. If there exists t^1, t^2 in some $C_{c''}$ such that $f_3(t^1) = f_3(t^2)$, then these two will satisfy condition 1. Otherwise, we can construct a set S taking one element from each of C_c's, and the elements of S will satisfy condition 3.

A.3 Proof of Lemma 4

Here our goal is to show that one of the following holds:

1. There exist $t^1, t^2 \in \{0,1\}^{3n}$ such that $f_s(t^1) = f_s(t^2)$ for all $s \in \{1, 2, 3, 4\}$.
2. There exist $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ satisfying $f_2(t^i) \neq f_2(t^j),\ f_4(t^i) = f_4(t^j)$, for all $i, j \in \{1, 2, \ldots, 2^{n/2}\}$.
3. There exist $t^1, t^2, \ldots, t^{2^{n/2}} \in \{0,1\}^{3n}$ satisfying $f_1(t^i) \neq f_1(t^j),\ f_3(t^i) \neq f_3(t^j),\ f_4(t^i) = f_4(t^j)$, for all $i, j \in \{1, 2, \ldots, 2^{n/2}\}$.

We again consider two cases following the dependency of the two functions f_2 and f_4 as follows:

Case 1: f_4 depends solely on f_2. In this case if $\{f_1, f_3\}$ are mutually dependent, or dependent with f_4 (consequently f_2), then we will have t^1, t^2 satisfying condition 1. Otherwise, we will have a set $A := \{t \in \{0,1\}^{3n} : f_4(t) = a\}$ for some $a \in \{0,1\}^n$, where $|A| \geq 2^{2n}$. Now, consider the sets $B_b := \{t \in A : f_1(t) = b\}$, for all $b \in \{0,1\}^n$. If we have at most $2^n - 1$ non-empty B_b's, then there exist one $b' \in \{0,1\}^n$ satisfying $|B_{b'}| \geq 2^n + 1$, and we can find $t^1, t^2 \in B_{b'}$ satisfying $f_3(t^1) = f_3(t^2)$. Hence, t^1, t^2 will satisfy condition 1. Otherwise, we have 2^n non-empty B_b's, then either f_3 is injective on each B_b, or there exist $b'' \in \{0,1\}^n$ such that $t^1, t^2 \in B_{b''}$ satisfying $f_3(t^1) = f_3(t^2)$. If such t^1, t^2 exists then those satisfy condition 1. If f_3 is injective on each B_b, then we can construct a set S taking one element from each B_b. The elements of the set S will satisfy condition 3.

Case 2: f_4 does not depend solely on f_2. Consider a set $A := \{t \in \{0,1\}^{3n} : f_4(t) = a\}$ for some $a \in \{0,1\}^n$, such that $|A| \geq 2^{2n}$. Now define $B_b := \{t \in A : f_2(t) = b\}$, $\forall\ b \in \{0,1\}^n$. If we have at least $2^{n/2}$ many non-empty B_b's, we can construct a set S with one element from each B_b's that satisfies condition 2. Otherwise, we will have some b' such that $|B_{b'}| \geq 2^{3n/2} + 1$. Now define $C_c := \{t \in B_{b'} : f_1(t) = c\}$, $\forall\ c \in \{c \in \{0,1\}^n\}$. If we have at most $2^{n/2} - 1$ non-empty C_c's then we will have some $C_{c'}$ with at least

$2^n + 1$ elements, that implies there exist $t^1, t^2 \in C_{c'}$ satisfying condition 1. Otherwise, we have at least $2^{n/2}$ non-empty C_c's. Now depending on whether f_3 is injective on all the non-empty C_c sets or not, we can have $2^{n/2}$ many tweaks satisfying condition 3, or two tweaks satisfying condition 1, respectively.

B Necessity of all tweak-dependent Keys if Message is fed into one of the non-final block-ciphers

In this section, we show that if the message is fed into the input of one of the non-final block cipher then again the construction requires all the block cipher keys to be tweak dependent.

B.1 Message is fed into First Block cipher

The generalized construction (ensuring the TBC is invertible) for this case, dubbed $\mathcal{C}_9$, is depicted in Fig. 16. Note that incorporating tweaks into the message or ciphertext does not amplify security. So, we refrain from using such modifications in our constructions. Now we consider cases with different numbers of tweak-independent keys.

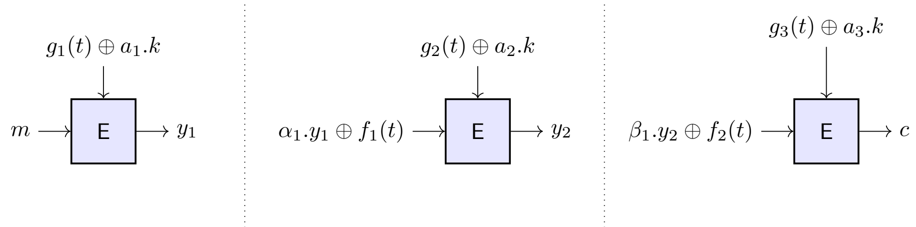


Figure 16: Construction $\mathcal{C}_9$: Message is fed in the first block cipher call

B.1.1 Constructions with Three Tweak-independent Keys

In this case, by definition, we have $g_1(t) = g_2(t) = g_3(t) = 0$. To attack this construction, our strategy is as follows:

1. Find two tweaks such that t^1, t^2 such that $f_1(t^1) = f_1(t^2)$, $f_2(t^1) = f_2(t^2)$. Note that, with this choice of tweaks, if we make two queries (m, t^1) and (m, t^2), we will have $y_1^1 = y_1^2$ as well as $y_2^1 = y_2^2$.
2. We can use the above observation to distinguish the TBC from a random tweakable permutation by making two oracle queries (m, t^1), (m, t^2), and verifying if the corresponding outputs match.

B.1.2 Constructions with Two Tweak-independent Keys

In this subsection, we consider all the possible TBC constructions with three block ciphers where we have two block cipher calls with tweak-independent keys.

Case 1: First block cipher uses the tweak-dependent key. In this case, we have $g_2(t) = g_3(t) = 0$. Here we consider two subcases. If we can find two tweaks t^1 and t^2 such that $f_1(t^1) = f_1(t^2)$, $f_2(t^1) = f_2(t^2)$ and $g_1(t^1) = g_1(t^2)$, we can simply carry out the previous attack. Otherwise, we can find $2^{n/2}$ many tweaks $t^1, \ldots, t^{2^{n/2}}$ for which

$f_2(t^i) = f_2(t^j)$ and $g_1(t^i) = g_1(t^j)$. Note that, for all i, j, we have $f_1(t^i) \neq f_1(t^j)$. In this case, we construct the attack as follows:

1. We make queries (m_i, t^i), for $i = 1, \ldots, 2^{n/2}$ with distinct messages, i.e., $m_i \neq m_j$, for all i and j. We expect to find a collision in the input of the second block cipher, i.e., find a and b such that $\alpha_1.y_1^a \oplus f_1(t^a) = \alpha_1.y_1^b \oplus f_1(t^b)$ and in that case, we have $y_2^a = y_2^b$, and the outputs, c_a and c_b matches.

2. Now we fix $\Delta \neq 0$ and find t^d such that $f_1(t^d) = f_1(t^a) + \Delta$, $f_2(t^d) = f_2(t^a)$ and $g_1(t^d) = g_1(t^a)$. We also find t^e such that $f_1(t^e) = f_1(t^b) + \Delta$, $f_2(t^e) = f_2(t^b)$ and $g_1(t^e) = g_1(t^b)$.

3. Finally, we make two queries: (m_a, t^d) and (m_b, t^e) and checks if the output matches.

Case 2: Second block cipher uses the tweak-dependent key. In this case, we have $g_1(t) = g_3(t) = 0$. Now we have two subcases. If we can find two tweaks t^1 and t^2 such that $f_1(t^1) = f_1(t^2)$, $f_2(t^1) = f_2(t^2)$ and $g_2(t^1) = g_2(t^2)$, we are done. Otherwise, find $2^{n/2}$ many tweaks $t^1, \ldots, t^{2^{n/2}}$ for which $f_2(t^i) = f_2(t^j)$ and $g_2(t^i) \neq g_2(t^j)$. Now we construct the attack as follows:

1. We make queries (m_i, t^i), for $i = 1, \ldots, 2^{n/2}$ with distinct messages, i.e., $m_i \neq m_j$, for all i and j. In this case, we expect to find a and b such that $y_2^a = y_2^b$ (b'day collision), and in that case the outputs, c_a and c_b match.

2. Now fix $\Delta \neq 0$. Find t^d such that $f_1(t^d) = f_1(t^a)$, $f_2(t^d) = f_2(t^a) + \Delta$ and $g_2(t^d) = g_2(t^a)$. Also, find t^e such that $f_1(t^e) = f_1(t^b)$, $f_2(t^e) = f_2(t^b) + \Delta$ and $g_2(t^e) = g_2(t^b)$.

3. Finally, query (m_a, t^d) and (m_b, t^e) and checks whether the output matches.

Case 3: Final block cipher uses the tweak-dependent key. In this case, we have $g_1(t) = g_2(t) = 0$. Now we have two subcases. If we can find two tweaks t^1 and t^2 such that $f_1(t^1) = f_1(t^2)$, $f_2(t^1) = f_2(t^2)$ and $g_3(t^1) = g_3(t^2)$, we have a trivial attack. Otherwise, find $2^{n/2}$ many tweaks $t^1, \ldots, t^{2^{n/2}}$ for which $f_2(t^i) = f_2(t^j)$ and $g_3(t^i) = g_3(t^j)$. Note that, for all i, j, we have $f_1(t^i) \neq f_1(t^j)$. Here we construct the attack as follows:

1. We make queries (m_i, t^i), for $i = 1, \ldots, 2^{n/2}$ with distinct messages, i.e., $m_i \neq m_j$, for all i and j. In this case, we expect to find a and b such that a collision occurs in the input of the second block cipher, i.e., $\alpha_1.y_1^a \oplus f_1(t^a) = \alpha_1.y_1^b \oplus f_1(t^b)$, and in that case the outputs, c_a and c_b matches.

2. Now fix $\Delta \neq 0$. Find t^d such that $f_1(t^d) = f_1(t^a)$, $f_2(t^d) = f_2(t^d) + \Delta$ and $g_3(t^d) = g_3(t^a)$. Also, find t^e such that $f_1(t^e) = f_1(t^b)$, $f_2(t^e) = f_2(t^b) + \Delta$ and $g_3(t^e) = g_3(t^b)$.

3. Finally, we query (m_a, t^d) and (m_b, t^e) and checks whether the output matches.

B.1.3 Constructions with One Tweak-independent Key

Case 1: First two block ciphers use tweak-dependent key. In this case, we have $g_3(t) = 0$. If we can find two tweaks t^1 and t^2 such that $f_1(t^1) = f_1(t^2)$, $f_2(t^1) = f_2(t^2)$, $g_1(t^1) = g_1(t^2)$ and $g_2(t^1) = g_2(t^2)$, we have a trivial attack. Otherwise, we break it into the following cases based on the dependency of f_1, f_2, g_1 and g_2 as given below.

Subcase 1: $\{f_2, g_1\}$ is linearly dependent. In this case, we proceed as follows. First we find $2^{n/2}$ many tweaks $t^1, \ldots, t^{2^{n/2}}$ such that $g_1(t^i) = g_1(t^j)$, $f_1(t^i) \neq f_1(t^j)$, $g_2(t^i) = g_2(t^j)$, $f_2(t^i) = f_2(t^j)$.

1. We make queries (m_i, t^i), for $i = 1, \ldots, 2^{n/2}$ such that $m_i \neq m_j$, for all i, j. Here we expect to find a and b such that $\alpha_1.y_1^a \oplus f_1(t^a) = \alpha_1.y_1^b \oplus f_1(t^b)$ (by birthday collision), and in that case the outputs, c_a and c_b match.
2. Now fix $\Delta \neq 0$. Find t^d such that $g_1(t^d) = g_1(t^a)$, $f_1(t^d) = f_1(t^a) \oplus \Delta$ and $g_2(t^d) = g_2(t^a)$. Also, find t^e such that $g_1(t^e) = g_1(t^b)$, $f_1(t^e) = f_1(t^b) \oplus \Delta$ and $g_2(t^e) = g_2(t^b)$.
3. Finally, we query (m_a, t^d) and (m_b, t^e) and checks whether the output matches.

Subcase 2: $\{f_1, f_2, g_1\}$ are linearly dependent. In this case we proceed as follows. First we find $2^{n/2}$ many tweaks $t^1, \ldots, t^{2^{n/2}}$ such that $g_1(t^i) \neq g_1(t^j)$, $g_2(t^i) = g_2(t^j)$, $f_2(t^i) = f_2(t^j)$.

1. We make queries (m, t^i), for $i = 1, \ldots, 2^{n/2}$. Here we expect to find a and b such that $\alpha_1.y_1^a \oplus f_1(t^a) = \alpha_1.y_1^b \oplus f_1(t^b)$ (by birthday collision), and in that case the outputs, c_a and c_b matches.
2. Now fix $\Delta \neq 0$. Find t^d such that $g_1(t^d) = g_1(t^a)$, $f_1(t^d) = f_1(t^a)$ and $g_2(t^d) = g_2(t^a) \oplus \Delta$. Also, find t^e such that $g_1(t^e) = g_1(t^b)$, $f_1(t^e) = f_1(t^b)$ and $g_2(t^e) = g_2(t^b) \oplus \Delta$.
3. Finally, we query (m, t^d) and (m, t^e) and checks whether the output matches.

Subcase 3: $\{f_1, f_2\}$ or $\{f_1, g_2\}$ or $\{f_2, g_2\}$ or $\{f_1, f_2, g_2\}$ are linearly dependent. First, let us consider the cases when $\{f_1, f_2\}$ or $\{f_1, f_2, g_2\}$ is linearly dependent. Here we proceed as follows. First we find $2^{n/2}$ many tweaks $t^1, \ldots, t^{2^{n/2}}$ such that $g_1(t^i) \neq g_1(t^j)$, $f_1(t^i) = f_1(t^j)$, $f_2(t^i) = f_2(t^j)$, $g_2(t^i) = g_2(t^j)$.

1. We make queries (m, t^i), for $i = 1, \ldots, 2^{n/2}$, for all i and j. In this case, we expect to find a and b such that $y_1^a = y_1^b$ (by birthday collision), and in that case the outputs, c_a and c_b match.
2. Now fix $\Delta \neq 0$. Find t^d such that $g_1(t^d) = g_1(t^a)$, $f_1(t^d) = f_1(t^a) \oplus \Delta$ and $g_2(t^d) = g_2(t^a)$. Also, find t^e such that $g_1(t^e) = g_1(t^b)$, $f_1(t^e) = f_1(t^b) + \Delta$ and $g_2(t^e) = g_2(t^b)$.
3. Finally, we query (m, t^d) and (m, t^e) and checks whether the output matches.

When $\{f_2, g_2\}$ is linearly dependent, we follow the same algorithm except that we choose t^d and t^e as follows: $g_1(t^d) = g_1(t^a)$, $f_1(t^d) = f_1(t^a)$ and $g_2(t^d) = g_2(t^a) \oplus \Delta$; $g_1(t^e) = g_1(t^b)$, $f_1(t^e) = f_1(t^b)$ and $g_2(t^e) = g_2(t^b) + \Delta$.

When $\{f_1, g_2\}$ is linearly dependent, we follow the same algorithm except that we choose t^d and t^e as follows: $g_1(t^d) = g_1(t^a)$, $f_2(t^d) = f_2(t^a)$ and $f_1(t^d) = f_1(t^a) \oplus \Delta$; $g_1(t^e) = g_1(t^b)$, $f_2(t^e) = f_2(t^b)$ and $f_1(t^e) = f_1(t^b) + \Delta$.

Subcase 4: $\{f_1, g_1\}$ or $\{f_1, g_1, g_2\}$ are linearly dependent. Here we find $2^{n/2}$ many tweaks $t^1, \ldots, t^{2^{n/2}}$ such that $g_1(t^i) \neq g_1(t^j)$, $f_1(t^i) \neq f_1(t^j)$, $f_2(t^i) = f_2(t^j)$, $g_2(t^i) = g_2(t^j)$.

1. We make queries (m, t_i), for $i = 1, \ldots, 2^{n/2}$, for all i and j. Here we expect to find a and b such that $\alpha_1.y_1^a \oplus f_1(t^a) = \alpha_1.y_1^b \oplus f_1(t^b)$ (by birthday collision), and in that case the outputs, c_a and c_b match.

2. Now fix $\Delta \neq 0$. Find t^d such that $g_1(t^d) = g_1(t^a)$, $f_1(t^d) = f_1(t^a)$, $f_2(t^d) = f_2(t^a) + \Delta$. Also, find t^e such that $g_1(t^e) = g_1(t^b)$, $f_1(t^e) = f_1(t^b)$, $f_2(t^e) = f_2(t^b) + \Delta$.

3. Finally, we query (m, t^d) and (m, t^e) and checks whether the output matches.

Subcase 5: $\{g_1, g_2\}$ or $\{f_2, g_1, g_2\}$ are linearly dependent. In this case we find $2^{n/2}$ many tweaks $t^1, \ldots, t^{2^{n/2}}$ such that $g_1(t^i) = g_1(t^j)$, $f_1(t^i) \neq f_1(t^j)$, $f_2(t^i) = f_2(t^j)$, $g_2(t^i) = g_2(t^j)$.

1. We make queries (m_i, t_i), for $i = 1, \ldots, 2^{n/2}$, where $m_i \neq m_j$, for all i and j. In this case, we expect to find a and b such that $y_2^a = y_2^b$ (by birthday collision), and in that case the outputs, c_a and c_b match.

2. Now fix $\Delta \neq 0$. Find t^d such that $g_1(t^d) = g_1(t^a)$, $f_1(t^d) = f_1(t^a)$, $f_2(t^d) = f_2(t^a) + \Delta$. Also, find t^e such that $g_1(t^e) = g_1(t^b)$, $f_1(t^e) = f_1(t^b)$, $f_2(t^e) = f_2(t^b) + \Delta$.

3. Finally, we query (m, t^d) and (m, t^e) and checks whether the output matches.

Subcase 6: None of the proper subsets of $\{f_1, f_2, g_1, g_2\}$ are linearly dependent. In this case, we proceed as follows. First we find $2^{n/2}$ many tweaks $t^1, \ldots, t^{2^{n/2}}$ such that $g_1(t^i) \neq g_1(t^j)$, $f_1(t^i) \neq f_1(t^j)$, $g_2(t^i) = g_2(t^j)$, $f_2(t^i) = f_2(t^j)$.

1. We make queries (m, t^i), for $i = 1, \ldots, 2^{n/2}$, for all i and j. We expect to find a and b such that $\alpha_1 y_1^a + f_1(t^a) = y_1^b + f_1(t^b)$ (by birthday collision), and in that case the outputs, c_a and c_b matches.

2. Now fix $\Delta \neq 0$. Find t^d such that $g_1(t^d) = g_1(t^a)$, $f_1(t^d) = f_1(t^a) + \Delta$, $f_2(t^d) = f_2(t^a)$. Also, find t^e such that $g_1(t^e) = g_1(t^b)$, $f_1(t^e) = f_1(t^b) + \Delta$, $f_2(t^e) = f_2(t^b)$.

3. Finally, we query (m, t^d) and (m, t^e) and checks whether the output matches.

B.2 Message is fed into Second Block cipher

The generalized construction (ensuring the TBC is invertible) for this case, dubbed $\mathcal{C}_{10}$, is depicted in Fig. 17. Note that incorporating tweaks into the message or ciphertext does not amplify security. So, we refrain from using such modifications in our constructions. Now we consider cases with different number of tweak-independent keys.

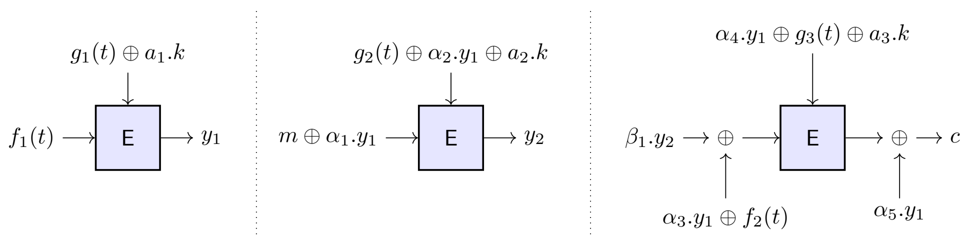


Figure 17: Construction $\mathcal{C}_{10}$: Message is fed into Second Block Cipher Call

B.2.1 Constructions with Three Tweak-independent Keys

In this case, by definition, we have $g_1(t) = g_2(t) = g_3(t) = 0$, $\alpha_2 = \alpha_4 = 0$. To attack this construction, our strategy is as follows:

1. Find two tweaks such that t^1, t^2 such that $f_1(t^1) = f_1(t^2)$, $f_2(t^1) = f_2(t^2)$. Note that, with this choice of tweaks, if we make two queries (m, t^1) and (m, t^2), we will have $y_1^1 = y_1^2$ as well as $y_2^1 = y_2^2$.
2. We can use the above observation to distinguish the TBC from a random tweakable permutation by making two oracle queries (m, t^1), (m, t^2), and verifying if the corresponding outputs match.

B.2.2 Constructions with Two Tweak-independent Keys

Here we consider the possible TBC constructions with three block ciphers where we have two block cipher calls with tweak-independent keys.

Case 1: First block cipher uses the tweak-dependent key. In this case, we have $g_2(t) = g_3(t) = 0$ and $\alpha_2 = \alpha_4 = 0$. Now we have two subcases. If we can find two tweaks t^1 and t^2 such that $f_1(t^1) = f_1(t^2)$, $f_2(t^1) = f_2(t^2)$ and $g_1(t^1) = g_1(t^2)$, we have the trivial attack. Otherwise, we find $2^{n/2}$ many tweaks $t^1, \ldots, t^{2^{n/2}}$ for which $f_1(t^i) = f_1(t^j)$ and $f_2(t^i) = f_2(t^j)$. Note that, for all i, j, we have $g_1(t^i) \neq g_1(t^j)$. In this case we construct the attack as follows:

1. We make queries (m, t^i), for $i = 1, \ldots, 2^{n/2}$. In this case we expect to find a and b such that $y_1^a = y_1^b$ (b'day collision), and in that case the outputs, c_a and c_b matches.
2. Now fix $\Delta \neq 0$. Make two queries $(m + \Delta, t^a)$ and $(m + \Delta, t^b)$ and checks whether the output matches.

Similar attacks will work for the other two cases.

B.2.3 Constructions with One Tweak-independent Key

Here we consider the possible TBC constructions with three block ciphers where we have one block cipher call with tweak-independent keys.

Case 1: First block cipher uses the tweak-dependent key. In this case, we have $g_3(t) = 0$ and $\alpha_4 = 0$. If we can find two tweaks t^1 and t^2 such that $f_1(t^1) = f_1(t^2)$, $f_2(t^1) = f_2(t^2)$, $g_1(t^1) = g_1(t^2)$ and $g_2(t^1) = g_2(t^2)$, we have a trivial attack. Otherwise, we break it into the following cases based on the dependency of f_1, f_2, g_1 and g_2 as given below.

Subcase 1: $\{g_1, g_2\}$ or $\{g_1, f_2\}$ or $\{g_1, g_2, f_2\}$ is linearly dependent. Here we can find $2^{n/2}$ many tweaks $t^1, \ldots, t^{2^{n/2}}$ for which $g_1(t^i) = g_1(t^j)$, $f_2(t^i) = f_2(t^j)$, $g_2(t^i) = g_2(t^j)$ and $f_1(t^i) \neq f_1(t^j)$. Now we mount the following attack:

1. We make queries (m_i, t^i), for $i = 1, \ldots, 2^{n/2}$, where $m_i := \alpha_1 \alpha_3^{-1} f_2(t^i)$, for all i. Note that by birthday assumption, we expect to find a and b such that $m_i \oplus \alpha_1.y_1^a = m_j \oplus \alpha_1.y_1^b$, then by definition, we will have $y_2^a = y_2^b$, and subsequently the outputs, c_a and c_b matches. Note that the choice of our messages ensures that we have $\alpha_3(y_1^i \oplus y_1^j) = f_2(t^i) \oplus f_2(t^j)$.
2. Now fix $\Delta \neq 0$. Make two queries $(m + \Delta, t^a)$ and $(m + \Delta, t^b)$ and checks whether the output matches.

Subcase 2: All Other Cases. For all the remaining cases, we can find $2^{n/2}$ many tweaks $t^1, \ldots, t^{2^{n/2}}$ for which $g_2(t^i) = g_2(t^j)$, $f_2(t^i) = f_2(t^j)$ and $g_1(t^i) \neq g_1(t^j)$, and mount the following attack:

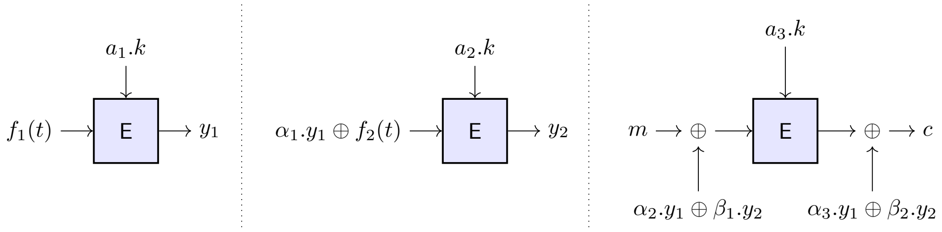
1. We make queries (m, t^i), for $i = 1, \ldots, 2^{n/2}$. Note that by birthday assumption, we expect to find a and b such that $y_1^a = y_1^b$, then by definition, we will have $y_2^a = y_2^b$, and subsequently the outputs, c_a and c_b matches.

2. Now fix $\Delta \neq 0$. Make two queries $(m + \Delta, t^a)$ and $(m + \Delta, t^b)$ and checks whether the output matches.

Similar attacks will work for the remaining two cases, i.e., when the second or third block cipher uses a tweak-independent key.

C Distinguishing Algorithms against various Constructions

C.1 Construction $\mathcal{C}_1$ and A Distinguishing Algorithm against It

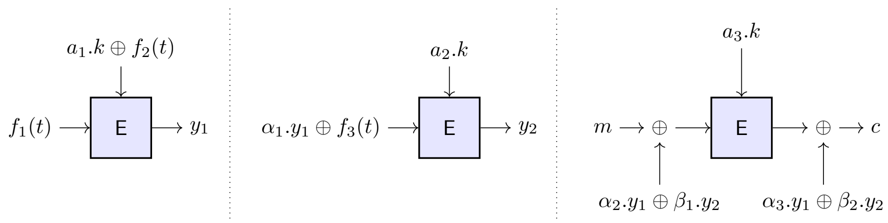


Distinguisher $\mathcal{D}_1$

```
Choose t^1, t^2 : f_1(t^1) = f_1(t^2) ∧ f_2(t^1) = f_2(t^2);
Make TBC Queries (m, t^1), (m, t^2); Let the responses be c^1, c^2;
Return 1, if c^1 = c^2;
```

Figure 18: Distinguishing Algorithm against Construction $\mathcal{C}_1$

C.2 Construction $\mathcal{C}_2$ and A Distinguishing Algorithm against It



Distinguisher $\mathcal{D}_2$

$\texttt{Choose}\ t^1, \ldots, t^{2n/2} :\ \forall i, j,\ f_2(t^i) \neq f_2(t^j) \wedge f_3(t^i) = f_3(t^j);$

$\texttt{Make}\ 2^{n/2}\ \texttt{TBC Queries}\ (m, t^1), \ldots, (m, t^q);$

$\texttt{Let the responses be}\ c_1, \ldots, c_q,\ \texttt{respectively};$

$\texttt{Find}\ a, b :\ c_a = c_b;$

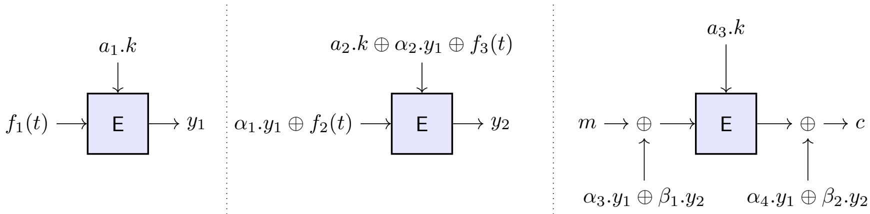
$\texttt{Make TBC Queries}\ (m^\star, t^a),\ (m^\star, t^b),\ m \neq m^\star;$

$\texttt{Let the responses be}\ c_a^\star,\ c_b^\star;$

$\texttt{Return}\ 1,\ \texttt{if}\ c_a^\star = c_b^\star;$

Figure 19: Distinguishing Algorithm against Construction $\mathcal{C}_2$

C.3 Construction $\mathcal{C}_3$ and A Distinguishing Algorithm against It



Distinguisher $\mathcal{D}_3$

$\texttt{Choose}\ t^1, \ldots, t^{2^{n/2}} :\ \forall i, j,\ f_1(t^i) = f_1(t^j) \wedge f_3(t^i) \neq f_3(t^j);$

$\texttt{Make}\ 2^{n/2}\ \texttt{TBC Queries}\ (m, t^1), \ldots, (m, t^q);$

$\texttt{Let the responses be}\ c_1, \ldots, c_q,\ \texttt{respectively};$

$\texttt{Find}\ a, b :\ c_a = c_b;$

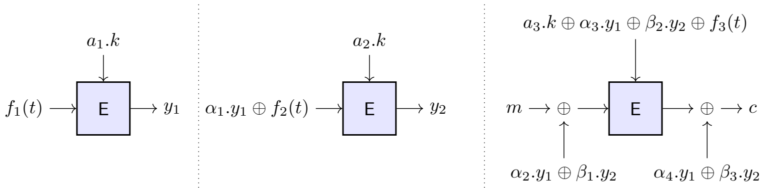
$\texttt{Make TBC Queries}\ (m^\star, t^a),\ (m^\star, t^b),\ m \neq m^\star;$

$\texttt{Let the responses be}\ c_a^\star,\ c_b^\star;$

$\texttt{Return}\ 1,\ \texttt{if}\ c_a^\star = c_b^\star;$

Figure 20: Distinguishing Algorithm against Construction $\mathcal{C}_3$

C.4 Construction $\mathcal{C}_4$ and A Distinguishing Algorithm against It



Distinguisher $\mathcal{D}_4$ against the Construction when $\beta_1 = \beta_2 = 0$

`Choose` $t^1, t^2 :\ f_1(t^1) = f_1(t^2) \wedge f_2(t^1) \neq f_2(t^2) \wedge f_3(t^1) = f_3(t^2)$;
`Make` 2 `TBC Queries` $(m, t^1), (m, t^2)$;
`Let the responses be` $c_1,\ c_2,$ `respectively`;
`Make TBC Queries` $(m \oplus \Delta, t^1),\ (m \oplus \Delta, t^2),\ \Delta \neq 0$;
`Let the responses be` $c_1^\star,\ c_2^\star$;
`Return` 1, `if` $c_1^\star \oplus c_2^\star = c_1 \oplus c_2$;

Distinguisher $\mathcal{D}_4$ against the Construction when $\beta_1 \neq 0,\ \beta_2 = 0$

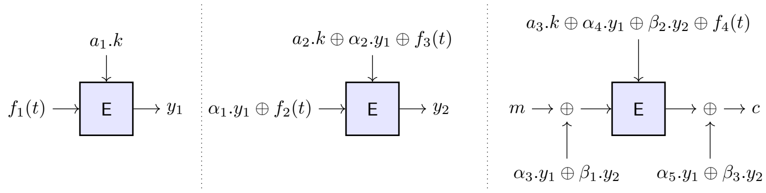
`Choose` $t^1, \ldots, t^{2^{n/2}} :\ \forall i, j,\ f_1(t^i) = f_1(t^j) \wedge f_2(t^i) \neq f_2(t^j) \wedge f_3(t^i) = f_3(t^j)$;
`Make` $2^{n/2}$ `TBC Queries` $(m_1, t^1), \ldots, (m_q, t^q) :\ \forall i, j\ m_i \neq m_j$;
`Let the responses be` $c_1, \ldots, c_q,$ `respectively`;
`Find` $i, j :\ c_i \oplus c_j = \beta_1^{-1}\beta_3(m_i \oplus m_j)$;
`Make TBC Queries` $(m_i \oplus \Delta, t^i),\ (m_j \oplus \Delta, t^j),\ \Delta \neq 0$;
`Let the responses be` $c_i^\star,\ c_j^\star$;
`Return` 1, `if` $c_i^\star \oplus c_i^\star = \beta_1^{-1}\beta_3(m_i \oplus m_j)$;

Distinguisher $\mathcal{D}_4$ against the Construction when $\beta_2 \neq 0$

`Choose` $t^1, \ldots, t^{2^{n/2}} :\ \forall i, j,\ f_1(t^i) = f_1(t^j) \wedge f_2(t^i) \neq f_2(t^j) \wedge f_3(t^i) \neq f_3(t^j)$;
`Make` $2^{n/2}$ `TBC Queries` $(m_1 := \beta_2^{-1}\beta_1 f_3(t^1), t^1), \ldots, (m_q := \beta_2^{-1}\beta_1 f_3(t^q), t^q)$;
`Let the responses be` $c_1, \ldots, c_q,$ `respectively`;
`Find` $i, j :\ c_i \oplus c_j = \beta_1^{-1}\beta_3(m_i \oplus m_j)$;
`Make TBC Queries` $(m_i \oplus \Delta, t^i),\ (m_j \oplus \Delta, t^j),\ \Delta \neq 0$;
`Let the responses be` $c_i^\star,\ c_j^\star$;
`Return` 1, `if` $c_i^\star \oplus c_i^\star = \beta_1^{-1}\beta_3(m_i \oplus m_j)$;

Figure 20: Distinguishing Algorithm against Construction $\mathcal{C}_4$.

C.5 Construction $\mathcal{C}_5$ and A Distinguishing Algorithm against It



Distinguisher $\mathcal{D}_5$ against the Construction when $\beta_1 = \beta_2 = 0$

`Choose` $t^1, t^2 :\ f_1(t^1) = f_1(t^2) \wedge f_4(t^1) = f_4(t^2)$;
`Make` 2 `TBC Queries` $(m, t^1), (m, t^2)$; `Let the responses be` $c_1,\ c_2$;
`Make TBC Queries` $(m \oplus \Delta, t^1),\ (m \oplus \Delta, t^2),\ \Delta \neq 0$;
`Let the responses be` $c_1^\star,\ c_2^\star$;
`Return` 1, `if` $c_1^\star \oplus c_2^\star = c_1 \oplus c_2$;

Distinguisher $\mathcal{D}_5$ against the Construction when $\beta_1 \neq 0,\ \beta_2 = 0$

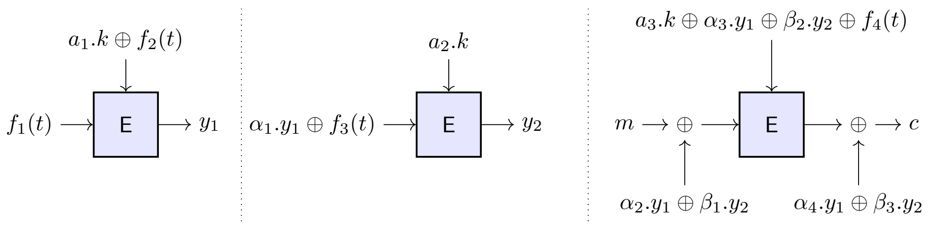
`Choose` $t^1, \ldots, t^{2^{n/2}} :\ \forall i, j,\ f_1(t^i) = f_1(t^j) \wedge f_4(t^i) = f_4(t^j)$
$\wedge\, (f_3(t^i) \neq f_3(t^j) \vee f_2(t^i) \neq f_2(t^j))$;
`Make` $2^{n/2}$ `TBC Queries` $(m_1, t^1), \ldots, (m_q, t^q) :\ \forall i, j\ m_i \neq m_j$;
`Let the responses be` $c_1, \ldots, c_q$, `respectively`;
`Find` $i, j :\ \beta_1(c_i \oplus c_j) = \beta_3(m_i \oplus m_j)$;
`Make TBC Queries` $(m_i \oplus \Delta, t^i),\ (m_j \oplus \Delta, t^j),\ \Delta \neq 0$;
`Let the responses be` $c_i^\star,\ c_j^\star$;
`Return` 1, `if` $c_i^* \oplus c_j^* = c_i \oplus c_j$;

Distinguisher $\mathcal{D}_4$ against the Construction when $\beta_2 \neq 0$

`Choose` $t^1, \ldots, t^{2^{n/2}} :\ \forall i, j,\ f_1(t^i) = f_1(t^j) \wedge f_3(t^i) \neq f_3(t^j)$ `or`
$f_1(t^i) = f_1(t^j) \wedge f_2(t^i) \neq f_2(t^j) \wedge f_4(t^i) \neq f_4(t^j)$;
`Make` $2^{n/2}$ `TBC Queries` $(m_1 := \beta_2^{-1}\beta_1 f_4(t^1), t^1), \ldots, (m_q := \beta_2^{-1}\beta_1 f_4(t^q), t^q)$;
`Let the responses be` $c_1, \ldots, c_q$, `respectively`;
`Find` $i, j :\ \beta_1(c_i \oplus c_j) = \beta_3(m_i \oplus m_j)$;
`Make TBC Queries` $(m_i \oplus \Delta, t^i),\ (m_j \oplus \Delta, t^j),\ \Delta \neq 0$;
`Let the responses be` $c_i^\star,\ c_j^\star$;
`Return` 1, `if` $c_i^\star \oplus c_i^\star = c_i \oplus c_j$;

Figure 21: Distinguishing Algorithm against Construction $\mathcal{C}_5$.

C.6 Construction $\mathcal{C}_6$ and A Distinguishing Algorithm against It



Distinguisher $\mathcal{D}_6$ against the Construction when $\alpha_1 \neq 0,\ \alpha_3 = 0$

1 : `Choose` $t^1, \ldots, t^{2^{n/2}}$: $\forall i, j,\ f_2(t^i) \neq f_2(t^j) \wedge f_4(t^i) = f_4(t^j)$ `or`
$f_1(t^i) \neq f_1(t^j) \wedge f_3(t^i) \neq f_3(t^j) \wedge f_4(t^i) = f_4(t^j)$;

2 : `Make` $2^{n/2}$ `TBC Queries` $(m_1 = \alpha_2\alpha_1^{-1} f_3(t^1), t^1), \ldots,$
$(m_q = \alpha_2\alpha_1^{-1} f_3(t^q), t^q)$: $\forall i, j\ m_i \neq m_j$;

3 : `Let the responses be` $c_1, \ldots, c_q$, `respectively`;

4 : `Find` i, j : $\alpha_1(c_i \oplus c_j) = \alpha_4(f_3(t^i) \oplus f_3(t^j))$;

5 : `Make TBC Queries` $(m_i \oplus \Delta, t^i),\ (m_j \oplus \Delta, t^j),\ \Delta \neq 0$;

6 : `Let the responses be` $c_i^\star,\ c_j^\star$;

7 : `Return` 1, `if` $c_i^* \oplus c_j^* = c_i \oplus c_j$;

Distinguisher $\mathcal{D}_6$ against the Construction when $\alpha_1 \neq 0,\ \alpha_3 \neq 0$

1 : `Choose` $t^1, \ldots, t^{2^{n/2}}$: $\forall i, j,\ f_2(t^i) \neq f_2(t^j),\ f_3(t^i) = f_3(t^j),\ f_4(t^i) = f_4(t^j)$ `or`
$f_1(t^i) \neq f_1(t^j),\ f_3(t^i) \neq f_3(t^j),\ f_4(t^i) = \alpha_3\alpha_1^{-1} f_3(t^i)$ `or`
$f_2(t^i) \neq f_2(t^j),\ f_4(t^i) = \alpha_3\alpha_1^{-1} f_3(t^i)$;

2 : `Make` $2^{n/2}$ `TBC Queries` $(m_1 := \alpha_2^{-1}\alpha_1 f_3(t^1), t^1), \ldots, (m_q := \alpha_2^{-1}\alpha_1 f_3(t^q), t^q)$;

3 : `Let the responses be` $c_1, \ldots, c_q$, `respectively`;

4 : `Find` i, j : $\alpha_1(c_i \oplus c_j) = \alpha_4(f_3(t^i) \oplus f_3(t^j))$;

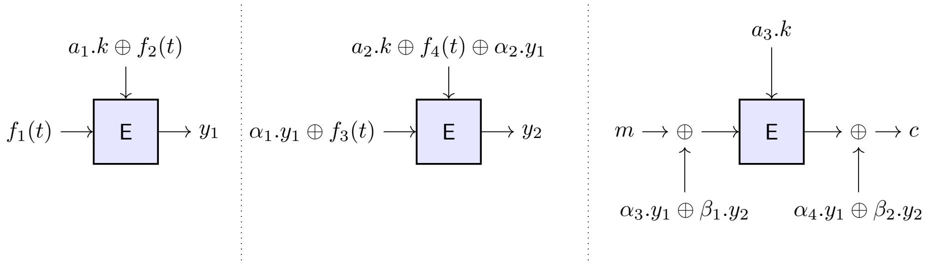
5 : `Make TBC Queries` $(m_i \oplus \Delta, t^i),\ (m_j \oplus \Delta, t^j),\ \Delta \neq 0$;

6 : `Let the responses be` $c_i^\star,\ c_j^\star$;

7 : `Return` 1, `if` $c_i^\star \oplus c_i^\star = c_i \oplus c_j$;

Figure 22: Distinguishing Algorithm against Construction $\mathcal{C}_6$.

C.7 Construction $\mathcal{C}_7$ and A Distinguishing Algorithm against It



Distinguisher $\mathcal{D}_7$ against the Construction when $\beta_1 \neq 0$

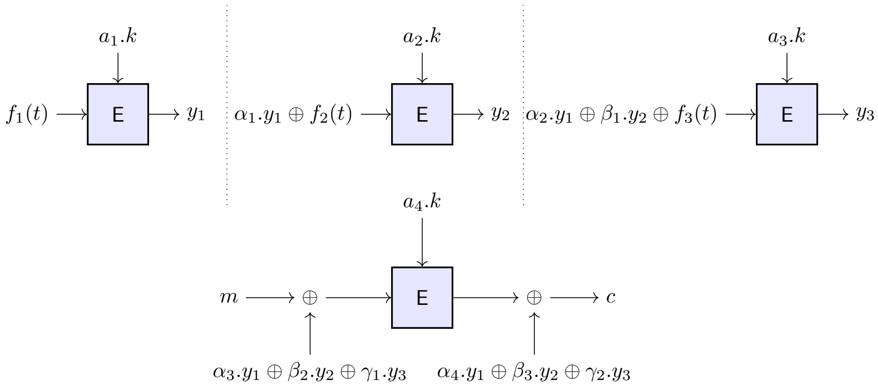
```
Choose $t^1, \ldots, t^{2^{n/2}}$ : $\forall i, j,\ f_1(t^i) = f_1(t^j) \wedge f_2(t^i) = f_2(t^j)$
        $\wedge\ (f_3(t^i) \neq f_3(t^j) \vee f_4(t^i) \neq f_4(t^j))$;
Make $2^{n/2}$ TBC Queries $(m_1, t^1), \ldots, (m_q, t^q)$ : $\forall i, j\ m_i \neq m_j$;
Let the responses be $c_1, \ldots, c_q$, respectively;
Find $i, j$ : $\beta_1(c_i \oplus c_j) = \beta_2(m_i \oplus m_j)$;
Make TBC Queries $(m_i \oplus \Delta, t^i)$, $(m_j \oplus \Delta, t^j)$, $\Delta \neq 0$;
Let the responses be $c_i^\star$, $c_j^\star$;
Return 1, if $c_i^* \oplus c_j^* = c_i \oplus c_j$;
```

Distinguisher $\mathcal{D}_7$ against the Construction when $\beta_1 = 0$

```
Choose $t^1, t^2$ such that $f_1(t^1) = f_1(t^2) \wedge f_2(t^1) = f_2(t^2)$;
Make 2 TBC Queries $(m, t^1)$ and $(m, t^2)$;
Let the responses be $c_1$, $C_2$ respectively;
Make TBC Queries $(m \oplus \Delta, t^1)$, $(m \oplus \Delta, t^1)$, $\Delta \neq 0$;
Let the responses be $c_1^\star$, $c_2^\star$;
Return 1, if $c_1^\star \oplus c_1^\star = c_1 \oplus c_2$;
```

Figure 23: Distinguishing Algorithm against Construction $\mathcal{C}_7$.

C.8 Construction $\mathcal{C}_8$ and A Distinguishing Algorithm against It



Distinguisher $\mathcal{D}_8$ against the Construction when $\gamma_1 = 0$

1 : `Choose` t^1, t^2 `satisfying` $f_1(t^1) = f_1(t^2)$ `and` $f_2(t^1) = f_2(t^2)$;
2 : `Make` $2^{n/2}$ `TBC Queries` $(m, t^1), (m, t^2)$;
3 : `Let the responses be` c_1, c_2 `respectively`;
4 : `Make TBC Queries` $(m \oplus \Delta, t^1),\ (m \oplus \Delta, t^2),\ \Delta \neq 0$;
5 : `Let the responses be` $c_1^\star,\ c_2^\star$;
6 : `Return` 1, `if` $c_1^\star \oplus c_2^\star = c_1 \oplus c_2$;

Distinguisher $\mathcal{D}_8$ against the Construction when $\gamma_1 \neq 0$

1 : `Choose` $t^1, \ldots, t^q :\ \forall i, j,\ f_1(t^i) = f_1(t^j) \wedge f_2(t^i) = f_2(t^j) \wedge f_3(t^i) \neq f_3(t^j)$;
2 : `Make` $2^{n/2}$ `TBC Queries` $(m^1, t^1), \ldots, (m^q, t^q) :\ \forall i, j,\ m^i \neq m^j$;
3 : `Let the responses be` $c_1, \ldots, c_q$, `respectively`;
4 : `Find` $i, j :\ \gamma_1(c_i \oplus c_j) = \gamma_2(m_i \oplus m_j)$;
5 : `Make TBC Queries` $(m_i \oplus \Delta, t^i),\ (m_j \oplus \Delta, t^j),\ \Delta \neq 0$;
6 : `Let the responses be` $c_i^\star,\ c_j^\star$;
7 : `Return` 1, `if` $c_i^\star \oplus c_j^\star = c_i \oplus c_j$;

Figure 24: Distinguishing Algorithm against Construction $\mathcal{C}_8$.