Tweakable ForkCipher from Ideal Block Cipher

Sougata Mandal

Institute for Advancing Intelligence, TCG CREST, Kolkata, India
Ramakrishna Mission Vivekananda Educational and Research Institute, India
sougatamandal2014@gmail.com

Abstract. In ASIACRYPT 2019, Andreeva et al. introduced a new symmetric key primitive called the *forkcipher*, designed for lightweight applications handling short messages. A forkcipher is a keyed function with a public tweak, featuring fixed-length input and fixed-length (expanding) output. They also proposed a specific forkcipher, ForkSkinny, based on the tweakable block cipher SKINNY, and its security was evaluated through cryptanalysis. Since then, several efficient AEAD and MAC schemes based on forkciphers have been proposed, catering not only to short messages but also to various purposes such as leakage resilience and cloud security. While forkciphers have proven to be efficient solutions for designing AEAD schemes, the area of forkcipher design remains unexplored, particularly the lack of provably secure forkcipher constructions. In this work, we propose forkcipher design for various tweak lengths, based on a block cipher as the underlying primitive. We provide proofs of security for these constructions, assuming the underlying block cipher behaves as an ideal block cipher. First, we present a forkcipher, $\widetilde{\mathsf{F}}1$, for an n-bit tweak and prove its optimal (n-bit) security. Next, we propose another construction, $\widetilde{\mathsf{F}}2$, for a $2n$-bit tweak, also proving its optimal (n-bit) security. Finally, we introduce a construction, $\widetilde{\mathsf{F}}r$, for a general rn-bit tweak, achieving n-bit security.

1 Introduction

Forkcipher [4] was introduced as a keyed function with an optional public tweak, taking an n-bit input and producing a $2n$-bit output. The security of a tweakable forkcipher is defined in terms of indistinguishability from two independently chosen tweakable permutations on the same tweak space and input space $\{0,1\}^n$. For the remainder of this work, we will focus on forkciphers with a tweak, i.e., tweakable forkciphers. At times, we will simply refer to them as forkciphers. In [4], a concrete forkcipher called ForkSkinny, based on the tweakable block cipher SKINNY, was proposed. This work also introduced three provably secure AEAD schemes (PAEF, RPAEF, SAEF) for short messages, achieving better efficiency than the best SKINNY-based AEAD modes. SAEF has also been shown to be OAE [16] and INT-RUP [3] secure [2, 10]. In [1], a variant of forkcipher called multi-forkcipher (MFC) was introduced for larger output lengths. This work also presented a generic CTR mode of called GCTR, based on MFC, achieving better efficiency than traditional CTR-based encryption schemes. In [11], the Eevee family of three AEAD schemes—Umbreon, Jolteon, and Espeon—was proposed based on forkcipher. These schemes are highly parallelizable, suitable for IoT devices, and efficient in MPC for distributed decryption. They achieved cost improvements over SKINNY-based schemes by using ForkSkinny as the underlying primitive family. Additionally, in [6], a Forkcipher-Based Pseudo-Random Number Generator called FCRNG was proposed, based on two forkcipher-based CTR style modes: FCTR-c and FCTR-t. A wide range of pseudorandom function constructions based on forkcipher were proposed in [15]. In [14], an efficient MAC scheme called LightFork, a forkcipher variant of LightMAC, was introduced. Furthermore, in [13], a leakage-resilient two-pass AEAD scheme, called FEDT, was proposed. This scheme demonstrated competitive performance rates compared to those based on tweakable block ciphers. More recently, in [9], two additional leakage-resilient AEAD schemes, ForkDTE1 and ForkDTE2, were introduced, utilizing forkciphers. Notably, FEDT, ForkDTE1, and ForkDTE2 all make use of forkciphers with a large tweak.

These recent works illustrate that, although forkcipher was originally introduced for AEAD schemes targeting short messages, its utility extends beyond that. Forkcipher has become an important primitive for various applications.

1.1 Design approach for Forkcipher

Although the forkcipher is becoming an important cryptographic primitive, the design of secure and efficient forkciphers has not been thoroughly explored. Authors of [4] have used the iterate-fork-iterate paradigm to realize a forkcipher. In which the plain text is encrypted by $r1$ rounds of the cipher. Then the output is "forked" along two parallel paths with $r2$ rounds. Half of the output is considered as the cipher text, while the other half is considered authenticating the message. To the best of our knowledge, there are two existing dedicated forkcipher constructions: ForkAES [5] and ForkSkinny [4]. Both of these design follows the iterate-fork-iterate paradigm. The security of both designs relies on heuristic cryptanalytic results. ForkAES leverages the key schedule and round function of AES-128. Furthermore, it operates as a tweakable block cipher by integrating principles from KIASU-BC [17]. While the authors believe that the security of ForkAES can be reduced to the security of the AES and KIASU ciphers, [7] mounted some attacks on ForkAES. Subsequently, [8] presented an improved attack on the full 10 rounds of ForkAES.

ForkSkinny processes a 128-bit plaintext x, a 64-bit tweak J, and a 128-bit secret key k, and produces two 128-bit ciphertext blocks c_0 and c_1. The initial 21 rounds of ForkSkinny closely mirror those of Skinny, differing primarily in the constant added to the internal state. After these rounds, the encryption splits, with a branch constant XORed into the internal state to facilitate the computation of the two n-bit outputs c_0 and c_1. Post-forking, two separate 27-round iterations of Skinny are executed to derive the final 128-bit outputs (c_0, c_1). The security of ForkSkinny is primarily argued from the security of SKINNY. In [8], it was shown that the best attacks on SKINNY could be extended by one round for most ForkSkinny variants and up to three rounds for ForkSkinny-128-256. While these attacks do not compromise the full-round ForkSkinny, they indicate a security degradation between ForkSkinny and the underlying block cipher.

While forkciphers are developed within the iterate-fork-iterate (IFI) framework, the question of their provable security remains unexplored. The first provably secure forkcipher design was introduced by Kim et al. in [19]. They detailed a method for constructing a forkcipher utilizing public permutations as core elements. This method essentially applies the IFI paradigm to the tweakable Even-Mansour cipher framework. They established that a (1, 1)-round FTEM cipher (where a single-round TEM is first applied to the plaintext, followed by two separate single-round TEM processes) achieved $2n/3$-bit security in the context of the ideal permutation model. However, their security bound is affected by an imbalance between the number of ideal cipher queries and construction queries. Specifically, to allow 2^n ideal cipher queries, the number of construction queries needs to be limited to around $2^{n/2}$. From a practical standpoint, this becomes problematic when the number of ideal cipher queries significantly exceeds the number of construction queries.

Despite ForkSkinny's efficiency and the lack of successful full-round attacks, the observed security degradation raises an important question: *can we design an optimally (n-bit) secure forkcipher with a provable security?* For a secure tweakable forkcipher, for each key and tweak, the functions from the input (X) to each half of the output ($M \rightarrow C_0$ and $M \rightarrow C_1$) should be a permutation, and the corresponding family should be a secure tweakable block cipher (TBC). To address this, we first examine the existing design approaches for TBCs. There are three main approaches: the *Dedicated Approach*, the *Standard Model*, and the *Ideal-Cipher Model*.

Designing TBCs from Block Ciphers in the Standard Model. In this approach, TBCs are designed from underlying block ciphers, with security argued under the assumption that the block cipher is a pseudorandom permutation. This method was introduced by Liskov et al. in [22]. Over the years, several constructions have been proposed, leading to improved security proofs [12, 18, 20, 28].

Designing TBCs from Block Ciphers in the Ideal-Cipher Model. In this approach, TBCs are designed from block ciphers, assuming the underlying block ciphers function as ideal ciphers. Mennink [24] first formally addressed this by proposing two TBC constructions from a block cipher with n-bit tweak, n-bit key, and n-bit data, called $\widetilde{F}[1]$ and $\widetilde{F}[2]$, claiming $2n/3$ bit security and optimal security, respectively. Later, Wang et al. [30] pointed out a birthday attack on $\widetilde{F}[2]$. Later, Mennink prevents this attack by a constant multiplication of the key in [25]. Wang et al. [30] also proposed 32 efficient TBC constructions with n-bit tweak, n-bit key, and n-bit data, achieving optimal security with two block cipher calls. They also mentioned 24 other schemes that achieve optimal security. Additionally, they noted that these schemes are similar to some of the 32 schemes that involve pre-computing a subkey. The $\widetilde{F}[2]$ construction from [25] corresponds to one of the 24 schemes in the framework for the specific parameter values: $a_{11} = 1,\ a_{12} = 0,\ b_{11} = 0,\ b_{12} = 1,\ a_{22} = 1,\ a_{21} = 2,\ a_{23} = 0,\ b_{24} = 1,\ b_{21} = 0,\ b_{31} = 0,\ b_{34} = 1$. Shen and Standaert [29] extended this research by studying TBCs with $2n$-bit tweaks, n-bit key, and n-bit data. They demonstrated that achieving beyond birthday bound security for $2n$-bit tweaks requires more than two block cipher calls and proposed an optimally secure construction using three calls. They conjectured that to build an n-bit secure TBC with tn-bit tweaks where $t > 2$, at least $(t + 1)$ block cipher calls are needed.

In this work, we will focus on the question that *Can we design optimally (n-bit) secure Tweakable Forkcipher from ideal block cipher?*

1.2 Contributions

In this work, we propose the first provable secure forkcipher designs with optimal (n-bit) security based on ideal block ciphers whose inputs and outputs are of size n bits.

- **Forkcipher with n-bit Tweak ($\widetilde{\mathsf{F1}}$):**
  - We introduce a forkcipher design with an n-bit tweak, denoted as $\widetilde{\mathsf{F1}}$.
  - $\widetilde{\mathsf{F1}}$ employs three block ciphers: the first block cipher uses the master key to process the tweak, and the final two parallel block ciphers use derived subkeys to produce a $2n$-bit output.
  - We have proved that $\widetilde{\mathsf{F1}}$ achieves optimal security of n-bits.
- **Forkcipher with $2n$-bit Tweak ($\widetilde{\mathsf{F2}}$):**
  - We propose another forkcipher design with a $2n$-bit tweak, denoted as $\widetilde{\mathsf{F2}}$.
  - $\widetilde{\mathsf{F2}}$ uses four block ciphers: the first two block ciphers use different keys derived from the master key to process the tweak, and the final two parallel block ciphers use derived subkeys to produce a $2n$-bit output.
  - We have also demonstrated that $\widetilde{\mathsf{F2}}$ achieves optimal security of n-bits.
- **Forkcipher with Arbitrary Length Tweak (rn-bit) ($\widetilde{\mathsf{Fr}}$):**
  - For an arbitrary length tweak of rn-bits, we propose a design using $(r+2)$ block ciphers, denoted as $\widetilde{\mathsf{Fr}}$.
  - In $\widetilde{\mathsf{Fr}}$, the first r parallel block ciphers process the tweak, and the final two parallel block ciphers, using derived subkeys, produce a $2n$-bit output.
  - $\widetilde{\mathsf{Fr}}$ also achieves optimal security of n-bits.

It is important to note that for n-bit and $2n$-bit tweaks, our constructions require only one more block cipher than the optimal secure tweakable block cipher (TBC) constructions. Additionally, previous works [24, 29, 30] have shown that these TBC designs are minimal in terms of block cipher usage. Although no optimal secure TBC designs exist for large tweaks (rn), where $r \geq 3$, [29] conjectures that a minimum of $r + 1$ block ciphers is necessary for an optimal secure TBC with an rn-bit tweak. Our design requires only $r + 2$ block ciphers for an rn-bit tweak forkcipher, and the extra block cipher is parallelizable. Moreover, our designs exhibit a similar level of parallelization as TBC designs, demonstrating that our forkciphers are more efficient than two TBCs with the same length tweak.

Table 1: Comparison of $\mathsf{TBC}/\mathsf{TFC}$ construction from ideal Block Cipher. $\mathsf{TBC}/\mathsf{TFC}$ = tweakable block cipher/tweakable forkcipher. TDK = Tweak dependent key

	Tweak length	Security	Block Cipher	TDK
$\tilde{F}[2]$ [24]	n	n-bit	2	1
$\widetilde{E1}, \cdots, \widetilde{E32}$ [30]	n	n-bit	2	1
$\widetilde{\mathsf{F1}}$ [This work]	n	n-bit	3	2
$\tilde{G}2$ [29]	$2n$	n-bit	3	1
$\widetilde{\mathsf{F2}}$ [This work]	$2n$	n-bit	4	2

2 Preliminaries

Notation: An adversary $\mathcal{A}$ is an algorithm. The notation $y \leftarrow \mathcal{A}(x_1, x_2, \ldots, x_i)$ means that $\mathcal{A}$ runs on inputs $x_1, \ldots, x_i$ and produces output y. For a set X, the notation $X \xleftarrow{\cup} x$ means that x is added to X. For bit strings x and y, $x \| y$ denotes their concatenation. The notation $[X]_x$ represents the encoding of a non-negative integer $X < 2^x$ as its x-bit binary representation. For any set X, $x \xleftarrow{\$} X$ denotes that x is chosen uniformly random from X.

When an adversary $\mathcal{A}$ interacts with an oracle $\mathcal{O}$, the output is written as $\mathcal{A}^{\mathcal{O}}$. After this interaction, it returns a bit $b \in \{0,1\}$. We write $\mathcal{A}^{\mathcal{O}} \rightarrow b$ to indicate that $\mathcal{A}$ outputs b after interacting with $\mathcal{O}$. The time complexity of the adversary is defined using the standard RAM (random-access machine) model of computation (see, e.g., [26]).

$\mathcal{P}(\mathcal{M})$ denote the set of all permuatation over $\mathcal{M}$. And let $\widetilde{\mathcal{P}}(\mathcal{T}, \mathcal{M})$ denote the family of all functions $\widetilde{f} : \mathcal{T} \times \mathcal{M} \rightarrow \mathcal{M}$ such that for each $J \in \mathcal{T}$, the function $\widetilde{f}(J, \cdot)$ is a permutation on $\mathcal{M}$. $\mathcal{P}(n)$ and $\widetilde{\mathcal{P}}(\mathcal{T}, n)$ denote the case where $\mathcal{M} = \{0,1\}^n$.

Block Cipher: A block cipher E is a keyed function $\mathsf{E} : \mathcal{K} \times \{0,1\}^n \rightarrow \{0,1\}^n$. The key space is $\mathcal{K}$, and both the domain and range are $\{0,1\}^n$. For each key $k \in \mathcal{K}$, $\mathsf{E}_k \stackrel{\Delta}{=} \mathsf{E}(k, .)$ gives a unique permutation over $\{0,1\}^n$.

We define the PRP security of E based on indistinguishability from a random permutation $\mathsf{P} \xleftarrow{\$} \mathcal{P}(n)$. The block cipher E is called a (q, t, ϵ)-secure pseudo-random permutation(PRP) if any adversary with running time at most t cannot distinguish E_k (for $k \xleftarrow{\$} \mathcal{K}$) from a random permutation P after making at most q queries. The probability of the adversary distinguishing them is at most ϵ. Formally, for any adversary $\mathcal{A}$, the PRP-advantage of $\mathcal{A}$ is defined as:

$$\mathbf{Adv}^{\mathsf{E}}_{\mathsf{PRP}}(\mathcal{A}) \stackrel{\Delta}{=} \left| \Pr[k \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathsf{E}_k(\cdot)} \rightarrow 1] - \Pr[\mathsf{P} \xleftarrow{\$} \mathcal{P}(n) : \mathcal{A}^{\mathsf{P}} \rightarrow 1] \right|$$

Similarly, strong pseudo-random permutation (SPRP) security is defined by its indistinguishability from a random permutation P when the adversary also has access to the inverse oracle (E_k^{-1} or P^{-1}) along with the forward oracle(E_k or

P). The block cipher E is called a (q,t,ϵ)-secure SPRP if any adversary with running time at most t cannot distinguish $(\mathsf{E}_k, \mathsf{E}_k^{-1})$ (for $k \xleftarrow{\$} \mathcal{K}$) from $(\mathsf{P}, \mathsf{P}^{-1})$ after making at most q queries, with a probability of success exceeding ϵ. Formally, for any adversary $\mathcal{A}$, the SPRP-advantage of $\mathcal{A}$ is defined as:

$$\mathbf{Adv}^{\mathsf{E}}_{\mathsf{SPRP}}(\mathcal{A}) \triangleq \left| \Pr[k \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathsf{E}_k(\cdot),\mathsf{E}_k^{-1}(\cdot)} \to 1] - \Pr[\mathsf{P} \xleftarrow{\$} \mathcal{P}(n) : \mathcal{A}^{\mathsf{P}(\cdot),\mathsf{P}^{-1}(\cdot)} \to 1] \right|.$$

We denote $\mathsf{BC}(\mathcal{K}, n)$ as the set of all possible SPRP secure block ciphers with keyspace $\mathcal{K}$ and $\{0,1\}^n$ as the input space.

2.1 Forkciphers

A tweakable forkcipher (TFC) [4] $\widetilde{\mathsf{F}} : \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^{2n}$ is a family of tweakable keyed functions with key space $\mathcal{K}$ and tweak space $\mathcal{T}$. It comprises a pair of deterministic algorithms $(\widetilde{\mathsf{F}}^{+}, \widetilde{\mathsf{F}}^{-})$, defined as follows:

The encryption algorithm

$$\widetilde{\mathsf{F}}^{+} : \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \times \{0,1,2\} \longrightarrow \{0,1\}^n \cup (\{0,1\}^n \times \{0,1\}^n)$$

takes a key $k \in \mathcal{K}$, a tweak $J \in \mathcal{T}$, a message $m \in \{0,1\}^n$, and a selector bit $s \in \{0,1,2\}$ as inputs. It outputs:

$$\widetilde{\mathsf{F}}^{+}(k, J, m, s) = \begin{cases} c_0, & \text{if } s = 0 \\ c_1, & \text{if } s = 1 \\ (c_0, c_1), & \text{if } s = 2 \end{cases}$$

where the ciphertext is $c = (c_0, c_1)$. Here, c_0 represents the left ciphertext block, and c_1 represents the right ciphertext block.

The decryption algorithm

$$\widetilde{\mathsf{F}}^{-} : \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \times \{0,1\} \times \{0,1,2\} \longrightarrow \{0,1\}^n \cup (\{0,1\}^n \times \{0,1\}^n)$$

accepts a key $k \in \mathcal{K}$, a tweak $J \in \mathcal{T}$, a ciphertext block c_b, and a bit $b \in \{0,$ $1\}$ indicating whether c_b is the left or the right ciphertext block, along with a selector bit $s \in \{0,1,2\}$. It outputs:

$$\widetilde{\mathsf{F}}^{-}(k, J, c_b, b, s) = \begin{cases} m & \text{if } s = 0 \\ c_{1-b}, & \text{if } s = 1 \\ (m, c_{1-b}), & \text{if } s = 2 \end{cases}$$

where m denotes the plaintext block.

The correctness of a forkcipher asserts that for any key $k \in \mathcal{K}$, tweak $J \in \mathcal{T}$, plaintext $m \in \{0,1\}^n$, and bit $b \in \{0,1\}$, the following conditions must hold:

1. $\widetilde{\mathsf{F}}^{-}(k, J, \widetilde{\mathsf{F}}^{+}(k, J, m, b), b, 0) = m$,
2. $\widetilde{\mathsf{F}}^{-}(k, J, \widetilde{\mathsf{F}}^{+}(k, J, m, b), b, 1) = \widetilde{\mathsf{F}}^{+}(k, J, m, 1-b)$,
3. $\widetilde{\mathsf{F}}^{-}(k, J, x, b, 2) = (\widetilde{\mathsf{F}}^{-}(k, J, x, b, 0), \widetilde{\mathsf{F}}^{-}(k, J, x, b, 1))$, and
4. $\widetilde{\mathsf{F}}^{+}(k, J, x, 2) = (\widetilde{\mathsf{F}}^{+}(k, J, x, 0), \widetilde{\mathsf{F}}^{+}(k, J, x, 1))$.

For nonempty sets $\mathcal{K}$, $\mathcal{T}$, and $\mathcal{B}$, we define $\mathsf{TFC}(\mathcal{K}, \mathcal{T}, \mathcal{B})$ as the set of all tweakable forkciphers with key space $\mathcal{K}$, tweak space $\mathcal{T}$, and input space $\mathcal{B}$.

Algorithm 1 STFP Game.

Real world

function Initialize
  $k \xleftarrow{\$} \mathcal{K}$

function Oracle $\widetilde{\mathsf{F}}^+_k(J, x, s)$
  return $\widetilde{\mathsf{F}}^+(k, J, x, s)$

function Oracle $\widetilde{\mathsf{F}}^-_k(J, x, b, s))$
  return $\widetilde{\mathsf{F}}^-(k, J, x, b, s)$

Ideal world

function Initialize
  $\mathsf{P}_0, \mathsf{P}_1 \xleftarrow{\$} \widetilde{\mathcal{P}}(\mathcal{T}, n)$

function Oracle $\$^+(J, x, s)$

$$\textbf{return} \begin{cases} \mathsf{P}_0(J, x) & \text{if } s = 0 \\ \mathsf{P}_1(J, x) & \text{if } s = 1 \\ (\mathsf{P}_0(J, x), \mathsf{P}_1(J, x)) & \text{if } s = 2 \end{cases}$$

function Oracle $\$^-(J, y, b, s)$

$$\textbf{return} \begin{cases} \mathsf{P}_b^{-1}(J, y) & \text{if } s = 0 \\ \mathsf{P}_{1\oplus b}(J, \mathsf{P}_b^{-1}(J, y)) & \text{if } s = 1 \\ ((\mathsf{P}_b^{-1}(J, y), \mathsf{P}_{1\oplus b}(J, \mathsf{P}_b^{-1}(J, y)) & \text{if } s = 2 \end{cases}$$

STFP Security of Forkciphers: The security of a Tweakable Forkcipher (TFC) is defined by the traditional notion of indistinguishability between a real oracle (TFC) and an ideal oracle (*tweakable forked permutation*). A *tweakable forked permutation* $\$ \triangleq \{\$^+(\mathsf{P}_0, \mathsf{P}_1), \$^-(\mathsf{P}_0, \mathsf{P}_1)\}$ is constructed of two independent permutations $\mathsf{P}_0, \mathsf{P}_1 \xleftarrow{\$} \widetilde{\mathcal{P}}(\mathcal{T}, n)$ as described in algorithm 1. A formal description of STFP security of a TFC is in algorithm 1.

STFP Security of forkciphers in Ideal cipher model: This work will focus on modular designs of TFC $\widetilde{\mathsf{F}}$ using a block cipher E as the only underlying primitive. For the security of these designs, we will consider the distinguisher having access to either $(\widetilde{\mathsf{F}}^+, \widetilde{\mathsf{F}}^-)$ in real oracle or $(\$^+(\mathsf{P}_0, \mathsf{P}_1), \$^-(\mathsf{P}_0, \mathsf{P}_1))$ in ideal oracle along with access to the underline block cipher E for both the real and ideal oracle, where $\mathsf{P}_0, \mathsf{P}_1 \xleftarrow{\$} \widetilde{\mathcal{P}}(\mathcal{T}, n)$ and tries to distinguish between real and ideal oracle. Moreover, we will consider that these distinguishers have limited resources, such as a maximum q many queries. Finally, for any such distinguisher $\mathcal{D}$, we define the STFP advantage of $\mathcal{D}$ against TFC $\widetilde{\mathsf{F}}$ as follows:

$$\mathbf{Adv}^{\widetilde{\mathsf{F}}}_{\mathsf{STFP}}(\mathcal{D}) \triangleq \left| \Pr[\mathcal{D}^{\widetilde{\mathsf{F}}^+_k, \widetilde{\mathsf{F}}^-_k, \mathsf{E}^\pm} \rightarrow 1] - \Pr[\mathcal{D}^{\$^+(\mathsf{P}_0, \mathsf{P}_1), \$^-(\mathsf{P}_0, \mathsf{P}_1), \mathsf{E}^\pm} \rightarrow 1] \right| \quad (1)$$

where the probabilities are taken over the random choices of $k \xleftarrow{\$} \mathcal{K}$, $\mathsf{E} \xleftarrow{\$} \mathsf{BC}(\mathcal{K}, n)$, and $\mathsf{P}_0, \mathsf{P}_1 \xleftarrow{\$} \widetilde{\mathcal{P}}(\mathcal{T}, n)$. We say that $\widetilde{\mathsf{F}}$ is (q, ϵ)-secure STFP, if the maximum STFP advantage of $\widetilde{\mathsf{F}}$ is ϵ, where the maximum is taken over all distinguisher that makes at most q many queries.

H-Coefficient technique: Consider a computationally unbounded deterministic distinguisher $\mathcal{D}$ that interacts with either the real-world oracle $\mathcal{O}_{re}$ or the ideal-world oracle $\mathcal{O}_{id}$. The set of all queries made by $\mathcal{D}$ and the respective responses received form the transcript τ. In some scenarios, additional internal information might be revealed to $\mathcal{D}$ after completing all its interactions with the oracle but before making its final decision. Let X_{re} and X_{id} denote random variables representing the probability distributions of the transcripts τ generated by the real and ideal oracles, respectively. The probability of observing a particular transcript τ under the ideal oracle, denoted by $\Pr[\mathsf{X}_{id} = \tau]$, is called the ideal interpolation probability. Similarly, the real interpolation probability is defined for the real oracle. A transcript τ is considered *attainable* by $\mathcal{D}$ if $\Pr[X_{id} = \tau] > 0$. We denote the set of all attainable transcripts by $\boldsymbol{\tau}$.

The main theorem of the H-coefficient technique, as detailed in [27], is presented below:

Theorem 1. *Let $\mathcal{D}$ be a deterministic distinguisher with access to either the real oracle $\mathcal{O}_{re}$ or the ideal oracle $\mathcal{O}_{id}$. Let $\boldsymbol{\tau} = \boldsymbol{\tau}_{\mathrm{g}} \sqcup \boldsymbol{\tau}_{\mathrm{b}}$ (disjoint union) be a partition of the set of all attainable transcripts of $\mathcal{D}$. Assume there exists $\epsilon_{\mathsf{good}} \geq 0$ such that for any $\tau \in \boldsymbol{\tau}_{\mathrm{g}}$,*

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

and there exists $\epsilon_{\mathsf{bad}} \geq 0$ such that $\Pr[X_{id} \in \boldsymbol{\tau}_{\mathrm{b}}] \leq \epsilon_{\mathsf{bad}}$. Then,

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

2.2 Birthday Attack on $\widetilde{\mathsf{F1}}$ from [23]

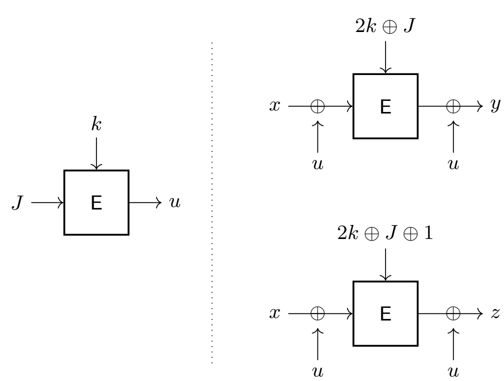

Fig. 1: $\widetilde{\mathsf{F1}}$ [23] : n-bit tweak TFC from 3 BC

Reviewers of ArcticCrypt 2025 identified a birthday distinguishing attack on $\widetilde{\mathsf{F1}}$ from [23], depicted in Figure 1. The attack exploits the fact that the key used in the final block cipher for producing the most significant n-bit output with a tweak J, i.e., $2k \oplus J$, matches with the key used for producing the least significant n-bit output for the tweak $J' = J \oplus 1$, which is $2k \oplus J' \oplus 1$.

By making $2^{\frac{n}{2}}$ queries with messages under both tweaks J and J', we can expect a collision between the (key, input) pairs of two such queries. This leads to a distinguishing attack with birthday queries. Essentially, the original analysis did not account for the possibility that the distinguisher may make multiple queries with each tweak when analyzing the probabilities of `Bad7` and `Bad8`. The concrete attack algorithm is shown below.

Attack Algorithm:

1. Choose any $J \in \{0,1\}^n$.
2. Fix $2^{\frac{n}{2}}$ distinct messages:
$$x_i = [i]_{\frac{n}{2}} \| 0^{\frac{n}{2}}, \quad \forall i \in [1, 2^{\frac{n}{2}}].$$
3. Make construction queries with tweak J and messages x_i for $i = 1, 2, \ldots, 2^{\frac{n}{2}}$. Let the responses be $y_i \| z_i$.
4. Fix another set of $2^{\frac{n}{2}}$ distinct messages:
$$x'_i = 0^{\frac{n}{2}} \| [i]_{\frac{n}{2}}, \quad \forall i \in [1, 2^{\frac{n}{2}}].$$
5. Make construction queries with tweak $J \oplus 1$ and messages x'_i for $i = 1, 2, \ldots, 2^{\frac{n}{2}}$. Let the responses be $y'_i \| z'_i$.
6. Find indices i_1, i_2 such that:
$$y_{i_1} = z'_{i_2}.$$
7. Make a construction query with tweak J and message $x_{i_1} \oplus 1$. Let the response be $y \| z$.
8. Make a construction query with tweak $J \oplus 1$ and message $x'_{i_2} \oplus 1$. Let the response be $y' \| z'$.
9. Return 1 if:
$$y = z'.$$

In this revised draft, we address this issue by modifying the key used for the block cipher producing the least significant n-bit output. Instead of using $2k \oplus J \oplus 1$, we use $4k \oplus J \oplus 1$. This modification restores the claim of optimal n-bit security with a negligible loss of:
$$\frac{4q_cq_p}{2^{2n}} + \frac{4q_c^2}{2^{2n}} + \frac{1}{2^n}.$$

3 Designing TFC with n-bit tweak using three Block Cipher

This section presents a modular design approach for a tweakable forkcipher (TFC) using an ideal block cipher E. We propose a construction called $\widetilde{\mathsf{F1}}$, which takes an n-bit key k, an n-bit tweak J, and an n-bit input x, producing a $2n$-bit

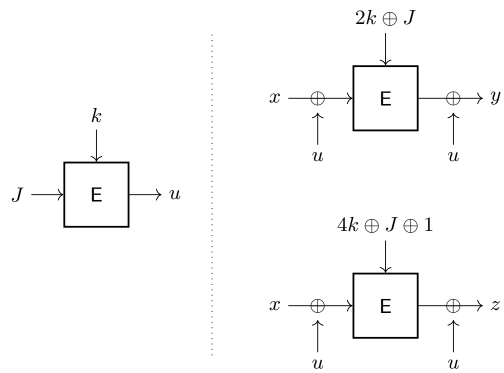

Fig. 2: $\widetilde{\mathsf{F1}}$: n-bit tweak TFC from 3 BC

output $y \| z$. This construction follows a similar design approach as the construction $\widetilde{F}[2]$ of [25].

The construction uses a block cipher with the master key as the key and the tweak as input to obtain an internal value. This internal value is then used to derive the keys for two parallel block ciphers, producing the final $2n$-bit output together. Formally, we define the construction $\widetilde{\mathsf{F1}}$ as follows:

$$\widetilde{\mathsf{F1}}(k, J, x) \triangleq \{\mathsf{E}(2k \oplus J, x \oplus \mathsf{E}(k, J)) \oplus \mathsf{E}(k, J)\} \| \{\mathsf{E}(4k \oplus J \oplus 1, x \oplus \mathsf{E}(k, J)) \oplus \mathsf{E}(k, J)\}.$$

This function is illustrated in Figure 2. The following theorem demonstrates that this construction achieves n-bit security.

Theorem 2. *Let $\mathcal{D}$ be a distinguisher making at most q_c construction queries and q_p ideal cipher queries. Then,*

$$\mathbf{Adv}^{\widetilde{\mathsf{F1}}}_{\mathsf{STFP}}(\mathcal{D}) \leq \frac{2q_c}{2^n} + \frac{q_p}{2^n} + \frac{8q_c q_p}{2^{2n}} + \frac{4q_c^2}{2^{2n}} + \frac{1}{2^n}. \tag{2}$$

Proof. Let $k \xleftarrow{\$} \{0,1\}^n$, $\mathsf{E} \xleftarrow{\$} \mathsf{BC}(\{0,1\}^n, n)$, and $\mathsf{P}_0, \mathsf{P}_1 \xleftarrow{\$} \widetilde{\mathcal{P}}(\{0,1\}^n, n)$. Let $\mathcal{D}$ be a distinguisher with access to one of the following oracles: $(\widetilde{\mathsf{F1}}, \mathsf{E})$ in the real world and $(\$(\mathsf{P}_0, \mathsf{P}_1), \mathsf{E})$ in the ideal world. Note that $\$(\mathsf{P}_0, \mathsf{P}_1)$ behaves exactly as described in algorithm 1. Moreover, $\mathcal{D}$ can make both forward and backward queries. The distinguisher $\mathcal{D}$ makes at most q_c construction queries to $\mathcal{O}_c \in \{\widetilde{\mathsf{F1}}, \$(\mathsf{P}_0, \mathsf{P}_1)\}$. We assume the adversary receives a $2n$-bit output regardless of the distinguisher's choice of selector bit s during the query. This implies that the distinguisher will receive an extra n-bit value along with the desired part, which can only increase the distinguisher's success probability.

We store these construction queries in a transcript as follows:

$$\tau_c = \{(J_1, x_1, y_1 \| z_1), (J_2, x_2, y_2 \| z_2), \ldots, (J_{q_c}, x_{q_c}, y_{q_c} \| z_{q_c})\},$$

where either $\widetilde{\mathsf{F1}}(k, J_i, x_i) = y_i \| z_i$ or $\mathsf{P}_0(J_i, x_i) = y_i$ and $\mathsf{P}_1(J_i, x_i) = z_i$ for all $i = 1, 2, \ldots, q_c$.

We also consider $\mathcal{D}$ making q_p queries to the ideal cipher oracle $\mathcal{O}_p = \mathsf{E}$. We store these queries in a transcript as

$$\tau_p = \{(l_1, a_1, b_1), (l_2, a_2, b_2), \ldots, (l_{q_p}, a_{q_p}, b_{q_p})\},$$

where $\mathsf{E}(l_i, a_i) = b_i$ for all $i = 1, 2, \ldots, q_p$.

After completing all queries to $\mathcal{O}_c$ and $\mathcal{O}_p$, before the decision bit, we reveal the master key k (a randomly chosen fake key k for the ideal world). We also release a tuple (k, J, u) corresponding to each construction query. We store them as

$$\tau_{int} = \{(k, J_1, u_1), (k, J_2, u_2), \ldots, (k, J_{q'_c}, u_{q'_c})\},$$

where $\mathsf{E}(k, J_i) = u_i$ for all $i = 1, 2, \ldots, q'_c$. Here, q'_c is the number of distinct tweaks in τ_c, and thus $q'_c \leq q_c$. Note that this additional information can only increase the advantage of the distinguisher. Thus, the complete transcript is

$$\tau = \{k, \tau_c, \tau_p, \tau_{int}\}.$$

Bad transcript: Definition and Bounds: Next, we will define some bad transcripts that allow the distinguisher to easily distinguish between the real and ideal worlds. The conditions are as follows:

- **Collision with the master key**
  - `Bad1`: $\exists\ (J_i, x_i, y_i \| z_i) \in \tau_c$ such that $2k \oplus J_i = k$ or $4k \oplus J_i \oplus 1 = k$. This condition occurs when, for some query, any one of the derived keys of the final two block ciphers matches with the master key.
  - `Bad2`: $\exists\ (l_i, a_i, b_i) \in \tau_p$ such that $l_i = k$. This condition occurs when the distinguisher queries the ideal cipher with the master key, i.e., the distinguisher can successfully guess the master key k.
- **Collision with ideal cipher query**
  - `Bad3`: $\exists\ (J_i, x_i, y_i \| z_i) \in \tau_c, (l_j, a_j, b_j) \in \tau_p, (k, J_i, u_i) \in \tau_{int}$ such that $2k \oplus J_i = l_j \wedge x_i \oplus u_i = a_j$. This condition occurs when the (key, input) pair of a certain ideal cipher query matches the (key, input) pair of the final block cipher, producing the first n bits of the output for a particular construction query.
  - `Bad4`: $\exists\ (J_i, x_i, y_i \| z_i) \in \tau_c, (l_j, a_j, b_j) \in \tau_p, (k, J_i, u_i) \in \tau_{int}$ such that $4k \oplus J_i \oplus 1 = l_j \wedge x_i \oplus u_i = a_j$. This condition occurs when the (key, input) pair of a certain ideal cipher query matches the (key, input) pair of the final block cipher, producing the second n bits of the output for a particular construction query.
  - `Bad5`: $\exists\ (J_i, x_i, y_i \| z_i) \in \tau_c, (l_j, a_j, b_j) \in \tau_p, (k, J_i, u_i) \in \tau_{int}$ such that $2k \oplus J_i = l_j \wedge y_i \oplus u_i = b_j$. This condition occurs when the (key, output) pair of a certain ideal cipher query matches the (key, output) pair of the final block cipher, producing the first n bits of the output for a particular construction query.

- $\texttt{Bad6}$: $\exists\ (J_i, x_i, y_i \| z_i) \in \tau_c, (l_j, a_j, b_j) \in \tau_p, (k, J_i, u_i) \in \tau_{int}$ such that $4k \oplus J_i \oplus 1 = l_j \wedge z_i \oplus u_i = b_j$. This condition occurs when the (key, output) pair of a certain ideal cipher query matches the (key, output) pair of the final block cipher, producing the second n bits of the output for a particular construction query.

– **Collision between internal block ciphers**

- $\texttt{Bad7}$: $\exists\ (J_i, x_i, y_i \| z_i), (J_j, x_j, y_j \| z_j) \in \tau_c$ and $(k, J_i, u_i), (k, J_j, u_j) \in \tau_{int}$ such that $2k \oplus J_i = 4k \oplus J_j \oplus 1 \wedge x_i \oplus u_i = x_j \oplus u_j$. This condition occurs when the (key, input) pair of the final block cipher, producing the first n bits of the output for one construction query, matches the (key, input) pair of the final block cipher, producing the second n bits of the output for another construction query.
- $\texttt{Bad8}$: $\exists\ (J_i, x_i, y_i \| z_i), (J_j, x_j, y_j \| z_j) \in \tau_c$ and $(k, J_i, u_i), (k, J_j, u_j) \in \tau_{int}$ such that $2k \oplus J_i = 4k \oplus J_j \oplus 1 \wedge y_i \oplus u_i = z_j \oplus u_j$. This condition occurs when the (key, output) pair of the final block cipher, producing the first n bits of the output for one construction query, matches the (key, output) pair of the final block cipher, producing the second n bits of the output for another construction query.
- $\texttt{Bad9}$ $\exists\ (J_i, x_i, y_i \| z_i) \in \tau_c$ and $(k, J_i, u_i) \in \tau_{int}$ such that $2k \oplus J_i = 4k \oplus J_i \oplus 1$. This occurs when two keys of the final pair of block cipher collides for some construction query.

We will call a transcript "$\texttt{Bad}$" if it satisfies any of the above nine conditions. Let τ_b denote the set of all $\texttt{Bad}$ transcripts. In the following lemma, we will show that the probability of these $\texttt{Bad}$ conditions occurring in the ideal world is low.

Lemma 1. *Let* τ_b *denote the set of all bad transcripts and* X_{id} *denotes the random variable of transcript* τ *induced in the ideal world. Then, we have the following:*

$$\Pr[\mathsf{X}_{id} \in \tau_b] \leq \frac{2q_c}{2^n} + \frac{q_p}{2^n} + \frac{8q_cq_p}{2^{2n}} + \frac{4q_c^2}{2^{2n}} + \frac{1}{2^n}\ . \tag{3}$$

Proof. Let us denote the event $\texttt{Bad} = \bigvee_{i=1}^{9} \texttt{Bad}i$. To bound the probability of the event $\texttt{Bad}$, we will first individually bound each $\texttt{Bad}i$ conditioned on the complement of all the previous $\texttt{Bad}j$'s. Then, we will apply the union bound for the final result.

– **Bounding** $\texttt{Bad1}$: This occurs if, for some $i \in [1, q_c]$, $J_i = 2k \oplus k$ or $J_i = 4k \oplus k \oplus 1$. The probability of choosing such a tweak for any i is at most $1/2^n$ due to the randomness of the key k. Hence, for at most q_c choices of i, we have:

$$\Pr[\texttt{Bad1}] \leq \frac{2q_c}{2^n}\ . \tag{4}$$

– **Bounding** $\texttt{Bad2}$: This occurs if the distinguisher $\mathcal{D}$ can guess the master key k among all the ideal cipher queries. Given the randomness of the key and at most q_p ideal cipher queries, we have:

$$\Pr[\texttt{Bad2}] \leq \frac{q_p}{2^n}\ . \tag{5}$$

– **Bounding** $\texttt{Bad3} \mid (\overline{\texttt{Bad1}} \wedge \overline{\texttt{Bad2}})$: This occurs if there exist $i \in [1, q_c]$ and $j \in [1, q_p]$ such that:

$$\begin{aligned} \mathcal{E}1: &\quad 2k = J_i \oplus l_j \\ \mathcal{E}2: &\quad u_i = x_i \oplus a_j \end{aligned}$$

From $\overline{\texttt{Bad1}}$, each u_i is independent of all $y \| z$ values. From $\overline{\texttt{Bad2}}$, the u_i's are independent of all ideal cipher query outputs. So, u_i's are chosen uniformly randomly from a set of at least $2^n - q_c$ many elements. Hence, from at most $q_c q_p$ choices of (i, j) and the randomness of k and u_i, we have:

$$\Pr[\texttt{Bad3} \mid (\overline{\texttt{Bad1}} \wedge \overline{\texttt{Bad2}})] \leq \frac{q_c q_p}{2^n(2^n - q_c)} \leq \frac{2q_c q_p}{2^{2n}} \; . \tag{6}$$

– **Bounding** $\texttt{Bad}i \mid (\overline{\texttt{Bad1}} \wedge \overline{\texttt{Bad2}})$ for $i = 4, 5, 6$: Following a similar argument as the previous case, we have for $i = 4, 5, 6$:

$$\Pr[\texttt{Bad}i \mid (\overline{\texttt{Bad1}} \wedge \overline{\texttt{Bad2}})] \leq \frac{2q_c q_p}{2^{2n}} \; . \tag{7}$$

– **Bounding** $\Pr[\texttt{Bad7} \mid (\overline{\texttt{Bad1}} \wedge \overline{\texttt{Bad2}})]$: The event $\texttt{Bad7}$ occurs if:

$$\begin{aligned} 1) &\quad 2k \oplus J_i = 4k \oplus J_j \oplus 1, \\ 2) &\quad x_i \oplus u_i = x_j \oplus u_j. \end{aligned}$$

We analyze this in two cases:

- **Case 1:** $J_i = J_j$.
In this case, we have $\mathsf{E}_k(J_i) = u_i = u_j = \mathsf{E}_k(J_j)$. Additionally, $x_i \neq x_j$ (as otherwise, the two queries would be identical). Thus, we have $x_i \oplus u_i \neq x_j \oplus u_j$, implying:

$$\Pr[\texttt{Bad7} \mid (\overline{\texttt{Bad1}} \wedge \overline{\texttt{Bad2}} \wedge \text{Case 1})] = 0.$$

- **Case 2:** $J_i \neq J_j$.
Here, we have $u_i \neq u_j$. Using the randomness of k for the first equation and the randomness of u_i due to $\overline{\texttt{Bad1}} \wedge \overline{\texttt{Bad2}}$ for the second equation, we get:

$$\Pr[\texttt{Bad7} \mid (\overline{\texttt{Bad1}} \wedge \overline{\texttt{Bad2}} \wedge \text{Case 2})] \leq \frac{q_c^2}{2^n(2^n - q_c)} \leq \frac{2q_c^2}{2^{2n}}.$$

Combining both cases, we obtain

$$\Pr[\texttt{Bad7} \mid (\overline{\texttt{Bad1}} \wedge \overline{\texttt{Bad2}})] \leq \frac{q_c^2}{2^n(2^n - q_c)} \leq \frac{2q_c^2}{2^{2n}}. \tag{8}$$

– **Bounding** $\texttt{Bad8} \mid (\overline{\texttt{Bad1}} \wedge \overline{\texttt{Bad2}})$: Following a similar argument as the previous case, we have:

$$\Pr[\texttt{Bad8} \mid (\overline{\texttt{Bad1}} \wedge \overline{\texttt{Bad2}})] \leq \frac{2q_c^2}{2^{2n}} \; . \tag{9}$$

– **Bounding** $\texttt{Bad9}$: This occurs if, for some $i \in [q_c]$, we have

$$2k \oplus J_i = 4k \oplus J_i \oplus 1 \implies 2k \oplus 4k = 1.$$

Thus, by the randomness of k, we obtain

$$\Pr[\texttt{Bad9}] \leq \frac{1}{2^n}. \tag{10}$$

Now, from the union bound and using equations (4) to (10), we have:

$$\Pr[\texttt{Bad}] \leq \frac{2q_c}{2^n} + \frac{q_p}{2^n} + \frac{8q_c q_p}{2^{2n}} + \frac{4q_c^2}{2^{2n}} + \frac{1}{2^n} \ . \tag{11}$$

Good Transcript analysis: We will denote all the transcripts that are not "$\texttt{Bad}$" as "$\texttt{Good}$" and let τ_g be the set of all $\texttt{Good}$ transcripts. Let Y_{re} denote the random variable of transcript τ induced in the real world. In this section, we will compute $\Pr[\mathsf{Y}_{re} \in \tau_g] / \Pr[\mathsf{X}_{id} \in \tau_g]$. Now, we will group all the transcripts based on distinct tweaks and keys as follows. For $J \in [0, 2^{n-1}]$ and $l \in [0, 2^{n-1}]$, we have

$$\begin{aligned} \alpha_J &= |\{(J', x', y' \| z') \in \tau_c \mid J' = J\}| \, , \ \forall J \in [0, 2^n - 1] \\ \beta_l &= |\{(l', a', b') \in \tau_p \sqcup \tau_{int} \mid l' = l\}| \, , \ \forall l \in [0, 2^n - 1] \end{aligned}$$

Note that due to $\overline{\texttt{Bad2}}$, $\tau_p \cap \tau_{int} = \phi$. We use the notation β_l to denote the number of block cipher calls with key l, and α_J to denote the number of construction queries with tweak J. Note that, any construction query with tweak $2k \oplus l'$ corresponds to two unique (due to $\overline{\texttt{Bad}}$) block cipher computations with key l' and $l' \oplus 1$. Now, let $\gamma_l = \alpha_{2k \oplus l} + \alpha_{2k \oplus l \oplus 1} + \beta_l$. Clearly, γ_l gives the number of total block cipher calls with key l in real-world.

First, we will compute $\Pr[\mathsf{Y}_{re} \in \tau_g]$:

$$\Pr[\mathsf{Y}_{re} \in \tau_g] = \frac{|\mathsf{Comp}_\mathsf{Y}|}{|\mathsf{All}_\mathsf{Y}|},$$

where Comp_Y is the set of possible transcripts from the real-world oracle compatible with τ_g, and All_Y is the set of all possible transcripts from the real-world oracle. Note that, $|\mathsf{All}_\mathsf{Y}|$ is equal to the number of all possible choices of key k and corresponding ideal cipher block cipher. Hence, $|\mathsf{All}_\mathsf{Y}| = 2^n \times (2^n!)^{2^n}$, where the first 2^n represents the choice of key and the second term represents the number of all possible block ciphers. So, we have the total number of block ciphers compatible with τ_g in the real world, $|\mathsf{Comp}_\mathsf{Y}| = \prod_{l=0}^{2^n-1} (2^n - \gamma_l)!$. Hence,

$$\Pr[\mathsf{Y}_{re} \in \tau_g] = \frac{|\mathsf{Comp}_\mathsf{Y}|}{|\mathsf{All}_\mathsf{Y}|} = \frac{\prod_{l=0}^{2^n-1} (2^n - \gamma_l)!}{2^n \times (2^n!)^{2^n}} \tag{12}$$

Similarly, for the ideal world we will compute $|\mathsf{All}_\mathsf{X}|$ and $|\mathsf{Comp}_\mathsf{X}|$. For the ideal world, we have to compute possible choices for $\mathsf{P}_0, \mathsf{P}_1$, and E. Clearly, $|\mathsf{All}_\mathsf{X}| =$

$2^n \times (2^n!)^{2^n} \times (2^n!)^{2^n} \times (2^n!)^{2^n}$. Here, the first 2^n corresponds to possible choice of key and the rest each $(2^n!)^{2^n}$ terms correspond to the choice of $\mathsf{P}_0, \mathsf{P}_1$, and E. For $\mathsf{Comp}_{\mathsf{X}}$, we will have already decided α_J (input, output) pair of P_0 corresponding to construction queries with tweak J. Moreover, each α_J (input, output) pair is distinct. So, the possible choice of P_0 compatible to τ_g is $\prod_{J=0}^{2^n-1}(2^n - \alpha_J)!$. Following a similar argument, we have a possible choice of P_1 compatible to τ_g is $\prod_{J=0}^{2^n-1}(2^n - \alpha_J)!$. Also, following a similar argument and β_l be the number of block cipher queries with key l, we have the number of possible choices for the underlying block cipher E is $\prod_{l=0}^{2^n-1}(2^n - \beta_l)!$. So, combining all these we have:

$$
\begin{aligned}
|\mathsf{Comp}_{\mathsf{X}}| &= \prod_{J=0}^{2^n-1}(2^n - \alpha_J)! \cdot \prod_{J=0}^{2^n-1}(2^n - \alpha_J)! \cdot \prod_{l=0}^{2^n-1}(2^n - \beta_l)! \\
&= \prod_{J=0}^{2^n-1}(2^n - \alpha_{2k\oplus J})! \cdot \prod_{J=0}^{2^n-1}(2^n - \alpha_{2k\oplus J\oplus 1})! \cdot \prod_{l=0}^{2^n-1}(2^n - \beta_l)! \\
&= \prod_{l=0}^{2^n-1}(2^n - \alpha_{2k\oplus l})! \cdot (2^n - \alpha_{2k\oplus l\oplus 1})! \cdot \prod_{l=0}^{2^n-1}(2^n - \beta_l)! \\
&\overset{[1]}{\leq} \prod_{l=0}^{2^n-1} 2^n! \cdot (2^n - \alpha_{2k\oplus l} - \alpha_{2k\oplus l\oplus 1})! \cdot \prod_{l=0}^{2^n-1}(2^n - \beta_l)! \\
&= (2^n!)^{2^n} \prod_{l=0}^{2^n-1}(2^n - \alpha_{2k\oplus l} - \alpha_{2k\oplus l\oplus 1})! \cdot (2^n - \beta_l)! \\
&\overset{[2]}{\leq} (2^n!)^{2^n} \times (2^n!)^{2^n} \prod_{l=0}^{2^n-1}(2^n - \alpha_{2k\oplus l} - \alpha_{2k\oplus l\oplus 1} - \beta_l)! \\
&= (2^n!)^{2^n} \times (2^n!)^{2^n} \prod_{l=0}^{2^n-1}(2^n - \gamma_l)!
\end{aligned}
$$

[1] and [2] follows from the fact: $(2^n - \delta)! \times (2^n - \mu)! \leq (2^n - \delta - \mu)! \times 2^n!$. So we have

$$
\Pr[\mathsf{X}_{id} \in \tau_g] \leq \frac{|\mathsf{Comp}_{\mathsf{X}}|}{|\mathsf{All}_{\mathsf{X}}|} = \frac{(2^n!)^{2^n} \times (2^n!)^{2^n} \prod_{l=0}^{2^n-1}(2^n - \gamma_l)!}{2^n \times (2^n!)^{2^n} \times (2^n!)^{2^n} \times (2^n!)^{2^n}} \tag{13}
$$

Then combining 12 and 13 we have,

$$
\begin{aligned}
\frac{\Pr[\mathsf{Y}_{re} \in \tau_g]}{\Pr[\mathsf{X}_{id} \in \tau_g]} &= \frac{|\mathsf{All}_{\mathsf{X}}| \times |\mathsf{Comp}_{\mathsf{Y}}|}{|\mathsf{All}_{\mathsf{Y}}| \times |\mathsf{Comp}_{\mathsf{X}}|} \\
&\geq \frac{2^n \times (2^n!)^{2^n} \times (2^n!)^{2^n} \times (2^n!)^{2^n} \times \prod_{l=0}^{2^n-1}(2^n - \gamma_l)!}{2^n \times (2^n!)^{2^n} \times (2^n!)^{2^n} \times (2^n!)^{2^n} \times \prod_{l=0}^{2^n-1}(2^n - \gamma_l)!} \geq 1 \quad (14)
\end{aligned}
$$

Finally, applying theorem 1, lemma 1 and 14 we have the theorem 2.

4 Designing TFC with 2n-bit tweak using four Block Cipher

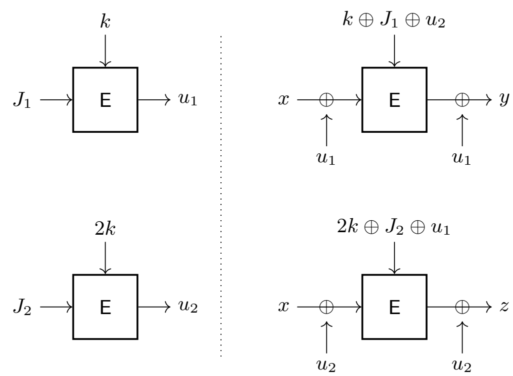

Fig. 3: $\widetilde{\mathsf{F2}}$: 2n-bit tweak TFC from 4 BC

This section presents a construction called $\widetilde{\mathsf{F2}}$, which takes an n-bit key k, an $2n$-bit tweak $J = J_1 \| J_2$, and an n-bit input x, producing a $2n$-bit output $y \| z$. The construction first uses two block ciphers with key k, and $2k$ respectively, taking input J_1, and J_2 respectively. Then, the output of these two ideal block ciphers is used to derive the input and key of the final two block ciphers outputting the final $2n$-bit output. This design follows a similar approach as the construction $\widetilde{G}2$ of [29]. Formally, we define the construction $\widetilde{\mathsf{F2}}$ as follows:

$$\begin{aligned}\widetilde{\mathsf{F2}}(k, J_1 \| J_2, x) \overset{\Delta}{=} \; & \mathsf{E}(k \oplus J_1 \oplus \mathsf{E}(2k, J_2), x \oplus \mathsf{E}(k, J_1)) \oplus \mathsf{E}(k, J_1) \; \| \\ & \mathsf{E}(2k \oplus J_2 \oplus \mathsf{E}(k, J_1), x \oplus \mathsf{E}(2k, J_2)) \oplus \mathsf{E}(2k, J_2).\end{aligned}$$

This function is illustrated in Figure 3. The following theorem demonstrates that this construction achieves n-bit security.

Theorem 3. *Let $\mathcal{D}$ be a distinguisher making at most q_c construction queries and q_p ideal cipher queries. Then,*

$$\mathbf{Adv}^{\widetilde{\mathsf{F2}}}_{\mathsf{STFP}}(\mathcal{D}) \leq \frac{10q_c}{2^n} + \frac{2q_p}{2^n} + \frac{24q_c^2}{2^{2n}} + \frac{8q_cq_p}{2^{2n}}.$$

Where $q_c \leq 2^{n-1}$.

Proof. Let $k \xleftarrow{\$} \{0,1\}^n$, $\mathsf{E} \xleftarrow{\$} \mathsf{BC}(\{0,1\}^n, n)$, and $\mathsf{P}_0, \mathsf{P}_1 \xleftarrow{\$} \widetilde{\mathcal{P}}(\{0,1\}^{2n}, n)$. Let $\mathcal{D}$ be a distinguisher with access to one of the following oracles: $(\widetilde{\mathsf{F2}}, \mathsf{E})$ in the real world and $(\$(\mathsf{P}_0, \mathsf{P}_1), \mathsf{E})$ in the ideal world. Note that $\$(\mathsf{P}_0, \mathsf{P}_1)$ behaves exactly as described in algorithm 1. Moreover, $\mathcal{D}$ can make both forward and backward queries. The distinguisher $\mathcal{D}$ makes at most q_c construction queries to $\mathcal{O}_c \in \{\widetilde{\mathsf{F2}}, \$(\mathsf{P}_0, \mathsf{P}_1)\}$. We assume the adversary receives a $2n$-bit output regardless of the distinguisher's choice of selector bit s during the query. This implies that the distinguisher will receive an extra n-bit value along with the desired part, which can only increase the distinguisher's success probability. We store these construction queries in a transcript as follows:

$$\tau_c = \{(J_1^1 \| J_2^1, x_1, y_1 \| z_1), (J_1^2 \| J_2^2, x_2, y_2 \| z_2), \ldots, (J_1^{q_c} \| J_2^{q_c}, x_{q_c}, y_{q_c} \| z_{q_c})\},$$

where either $\widetilde{\mathsf{F2}}(k, J_1^i \| J_2^i, x_i) = y_i \| z_i$ or $\mathsf{P}_0(J_1^i \| J_2^i, x_i) = y_i$ and $\mathsf{P}_1(J_1^i \| J_2^i, x_i) = z_i$ for all $i = 1, 2, \ldots, q_c$.

We also consider $\mathcal{D}$ making q_p queries to the ideal cipher oracle $\mathcal{O}_p = \mathsf{E}$. We store these queries in a transcript as

$$\tau_p = \{(l_1, a_1, b_1), (l_2, a_2, b_2), \ldots, (l_{q_p}, a_{q_p}, b_{q_p})\},$$

where $\mathsf{E}(l_i, a_i) = b_i$ for all $i = 1, 2, \ldots, q_p$. After completing all queries to $\mathcal{O}_c$ and $\mathcal{O}_p$, and before making the decision, we reveal the master key k (or a randomly chosen fake key k in the ideal world). Additionally, we release two tuples (k, J_1, u_1) and $(2k, J_2, u_2)$ corresponding to each construction query with tweak $J = J_1 \| J_2$. We store them as:

$$\tau_{int}^1 = \{(k, J_1^1, u_1^1), (k, J_1^2, u_1^2), \ldots, (k, J_1^{q_c'}, u_1^{q_c'})\},$$
$$\tau_{int}^2 = \{(2k, J_2^1, u_2^1), (2k, J_2^2, u_2^2), \ldots, (2k, J_2^{q_c''}, u_2^{q_c''})\},$$

where $\mathsf{E}(k, J_1^i) = u_1^i$ for all $i = 1, 2, \ldots, q_c'$ and $\mathsf{E}(2k, J_2^j) = u_2^j$ for all $j = 1, 2, \ldots, q_c''$. Here, q_c' and q_c'' are the numbers of tweaks with distinct values in the left and right n-bits of τ_c, respectively, and both satisfy $q_c', q_c'' \leq q_c$. Note that this additional information can only increase the advantage of the distinguisher. Thus, the complete transcript is

$$\tau = \{k, \tau_c, \tau_p, \tau_{int}^1, \tau_{int}^2\}.$$

Bad transcript: Definition and Bounds: Next, we will define some bad transcripts that enable the distinguisher to differentiate between the real and ideal worlds easily. The conditions are as follows:

- **Collision with ideal cipher query**
  - `Bad1`: There exists $(l_i, *, *) \in \tau_p$ such that l_i is equal to k or $2k$. This occurs when the distinguisher correctly guesses the master key for an ideal cipher query.
  - `Bad2`: There exist $(J_1^i \| J_2^i, x_i, y_i \| z_i) \in \tau_c$, $(k, J_1^i, u_1^i) \in \tau_{int}^1$, and $(2k, J_2^i, u_2^i) \in \tau_{int}^2$, as well as an entry $(l_j, a_j, b_j) \in \tau_p$, such that $k \oplus J_1^i \oplus u_2^i = l_j$ and $x_i \oplus u_1^i = a_j$. This arises when the (key, input) of internal block cipher generating the first n bits of the final output during a construction query matches that of an ideal cipher query.

- $\mathtt{Bad3}$: There exist $(J_1^i \| J_2^i, x_i, y_i \| z_i) \in \tau_c,\ (k, J_1^i, u_1^i) \in \tau_{int}^1,\ (2k, J_2^i, u_2^i) \in \tau_{int}^2$, and $(l_j, a_j, b_j) \in \tau_p$ such that $k \oplus J_1^i \oplus u_2^i = l_j \wedge y_i \oplus u_1^i = b_j$. This arises when the (key, output) of internal block cipher generating the first n bits of the final output during a construction query matches that of an ideal cipher query.
- $\mathtt{Bad4}$: There exists $(J_1^i \| J_2^i, x_i, y_i \| z_i) \in \tau_c, (k, J_1^i, u_1^i) \in \tau_{int}^1, (2k, J_2^i, u_2^i) \in \tau_{int}^2$, and $(l_j, a_j, b_j) \in \tau_p$ such that $2k \oplus J_2^i \oplus u_1^i = l_j$ and $x_i \oplus u_2^i = a_j$. This arises when the (key, input) pair of internal block cipher generating the second n bits of the final output during a construction query matches that of an ideal cipher query.
- $\mathtt{Bad5}$: $\exists (J_1^i \| J_2^i, x_i, y_i \| z_i) \in \tau_c,\ (k, J_1^i, u_1^i) \in \tau_{int}^1,\ (2k, J_2^i, u_2^i) \in \tau_{int}^2$, and $(l_j, a_j, b_j) \in \tau_p$ such that $2k \oplus J_2^i \oplus u_1^i = l_j \wedge z_i \oplus u_2^i = b_j$. This arises when the (key, output) pair of internal block cipher generating the second n bits of the final output during a construction query matches that of an ideal cipher query.

– **Collision between master key and internal key**
  - $\mathtt{Bad6}$: $\exists (J_1^i \| J_2^i, x_i, y_i \| z_i) \in \tau_c,\ (k, J_1^i, u_1^i) \in \tau_{int}^1,\ (2k, J_2^i, u_2^i) \in \tau_{int}^2$ such that $k \oplus J_1^i \oplus u_2^i = k$ or $2k \oplus J_2^i \oplus u_1^i = k$. This happens if, for some construction query, one of the two derived subkeys collides with the master key.
  - $\mathtt{Bad7}$: $\exists (J_1^i \| J_2^i, x_i, y_i \| z_i) \in \tau_c,\ (k, J_1^i, u_1^i) \in \tau_{int}^1,\ (2k, J_2^i, u_2^i) \in \tau_{int}^2$ such that $k \oplus J_1^i \oplus u_2^i = 2k$ or $2k \oplus J_2^i \oplus u_1^i = 2k$. This happens if, for some construction query, one of the two derived subkeys collides with $2k$.

– **Collision between keys of final two block cipher(same construction query)**
  - $\mathtt{Bad8}$: $\exists (J_1^i \| J_2^i, x_i, y_i \| z_i) \in \tau_c,\ (k, J_1^i, u_1^i) \in \tau_{int}^1,\ (2k, J_2^i, u_2^i) \in \tau_{int}^2$ such that $k \oplus J_1^i \oplus u_2^i = 2k \oplus J_2^i \oplus u_1^i$. This occurs if two keys correspond to the final two block ciphers producing the output of a construction query, resulting in a collision.

– **Collision between key, input, output of final two block cipher(different construction query)**
  - $\mathtt{Bad9}$: $\exists (J_1^i \| J_2^i, x_i, y_i \| z_i) \in \tau_c,\ (k, J_1^i, u_1^i) \in \tau_{int}^1,\ (2k, J_2^i, u_2^i) \in \tau_{int}^2$ and $\exists (J_1^j \| J_2^j,\ x_j, y_j \| z_j) \in \tau_c,\ (k, J_1^j, u_1^j) \in \tau_{int}^1,\ (2k, J_2^j, u_2^j) \in \tau_{int}^2$ such that $(k \oplus J_1^i \oplus u_2^i = k \oplus J_1^j \oplus u_2^j) \wedge (x_i \oplus u_1^i = x_j \oplus u_1^j)$. This occurs if the (key, input) pair of the block cipher producing the first n-bits of the output correspond to two construction query matches.
  - $\mathtt{Bad10}$: $\exists (J_1^i \| J_2^i, x_i, y_i \| z_i) \in \tau_c,\ (k, J_1^i, u_1^i) \in \tau_{int}^1,\ (2k, J_2^i, u_2^i) \in \tau_{int}^2$ and $\exists (J_1^j \| J_2^j,\ x_j, y_j \| z_j) \in \tau_c,\ (k, J_1^j, u_1^j) \in \tau_{int}^1,\ (2k, J_2^j, u_2^j) \in \tau_{int}^2$ such that $(k \oplus J_1^i \oplus u_2^i = k \oplus J_1^j \oplus u_2^j) \wedge (y_i \oplus u_1^i = y_j \oplus u_1^j)$. This occurs if the (key, output) pair of the block cipher producing the first n-bits of the output correspond to two construction query matches.
  - $\mathtt{Bad11}$: $\exists (J_1^i \| J_2^i, x_i, y_i \| z_i) \in \tau_c,\ (k, J_1^i, u_1^i) \in \tau_{int}^1,\ (2k, J_2^i, u_2^i) \in \tau_{int}^2$ and $\exists (J_1^j \| J_2^j,\ x_j, y_j \| z_j) \in \tau_c,\ (k, J_1^j, u_1^j) \in \tau_{int}^1,\ (2k, J_2^j, u_2^j) \in \tau_{int}^2$ such that $(2k \oplus J_2^i \oplus u_1^i = 2k \oplus J_2^j \oplus u_1^j) \wedge (x_i \oplus u_2^i = x_j \oplus u_2^j)$. This occurs if the (key, input) pair of the block cipher producing the second n-bits of the output correspond to two construction query matches.

- $\texttt{Bad12}$: $\exists(J_1^i \| J_2^i, x_i, y_i \| z_i) \in \tau_c,\ (k, J_1^i, u_1^i) \in \tau_{int}^1,\ (2k, J_2^i, u_2^i) \in \tau_{int}^2$ and $\exists(J_1^j \| J_2^j,\ x_j, y_j \| z_j) \in \tau_c,\ (k, J_1^j, u_1^j) \in \tau_{int}^1,\ (2k, J_2^j, u_2^j) \in \tau_{int}^2$ such that $(2k \oplus J_2^i \oplus u_1^i = 2k \oplus J_2^j \oplus u_1^j) \wedge (z_i \oplus u_2^i = z_j \oplus u_2^j)$. This occurs if the (key, output) pair of the block cipher producing the second n-bits of the output correspond to two construction query matches.
- $\texttt{Bad13}$: $\exists(J_1^i \| J_2^i, x_i, y_i \| z_i) \in \tau_c,\ (k, J_1^i, u_1^i) \in \tau_{int}^1,\ (2k, J_2^i, u_2^i) \in \tau_{int}^2$ and $\exists(J_1^j \| J_2^j,\ x_j, y_j \| z_j) \in \tau_c,\ (k, J_1^j, u_1^j) \in \tau_{int}^1,\ (2k, J_2^j, u_2^j) \in \tau_{int}^2$ such that $(k \oplus J_1^i \oplus u_2^i = 2k \oplus J_2^j \oplus u_1^j) \wedge (x_i \oplus u_1^i = x_j \oplus u_2^j)$. This occurs if the (key, input) pair of the block cipher producing the first n-bits of the final output in one construction query collides with the (key, input) pair of the block cipher producing the second n-bits of the final output in another construction query.
- $\texttt{Bad14}$: $\exists(J_1^i \| J_2^i, x_i, y_i \| z_i) \in \tau_c,\ (k, J_1^i, u_1^i) \in \tau_{int}^1,\ (2k, J_2^i, u_2^i) \in \tau_{int}^2$ and $\exists(J_1^j \| J_2^j,\ x_j, y_j \| z_j) \in \tau_c,\ (k, J_1^j, u_1^j) \in \tau_{int}^1,\ (2k, J_2^j, u_2^j) \in \tau_{int}^2$ such that $(k \oplus J_1^i \oplus u_2^i = 2k \oplus J_2^j \oplus u_1^j) \wedge (y_i \oplus u_1^i = z_j \oplus u_2^j)$. This occurs if the (key, output) pair of the block cipher producing the first n-bits of the final output in one construction query collides with the (key, output) pair of the block cipher producing the second n-bits of the final output in another construction query.

Similar to the proof of $\widetilde{\mathsf{F1}}$, we define a transcript as "$\texttt{Bad}$" if it satisfies any of the 14 conditions mentioned above. Let τ_{b} denote the set of all $\texttt{Bad}$ transcripts. In the following lemma, we will demonstrate that the probability of these $\texttt{Bad}$ conditions occurring in the ideal world is low.

Lemma 2. *Let τ_{b} denote the set of all bad transcripts and X_{id} denotes the random variable of transcript τ induced in the ideal world. Then, we have the following:*

$$\Pr[\mathsf{X}_{id} \in \tau_{\mathrm{b}}] \leq \frac{9q_c}{2^n} + \frac{2q_p}{2^n} + \frac{20q_c^2}{2^{2n}} + \frac{8q_cq_p}{2^{2n}}\ ,$$

where $q_c \leq 2^{n-1}$.

Proof. Let us denote the event $\texttt{Bad} = \bigvee_{i=1}^{14} \texttt{Bad}i$. To bound the probability of the event $\texttt{Bad}$, we will first individually bound each $\texttt{Bad}i$ conditioned on the complement of $\texttt{Bad1}$. Then, we will apply the union bound to obtain the final result.

- **Bounding** $\texttt{Bad1}$: This occurs if the distinguisher can guess either the key k or $2k$. Hence, considering the randomness of the master key k and at most q_p ideal cipher queries, we have:

$$\Pr[\texttt{Bad1}] \leq \frac{2q_p}{2^n}. \tag{15}$$

- **Bounding** $\texttt{Bad2} \mid \overline{\texttt{Bad1}}$: Note that $\overline{\texttt{Bad1}}$ ensures that each u_1^i is chosen uniformly from a set of at least $2^n - q_c$ elements. Moreover, there are at most $q_c \cdot q_p$ pairs (i, j). Therefore, considering the randomness of k and u_1^i, we have:

$$\Pr[\texttt{Bad2} \mid \overline{\texttt{Bad1}}] \leq \frac{q_c \cdot q_p}{2^n(2^n - q_c)} \leq \frac{2q_c \cdot q_p}{2^{2n}}. \tag{16}$$

– **Bounding** $\texttt{Bad}l \mid \overline{\texttt{Bad1}}$ for $l = 3, 4, 5$: Following a similar argument as the previous case, we have

$$\Pr\left[\bigvee_{l=3}^{5}(\texttt{Bad}l \mid \overline{\texttt{Bad1}})\right] \leq \frac{6q_c \cdot q_p}{2^{2n}}. \tag{17}$$

– **Bounding** $\texttt{Bad6} \mid \overline{\texttt{Bad1}}$: This occurs if the distinguisher can find a tweak $J_i = J_1^i \| J_2^i$ such that either (1) $J_1^i \oplus u_2^i = 0^n$ or (2) $J_2^i \oplus u_1^i = k \oplus 2k$. Moreover, due to $\overline{\texttt{Bad1}}$, each u_1^i is chosen uniformly from a set of at least $2^n - q_c$ elements, and the same applies to u_2^i. Thus, for at most q_c choices of i, we have:

$$\Pr[\texttt{Bad6} \mid \overline{\texttt{Bad1}}] \leq \frac{2q_c}{2^n - q_c} \leq \frac{4q_c}{2^n}. \tag{18}$$

– **Bounding** $\texttt{Bad7} \mid \overline{\texttt{Bad1}}$: Following a similar argument as the previous case for $\texttt{Bad6}$, we have:

$$\Pr[\texttt{Bad7} \mid \overline{\texttt{Bad1}}] \leq \frac{4q_c}{2^n}. \tag{19}$$

– **Bounding** $\texttt{Bad8} \mid \overline{\texttt{Bad1}}$: This occurs if the adversary can find a tweak value $J = J_1^i \| J_2^i$ satisfying $J_1^i \oplus u_2^i \oplus J_2^i \oplus u_1^i = k \oplus 2k$. So, from the randomness of key k, we have:

$$\Pr[\texttt{Bad8} \mid \overline{\texttt{Bad1}}] \leq \frac{q_c}{2^n}. \tag{20}$$

– **Bounding** $\texttt{Bad9} \mid \overline{\texttt{Bad1}}$: This occurs if the distinguisher can find two construction queries $(J_i = J_1^i \| J_2^i, x_i)$ and $(J_j = J_1^j \| J_2^j, x_j)$ such that: 1) $J_1^i \oplus u_2^i = J_1^j \oplus u_2^j$, and 2) $x_i \oplus u_1^i = x_j \oplus u_1^j$.
If $J_1^i = J_1^j$, $J_2^i = J_2^j$, or $x_i = x_j$, the probability of this event is 0. Otherwise, from a similar argument as before, considering the randomness of u_2^i and u_1^i, we have:

$$\Pr[\texttt{Bad9} \mid \overline{\texttt{Bad1}}] \leq \frac{q_c^2}{(2^n - q_c)(2^n - q_c)} \leq \frac{4q_c^2}{2^{2n}}. \tag{21}$$

– **Bounding** $\texttt{Bad}l \mid \overline{\texttt{Bad1}}$ for $l = 10, 11, 12$: Following a similar argument as the previous case, we have

$$\Pr\left[\bigvee_{l=10}^{12} \texttt{Bad}l \mid \overline{\texttt{Bad1}}\right] \leq \frac{12q_c^2}{2^{2n}}. \tag{22}$$

– **Bounding** $\texttt{Bad13} \mid \overline{\texttt{Bad1}}$: This occurs if: $\mathcal{E}1 : J_1^i \oplus u_2^i \oplus J_2^j \oplus u_1^j = k \oplus 2k$, and $\mathcal{E}2 : x_i \oplus x_j = u_1^i \oplus u_2^j$. Note that u_1^i is independently chosen from u_2^i and u_2^j, as u_1 and u_2 values are outputs of E with two different keys k and

$2k$ respectively. Moreover, due to $\overline{\texttt{Bad1}}$, we have the randomness of u_1^i, u_2^i and k. Therefore, we have:

$$\Pr[\texttt{Bad13} \mid \overline{\texttt{Bad1}}] \leq \frac{q_c^2}{2^n(2^n - q_c)} \leq \frac{2q_c^2}{2^{2n}}. \tag{23}$$

– **Bounding** $\texttt{Bad14} \mid \overline{\texttt{Bad1}}$: Following a similar argument as the previous case, we have:

$$\Pr[\texttt{Bad14} \mid \overline{\texttt{Bad1}}] \leq \frac{2q_c^2}{2^{2n}}. \tag{24}$$

Now, from the union bound and using equations (15) to (24), we have:

$$\Pr[\texttt{Bad}] \leq \frac{9q_c}{2^n} + \frac{2q_p}{2^n} + \frac{20q_c^2}{2^{2n}} + \frac{8q_cq_p}{2^{2n}} \ . \tag{25}$$

Good Transcript analysis: We will denote all the transcripts that are not "Bad" as "Good" and let τ_g be the set of all Good transcripts. Let Y_{re} denote the random variable of transcript τ induced in the real world. In this section, we will compute $\Pr[\mathsf{Y}_{re} \in \tau_g]/\Pr[\mathsf{X}_{id} \in \tau_g]$. For this, we will first define some set for partitioning all the query responses depending on keys and tweaks as follows:

– For the master key k, let $S_1(k)$ denote the set of all revealed ideal cipher (input, output) pairs corresponding to the key k. Formally, we define $S_1(k) = \tau_{int}^1$.
– For the master key k, let $S_2(2k)$ denote the set of all revealed ideal cipher (input, output) pairs corresponding to the key $2k$. Formally, we define $S_2(2k) = \tau_{int}^2$.
– Let $S_3(K)$ denote the set of all ideal cipher queries with key K. Formally, we define $S_3(K) = \{(l, *, *) \in \tau_p \mid l = K\}$ for any $K \in \{0,1\}^n$.
– Let $S_4(J)$ denote the set of all construction queries with the tweak J. Formally, we define $S_4(J) = \{(J, *, *\|*) \in \tau_c\}$, for all $J \in \{0,1\}^{2n}$.
– Let $S_5(K)$ denote the set of all tuples corresponding to the final two blocks that produce the final output with the key K. Formally, we define

$$\begin{aligned} S_5(K) = \{(k \oplus J_1 \oplus u_2, x \oplus u_1, y \oplus u_1) : (J_1\|J_2, x, y\|z) \in \tau_c \wedge K = k \oplus J_1 \oplus u_2\} \cup \\ \{(2k \oplus J_2 \oplus u_1, x \oplus u_2, z \oplus u_2) : (J_1\|J_2, x, y\|z) \in \tau_c \wedge K = 2k \oplus J_2 \oplus u_1\}\,. \end{aligned}$$

Due to $\overline{\texttt{Bad1}}$, we have the following for all $K \in \{0,1\}^n$:

$$S_1(k) \cap S_3(K) = \emptyset$$
$$S_2(2k) \cap S_3(K) = \emptyset$$

Similarly, due to $\overline{\texttt{Bad6}}$ and $\overline{\texttt{Bad7}}$, we have for any $K \in \{0,1\}^n$:

$$S_1(k) \cap S_5(K) = \emptyset$$
$$S_2(2k) \cap S_5(K) = \emptyset$$

Additionally, due to $\overline{\texttt{Bad2}} - \overline{\texttt{Bad5}}$, we have for any $K \in \{0,1\}^n$:

$$S_3(K) \cap S_5(K) = \emptyset$$

Moreover, due to $\overline{\texttt{Bad8}} - \overline{\texttt{Bad14}}$, $S_5(K)$ has no duplicate elements. This implies that each construction query contributes exactly two elements to $\bigcup_{K=0}^{2^n-1} S_5(K)$. Therefore, we have:

$$\sum_{K=0}^{2^n-1} |S_5(K)| = \sum_{J=0}^{2^{2n}-1} \left(|S_4(J)| + |S_4(J)|\right).$$

In the real world, we have a total of $|S_3(K)|+|S_5(K)|$ (input, output) pairs of the ideal cipher corresponding to the key K, where $K \neq k$ and $K \neq 2k$. Additionally, we have $|S_1(k)|$ (input, output) pairs of the ideal cipher corresponding to the key k and $|S_2(2k)|$ (input, output) pairs of the ideal cipher corresponding to the key $2k$. Therefore, we have:

$$\Pr[\mathsf{Y}_{re} \in \tau_{\mathrm{g}}] = \frac{1}{2^n} \cdot \prod_{i=0}^{|S_1(k)|-1} \frac{1}{2^n - i} \cdot \prod_{j=0}^{|S_2(k)|-1} \frac{1}{2^n - j} \cdot \prod_{K=0}^{2^n-1} \prod_{l=0}^{|S_3(K)|+|S_5(K)|-1} \frac{1}{2^n - l} \tag{26}$$

Now in the ideal world, we have a total of $|S_3(K)|$ (input, output) pairs of the ideal cipher corresponding to the key K, where $K \neq k$ and $K \neq 2k$. Additionally, we have $|S_1(k)|$ (input, output) pairs of the ideal cipher corresponding to the key k and $|S_2(2k)|$ (input, output) pairs of the ideal cipher corresponding to the key $2k$. Moreover, there is $S_4(J)$ many (input, output) pair correspond to both P_0 and P_1 for any tweak $J \in \{0,1\}^{2n}$. Hence,

$$\begin{aligned}
\Pr[\mathsf{X}_{id} \in \tau_{\mathrm{g}}] &= \frac{1}{2^n} \cdot \prod_{i=0}^{|S_1(k)|-1} \frac{1}{2^n - i} \cdot \prod_{j=0}^{|S_2(k)|-1} \frac{1}{2^n - j} \cdot \prod_{K=0}^{2^n-1} \prod_{l=0}^{|S_3(K)|-1} \frac{1}{2^n - l} \cdot \\
&\quad \prod_{J=0}^{2^{2n}-1} \prod_{s=0}^{|S_4(J)|-1} \frac{1}{2^n - s} \cdot \prod_{J=0}^{2^{2n}-1} \prod_{r=0}^{|S_4(J)|-1} \frac{1}{2^n - r} \\
&\overset{[1]}{\leq} \frac{1}{2^n} \cdot \prod_{i=0}^{|S_1(k)|-1} \frac{1}{2^n - i} \cdot \prod_{j=0}^{|S_2(k)|-1} \frac{1}{2^n - j} \cdot \prod_{K=0}^{2^n-1} \prod_{l=0}^{|S_3(K)|-1} \frac{1}{2^n - l} \cdot \\
&\quad \prod_{J=0}^{2^{2n}-1} \prod_{s=0}^{|S_4(J)|+|S_4(J)|-1} \frac{1}{2^n - s} \\
&\overset{[2]}{\leq} \frac{1}{2^n} \cdot \prod_{i=0}^{|S_1(k)|-1} \frac{1}{2^n - i} \cdot \prod_{j=0}^{|S_2(k)|-1} \frac{1}{2^n - j} \cdot \prod_{K=0}^{2^n-1} \prod_{l=0}^{|S_3(K)|-1} \frac{1}{2^n - l} \cdot \\
&\quad \prod_{K=0}^{2^n-1} \prod_{s=0}^{|S_5(K)|-1} \frac{1}{2^n - s} \\
&\overset{[3]}{\leq} \frac{1}{2^n} \cdot \prod_{i=0}^{|S_1(k)|-1} \frac{1}{2^n - i} \cdot \prod_{j=0}^{|S_2(k)|-1} \frac{1}{2^n - j} \cdot \prod_{K=0}^{2^n-1} \prod_{l=0}^{|S_3(K)|+|S_5(K)|-1} \frac{1}{2^n - l}
\end{aligned} \tag{27}$$

Here, inequalities [1], [2], and [3] follow from the facts 1, 2, and 3, respectively.

1. $\prod_{J=0}^{2^{2n}-1}\prod_{s=0}^{|S_4(J)|-1}\frac{1}{2^n-s}\cdot\prod_{J=0}^{2^{2n}-1}\prod_{r=0}^{|S_4(J)|-1}\frac{1}{2^n-r}\leq\prod_{J=0}^{2^{2n}-1}\prod_{s=0}^{|S_4(J)|+|S_4(J)|-1}\frac{1}{2^n-s}$.
2. $\prod_{J=0}^{2^{2n}-1}\prod_{s=0}^{|S_4(J)|+|S_4(J)|-1}\frac{1}{2^n-s}\leq\prod_{K=0}^{2^n-1}\prod_{s=0}^{|S_5(K)|-1}\frac{1}{2^n-s}$.
3. $\prod_{l=0}^{|S_3(K)|-1}\frac{1}{2^n-l}\cdot\prod_{K=0}^{2^n-1}\prod_{s=0}^{|S_5(K)|-1}\frac{1}{2^n-s}\leq\prod_{K=0}^{2^n-1}\prod_{l=0}^{|S_3(K)|+|S_5(K)|-1}\frac{1}{2^n-l}$.

So, from 26 and 27 we have

$$\frac{\Pr[\mathsf{Y}_{re}\in\tau_{\mathrm{g}}]}{\Pr[\mathsf{X}_{id}\in\tau_{\mathrm{g}}]}\geq 1.$$

5 Designing TFC with rn-bit tweak using (r+2) Block Cipher

Let $\mathsf{E}\xleftarrow{\$}\mathsf{BC}(\{0,1\}^n,n)$ be an n-bit block cipher. The tweakable forkcipher $\widetilde{\mathsf{Fr}}:0,1^n\times 0,1^{rn}\times 0,1^n\rightarrow 0,1^{2n}$, with an rn-bit tweak and using $(r+2)$ block cipher calls, is constructed as follows: First, r block cipher calls are invoked in parallel to produce r masks $u_1,u_2,\ldots,u_r$ from the tweaks $J_1,J_2,\ldots,J_r$ and the master key k. By using $\sum_{i=1}^{r}u_i$ to mask the input and output, and $\sum_{i=1}^{r}2^{i-1}u_i$ to provide variety in the sub-key, another block cipher call is made to encrypt the plaintext x into the left n-bit ciphertext y. Similarly, by using $\sum_{i=1}^{r}2^{i-1}u_i$ to mask the input and output, and $\sum_{i=1}^{r}u_i$ to provide variety in the sub-key, another parallel block cipher call is made to encrypt the plaintext x into the right n-bit ciphertext z. A pictorial illustration of the construction $\widetilde{\mathsf{Fr}}$ is given in Fig. 4.

The optimal (n-bit) security of this $\widetilde{\mathsf{Fr}}$ construction is similar to Theorem 3 for the $\widetilde{\mathsf{F2}}$ construction with a $2n$-bit tweak. Therefore, we omit the proof.

6 Conclusion

In this work, we study the problem of building tweakable forkciphers from an n-bit block cipher. We begin by proposing a design, $\widetilde{\mathsf{F1}}$, for an n-bit tweak and proving its n-bit security. Next, we propose another design, $\widetilde{\mathsf{F2}}$, for a $2n$-bit tweak and prove its n-bit security. Finally, we propose a $\widetilde{\mathsf{Fr}}$ design for an rn-bit tweak, achieving n-bit security. To the best of our knowledge, this is the first design proposal for building tweakable forkciphers from block ciphers. We have proved the security of all these constructions by assuming the underlying block cipher is an ideal cipher.

An interesting direction for future work is to consider designing efficient forkciphers from block ciphers in the standard model. Another promising approach is to design forkciphers based on other primitives, using block ciphers in hash-based designs such as LRW2 [21, 22].

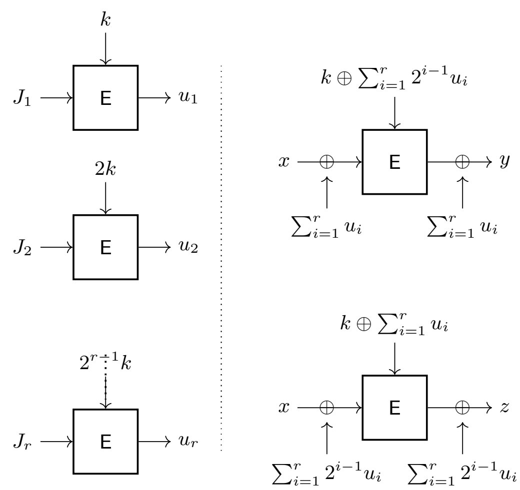

Fig. 4: $\widetilde{\mathsf{Fr}}$: rn-bit tweak TFC from (r+2) BC

References

1. Elena Andreeva, Amit Singh Bhati, Bart Preneel, and Damian Vizár. 1, 2, 3, fork: Counter mode variants based on a generalized forkcipher. *IACR Trans. Symmetric Cryptol.*, 2021(3):1–35, 2021.
2. Elena Andreeva, Amit Singh Bhati, and Damian Vizár. Nonce-misuse security of the saef authenticated encryption mode. In Orr Dunkelman, Michael J. Jacobson, Jr., and Colin O'Flynn, editors, *Selected Areas in Cryptography*, pages 512–534, Cham, 2021. Springer International Publishing.
3. Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. How to securely release unverified plaintext in authenticated encryption. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014*, pages 105–125, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
4. Elena Andreeva, Virginie Lallemand, Antoon Purnal, Reza Reyhanitabar, Arnab Roy, and Damian Vizár. Forkcipher: A New Primitive for Authenticated Encryption of Very Short Messages. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT II*, volume 11922 of *LNCS*, pages 153–182. Springer, 2019.
5. Elena Andreeva, Reza Reyhanitabar, Kerem Varici, and Damian Vizár. Forking a blockcipher for authenticated encryption of very short messages. Cryptology ePrint Archive, Paper 2018/916, 2018. `https://eprint.iacr.org/2018/916`.
6. Elena Andreeva and Andreas Weninger. A forkcipher-based pseudo-random number generator. In Mehdi Tibouchi and XiaoFeng Wang, editors, *Applied Cryptography and Network Security*, pages 3–31, Cham, 2023. Springer Nature Switzerland.
7. Subhadeep Banik, Jannis Bossert, Amit Jana, Eik List, Stefan Lucks, Willi Meier, Mostafizar Rahman, Dhiman Saha, and Yu Sasaki. Cryptanalysis of forkaes. In

Applied Cryptography and Network Security: 17th International Conference, ACNS 2019, Bogota, Colombia, June 5–7, 2019, Proceedings, page 43–63, Berlin, Heidelberg, 2019. Springer-Verlag.
8. Augustin Bariant, Nicolas David, and Gaëtan Leurent. Cryptanalysis of Forkciphers. *IACR Transactions on Symmetric Cryptology*, 2020(1):233–265, May 2020.
9. Francesco Berti, François-Xavier Standaert, and Itamar Levi. Authenticity in the presence of leakage using a forkcipher. Cryptology ePrint Archive, Paper 2024/1325, 2024.
10. Amit Singh Bhati, Elena Andreeva, and Damian Vizár. OAE-RUP: A strong online AEAD security notion and its application to SAEF. In Clemente Galdi and Duong Hieu Phan, editors, *Security and Cryptography for Networks - 14th International Conference, SCN 2024, Amalfi, Italy, September 11-13, 2024, Proceedings, Part II*, volume 14974 of *Lecture Notes in Computer Science*, pages 117–139. Springer, 2024.
11. Amit Singh Bhati, Erik Pohle, Aysajan Abidin, Elena Andreeva, and Bart Preneel. Let's go eevee! A friendly and suitable family of AEAD modes for iot-to-cloud secure computation. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, pages 2546–2560. ACM, 2023.
12. 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.
13. Nilanjan Datta, Avijit Dutta, Eik List, and Sougata Mandal. FEDT: Forkcipher-based leakage-resilient beyond-birthday-secure AE. *IACR Communications in Cryptology*, 1(2), 2024.
14. Nilanjan Datta, Avijit Dutta, and Cuauhtemoc Mancillas-López. LightMAC: Fork it and make it faster, 2024.
15. Avijit Dutta, Jian Guo, and Eik List. Forking sums of permutations for optimally secure and highly efficient PRFs. Cryptology ePrint Archive, Paper 2022/1609, 2022. `https://eprint.iacr.org/2022/1609`.
16. Ewan Fleischmann, Christian Forler, and Stefan Lucks. Mcoe: A family of almost foolproof on-line authenticated encryption schemes. In Anne Canteaut, editor, *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, volume 7549 of *Lecture Notes in Computer Science*, pages 196–215. Springer, 2012.
17. Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and keys for block ciphers: The TWEAKEY framework. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2014.
18. Ashwin Jha, Mustafa Khairallah, Mridul Nandi, and Abishanka Saha. Tight security of TNT and beyond - attacks, proofs and possibilities for the cascaded LRW paradigm. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part I*, volume 14651 of *Lecture Notes in Computer Science*, pages 249–279. Springer, 2024.
19. Hwigyeom Kim, Yeongmin Lee, and Jooyoung Lee. Forking tweakable even-mansour ciphers. *IACR Trans. Symmetric Cryptol.*, 2020(4):71–87, 2020.

20. 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.
21. 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.
22. 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.
23. Sougata Mandal. Tweakable forkcipher from ideal block cipher. *IACR Communications in Cryptology*, 1(3), 2024.
24. 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.
25. Bart Mennink. Optimally secure tweakable blockciphers. Cryptology ePrint Archive, Paper 2015/363, 2015.
26. Rafail Ostrovsky. Efficient computation on oblivious rams. In Harriet Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 514–523. ACM, 1990.
27. 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.
28. 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.
29. 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.
30. 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.