Populating the Zoo of Rugged Pseudorandom Permutations[⋆]

Jean Paul Degabriele[1] and Vukašin Karadžić[2]

[1] Technology Innovation Institute, UAE
jeanpaul.degabriele@tii.ae
[2] Technische Universität Darmstadt, Germany
vukasin.karadzic@tu-darmstadt.de

Abstract. A Rugged Pseudorandom Permutation (RPRP) is a variable-input-length tweakable cipher satisfying a security notion that is intermediate between tweakable PRP and tweakable SPRP. It was introduced at CRYPTO 2022 by Degabriele and Karadžić, who additionally showed how to generically convert such a primitive into nonce-based and nonce-hiding AEAD schemes satisfying either misuse-resistance or release-of-unverified-plaintext security as well as Nonce-Set AEAD which has applications in protocols like QUIC and DTLS. Their work shows that RPRPs are powerful and versatile cryptographic primitives. However, the RPRP security notion itself can seem rather contrived, and the motivation behind it is not immediately clear. Moreover, they only provided a single RPRP construction, called UIV, which puts into question the generality of their modular approach and whether other instantiations are even possible. In this work, we address this question positively by presenting new RPRP constructions, thereby validating their modular approach and providing further justification in support of the RPRP security definition. Furthermore, we present a more refined view of their results by showing that strictly weaker RPRP variants, which we introduce, suffice for many of their transformations. From a theoretical perspective, our results show that the well-known three-round Feistel structure achieves stronger security as a permutation than a mere pseudorandom permutation—as was established in the seminal result by Luby and Rackoff. We conclude on a more practical note by showing how to extend the left domain of one RPRP construction for applications that require larger values in order to meet the desired level of security.

Keywords: Tweakable Wide-Block Ciphers · Rugged Pseudorandom Permutations · Hash-Encipher-Counter · Three-Round Feistel · Domain Extension

Table of Contents

1 Introduction

A Rugged Pseudorandom Permutation (RPRP) is a tweakable variable-input-length cipher satisfying a security notion intermediate between a tweakable Pseudorandom Permutation (PRP) and a tweakable Strong Pseudorandom Permutation (SPRP). It was introduced in [13] where it was shown how to generically convert such a primitive into nonce-based and nonce-hiding AEAD schemes that are either misuse-resistant [22] or secure under the release of unverified plaintext [3]. That work revisited the classical encode-then-encipher paradigm [5,23] and showed analogous constructions that can be instantiated with a weaker primitive—a Rugged PRP instead of a tweakable Strong PRP. Although the encode-then-encipher paradigm is more than twenty years old, it is often dismissed because variable-length tweakable SPRPs are rather inefficient to construct. However, Rugged PRPs can be constructed more efficiently, and their introduction extends the encode-then-encipher paradigm with a new set of tradeoffs between security and efficiency. In addition, Degabriele and Karadžić introduced Nonce-Set AEAD as a conceptual building block from which a variety of order-resilient secure channels, such as QUIC and DTLS, can be easily realised. Indeed they presented a generic way of transforming any Nonce-Set AEAD scheme into an order-resilient channel with any desired functionality, and, in addition, it is simpler than QUIC. Thus, another application of Rugged PRPs is that they can easily be transformed into Nonce-Set AEAD schemes with the added benefit of yielding more compact ciphertexts than alternative constructions. The Authenticate-with-Nonce (AwN) construction, presented in [13], does exactly this. It outperforms other constructions by ‘overloading’ the use of the nonce to additionally provide authentication without introducing further redundancy in the ciphertext. Another important application of Rugged PRPs is that they suffice to construct onion encryption schemes that can be used in Tor [12].

Taking a closer look at Rugged Pseudorandom Permutations, one of their salient features is the asymmetry in the security required from the encipher and decipher algorithms. Roughly speaking, the security definition requires the encipher algorithm to be pseudorandom, but it only imposes a strictly weaker requirement on the decipher algorithm. In the security game, the adversary is given three oracles: an Encipher oracle, a Decipher oracle, and a Guess oracle. The Encipher oracle is equivalent to that in the tweakable (S)PRP games. The Decipher oracle works analogously, but the adversary is significantly restricted in what it can query to this oracle. Finally, the Guess oracle provides an alternative way of interacting with the decipher algorithm. Namely, the adversary can attempt to guess part of the output of the decipher algorithm for an input of its choice, and the oracle returns a single bit indicating success or failure. In contrast to the Decipher oracle, there are no restrictions on the adversary besides that it does not query an input for which it already knows the corresponding output of the decipher algorithm, which is necessary as it would allow for trivial win conditions. This way, the two oracles offer different tradeoffs in how the adversary can interact with the decipher algorithm. Nevertheless, the combination of these two oracles still exposes the decipher algorithm significantly less than the tweakable SPRP game—which is why the RPRP notion is strictly weaker.

As can be noted from the above, the RPRP definition is more involved than the better-known tweakable PRP and SPRP definitions, and the intuition behind it is not immediately clear. Degabriele and Karadžić state that the RPRP definition is tailored to capture the features needed by the encode-then-encipher paradigm and other transforms while at the same time being within reach of more efficient constructions. However, they only present *a single* RPRP construction, called UIV [13], which raises the question of whether this is a contrived security definition that revolves around this single construction. That is, is the abstract notion of a Rugged PRP really justified and is it natural enough for it to be instantiable by other constructions? Their work exposes several applications of RPRPs where they present several transformations for realising higher-level primitives generically from any RPRP. However, the value of their modular approach is rather limited if there exists no other instantiation thereof. In that case, we could just as well focus our attention on this single construction and ignore the security definition. Another limitation of the UIV construction, and [13] more generally, is its rigid security parameterisation. The quantitative security of the UIV construction is closely tied to the block size of the underlying tweakable blockcipher. In the AwN construction, which is used to construct order-resilient channels like QUIC and DTLS, this block size corresponds to a security budget that has to be divided between the overall bit-level security and the amount of reordering that the channel can tolerate. Accordingly, the AES-based instantiation of UIV suggested in [13], while offering good performance on hardware with AES-NI support, may be incapable of delivering the required tradeoff between (multi-user) security and tolerance to reordering that is required in practice by protocols like QUIC and DTLS.

NIST has recently renewed its interest in blockcipher modes of operation with the potential goal of standardising constructions of tweakable variable-length ciphers [19]. In this work, we take a deeper look into Rugged PRPs by revisiting their security definition and presenting new constructions that address the above limitations. Our results complement the work of Degabriele and Karadžić by making a stronger case for the general applicability of Rugged PRPs and their potential role in the upcoming NIST standardisation effort. More specifically, we make contributions in the following directions:

Security Definitions. The asymmetry between the encipher and decipher algorithms gives rise to a broader set of possibilities when applying the encode-then-encipher paradigm. Namely, one could naturally use the encipher algorithm to encrypt and decipher to decrypt, or alternatively, use the decipher algorithm to encrypt and encipher to decrypt. These correspond to the EtE and EtD transforms presented in [13], which have two variants each—yielding either nonce-based AEAD or nonce-hiding AEAD. Compared to the classical encode-then-encipher paradigm (relying on an SPRP), the restrictions on the decipher algorithm render the analysis of these transforms more challenging. A notable feature of these transforms is that their security proofs do not require all three oracles at once. More specifically, the EtE security proofs do not make any use of the Decipher oracle, whereas the EtD ones do not make any use of the Guess oracle. This prompts us to consider two natural relaxations of the RPRP notion, which were not considered in [13], but which still suffice to enable these transforms. By dropping access to the Guess oracle, we obtain the RPRPd notion, and similarly, removing access to the Decipher oracle yields the RPRPg notion. We study the relation between the three notions and present separations showing that these two relaxations result in strictly weaker notions. We will show that introducing these relaxed notions allows us to instantiate the EtE and EtD transforms with a wider class of constructions. That said, there are other applications—such as onion encryption [12]—which still require a full-fledged RPRP, and thus we do not consider our notions to be a replacement but rather a more refined characterisation.

New Constructions. We present three new variable-length tweakable cipher constructions that meet on the three Rugged PRP notions. The first construction, and the one that achieves the strongest of the three notions, namely the RPRP security, is the **H**ash–**E**ncipher–**C**ounter (HEC) construction. It is based on the HCTR construction [24], which achieves tweakable SPRP security and can be seen as a lightweight version of it. It improves over UIV by making do with just a blockcipher rather than a tweakable blockcipher and requiring only a single blockcipher key rather than two, thereby reducing the key-scheduling time. This latter aspect is beneficial, for instance, when it (or the corresponding RPRP-based AEAD scheme) is used in a ratcheted configuration where its key is updated after every message that is encrypted. The other two constructions are based on the classical Feistel construction. More specifically, they consist of three rounds of an unbalanced Feistel structure, which we refer to as **E**xpand-**C**ompress-**E**xpand (ECE) and **C**ompress-**E**xpand-**C**ompress (CEC), where the naming refers to the order in which the underlying pseudorandom functions appear in the construction. Here, we supersede the classical result of Luby and Rackoff by showing that each of these three-round Feistel constructions achieves one of the two restricted RPRP variants (each of which is strictly stronger than tweakable PRP) but not the other. We note that the Feistel constructions are not of mere theoretical interest as they can be instantiated quite efficiently, even if they require three rounds. In particular, recent work has shown efficient instantiations using permutation-based cryptography with very competitive performance [4].

Left-Domain Extension. The security definition of Rugged PRPs requires the tweakable cipher to be defined over a split domain. In [13], the authors assume a split domain of the form $\{0,1\}^n \times \{0,1\}^{\geq m}$ and refer to the two strings that compose an element in this domain as the left and right components. Indeed, their UIV construction, as well as the constructions we introduce, satisfy this syntax. In their transforms, the security of the resulting scheme is always dependent on the size of the left part of the domain n. In the UIV construction as well as our HEC construction the value of n is fixed by the block size of the underlying (tweakable) blockcipher, which is typically 128 bits. As mentioned earlier, in the Nonce-Set AEAD construction presented in [13], the value n has to be divided between the overall bit-level security and the amount of reordering that the channel can tolerate. In a setting like QUIC and DTLS, where an adversary may have multiple forgery attempts and a high degree of reordering should be tolerated, the resulting quantitative security for $n = 128$ may not be satisfactory, especially when considering multi-user security.

One advantage of the Feistel constructions, especially when instantiated with permutation-based primitives, is that they allow for a high degree of freedom in tuning the value of n. In the case of UIV and HEC, adjusting n is not as straightforward, however. Domain extension for blockciphers and tweakable blockciphers has been studied in several prior works. In HEC, there is a single blockcipher instance used throughout various parts of the construction, and replacing all instances would be rather detrimental to performance. On the other hand, in UIV, the tweakable blockcipher whose blocksize determines n is keyed with a separate key, allowing us to replace it with other constructions. We identify two suitable constructions and show how they can be used to extend the left domain of UIV and improve its security when used to construct Nonce-Set AEAD and order-resilient channels like QUIC and DTLS.

1.1 Related Work

The HCTR construction, which our RPRP scheme HEC is based on, was introduced in 2005 by Wang, Feng and Wu. A THCTR [14] is a "tweakable HCTR" construction that appeared in 2019. The authors claimed it achieves beyond-birthday-bound security. However, that was disproven in [2]. The HCTR2 construction [9] is another recent direct "descendant" of HCTR. The HCTR2 mitigates two minor bugs in HCTR specification by changing the hash function and introducing one more masking value in the construction. In addition, HCTR2 construction has a smaller key size than HCTR and a tighter bound. Minematsu and Iwata proposed a beyond-birthday-bound scheme called LargeBlock1 that is similar to HCTR [18]. A more interesting point about this construction is the extended size of the left input, which makes it related to the domain extender idea we deal with in Section 6. However, the LargeBlock1 construction in question is neither a tweakable cipher, nor is it VIL.

As mentioned before, the UIV construction from [13] is the only other construction proved so far to be a RPRP. It has the same number of keys as our HEC construction, though it needs one more key-scheduling setup step. The constructions are similar in the sense that both have a 2-round pass, but they differ in the underlying building blocks (e.g., UIV uses a tweakable blockcipher, HEC a blockcipher).

Using Feistel schemes to build PRPs or SPRPs is an idea that dates back to the seminal work of Luby and Rackoff [16]. Since then, there has been much work on this conceptual idea. We are interested in more recent work, namely that of [1] and [4]. The unbalanced Feistel schemes we present in this work closely resemble the schemes based on the three-round unbalanced Feistel that appear in those works. First of our unbalanced three-round Feistel schemes, the ECE scheme, looks similar to Deck-JAMBO [4]. The other, CEC constructions, is similar to Deck-BOREE [4] and could be seen as an abstraction of the RIV scheme [1]. However, there is one crucial distinction between our work and theirs. The target cryptographic primitive and security notion they target is AE(AD). We treat the aforementioned schemes ECE and CEC in the setting of VIL tweakable ciphers.

2 Preliminaries

Notation. For any string X we denote its length in bits by $|X|$ and ε denotes the empty string. For any integer $0 < a \leq |X|$, $\lfloor X \rfloor_a$ denotes the substring consisting of the first a bits of X, and $\lceil X \rceil_a$ denotes the substring consisting of the last a bits of X. For any two integers a and b, $\langle a \rangle_2$ denotes a's representation as a binary string, and if $0 < b \leq a$ we denote the falling factorial $a(a-1)\cdots(a-b+1)$ by $(a)_b$. For a real number $r > 0$, $\lceil r \rceil$ denotes the first integer that is greater than or equal to r.

For any set $\mathcal{S}$, $s \leftarrow\!\!\$\ \mathcal{S}$ denotes the process of uniformly sampling an element from the set $\mathcal{S}$ and assigning it to s. We use $\mathsf{IC}(\mathcal{K}, \mathcal{X})$ to denote the set of all ciphers over the domain $\mathcal{X}$ and key space $\mathcal{K}$. Similarly 2-Func$(\mathcal{T}, \mathcal{X})$ denotes the set of all functions $\{+,-\} \times \mathcal{T} \times \mathcal{X} \to \mathcal{X}$. Sampling uniformly at random from 2-Func$(\mathcal{T}, \mathcal{X})$ yields what is sometimes referred to as a two-sided random function, that can alternatively be viewed as a pair of independent random functions $\mathcal{T} \times \mathcal{X} \to \mathcal{X}$.

For an event E and process P, we denote with $\Pr[P : E]$ the probability of event E occuring after running process P.

Tweakable Ciphers. A tweakable cipher is an algorithm

$$\widetilde{\mathsf{EE}} : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \to \mathcal{X}$$

that, for $(K, T) \in \mathcal{K} \times T$, identifies a permutation $\widetilde{\mathsf{EE}}(K, T, \cdot)$ over the domain $\mathcal{X}$. We refer to $\mathcal{K}$ and $\mathcal{T}$ as key space and tweak space, respectively. We write the inverse of $\widetilde{\mathsf{EE}}$ as $\widetilde{\mathsf{EE}}^{-1}(K, T, \cdot)$. We define $\widetilde{\mathsf{EE}}_K(T, \cdot) := \widetilde{\mathsf{EE}}(K, T, \cdot)$ and $\widetilde{\mathsf{EE}}_K^{-1}(T, \cdot) := \widetilde{\mathsf{EE}}^{-1}(K, T, \cdot)$. One of the two classical security definitions for tweakable ciphers is the *strong tweakable pseudorandom permutation* (STPRP) security notion. Intuitively the notion implies that an adversary cannot distinguish between a STPRP-secure cipher keyed with a random key and an ideal cipher with key space $\mathcal{T}$. The definition of STPRP advantage is given below.

Definition 1 (STPRP **Advantage**). *Let $\widetilde{\mathsf{EE}}$ be a tweakable cipher defined over $(\mathcal{K}, \mathcal{T}, \mathcal{X})$. Then for any adversary $\mathcal{A}$ its* STPRP *advantage is defined as:*

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{EE}}}(\mathcal{A}) = \left|\Pr\left[K \leftarrow\!\$\, \mathcal{K} : \mathcal{A}^{\widetilde{\mathsf{EE}}_K(\cdot,\cdot), \widetilde{\mathsf{EE}}^{-1}_K(\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\widetilde{\Pi} \leftarrow\!\$\, \mathsf{IC}(\mathcal{T}, \mathcal{X}) : \mathcal{A}^{\widetilde{\Pi}(\cdot,\cdot), \widetilde{\Pi}^{-1}(\cdot,\cdot)} \Rightarrow 1\right]\right|$$

In the weaker TPRP notion the adversary only has access to the encipher oracle, and the advantage is then defined analogously.

If the tweak set is a singleton, then a tweakable cipher becomes just a *cipher*. Furthermore, if $\mathcal{X} = \{0,1\}^n$, we call the cipher a *blockcipher*. The security notion for (block)ciphers adjust accordingly, and we denote them with PRP and SPRP.

Hash Functions. A hash function is a function

$$\mathsf{H} : \mathcal{H} \times \{0,1\}^* \to \mathcal{Y}$$

taking as an input a hash key $h \in \mathcal{H}$ and a string $X \in \{0,1\}^*$ and outputting an element from output space $\mathcal{Y}$. In this work, we will mainly use hash functions with output space $\{0,1\}^n$.

Security. There are many security notions a hash function can satisfy. We are interested in the *almost-XOR-universal* (AXU) hash functions, the definition of which follows.

Definition 2. *Let* H *be a hash function with key space* $\mathcal{H}$ *and output space* $\mathcal{Y}$*. We call* H ϵ_1*-AXU if for all bit string pairs* (X_1, X_2)*, with* $X_1 \neq X_2$*, and* $Y \in \mathcal{Y}$ *it holds*

$$\Pr_{h \leftarrow\$\mathcal{H}}[\mathsf{H}_h(X_1) \oplus \mathsf{H}_h(X_2) = Y] \leq \epsilon_1.$$

PRFs. Let $\mathsf{FE} : \{0,1\}^k \times \{0,1\}^{\geq n} \times \{0,1\}^l \to \{0,1\}^*$ be a variable-input-length (VIL) variable-output-length (VOL) function with key of size k bits. The first input is $X \in \{0,1\}^{\geq n}$ and the second input $L \in \{0,1\}^l$ is the size of output the function should produce.

We expect the function FE to behave as an independent PRF for every output length L. The PRF security definition of FE uses a VOL random function R^∞. For an input (X, L), function R^∞ outputs a uniformly random string of length L bits. Formally, the security is then defined as follows.

Definition 3. *For an adversary* $\mathcal{A}$*, the PRF advantage of VOL function* $\mathsf{FE} : \{0,1\}^k \times \{0,1\}^{\geq n} \times \{0,1\}^l \to \{0,1\}^*$ *is defined as*

$$\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{FE}}(\mathcal{A}) = \left|\Pr\left[K \leftarrow\!\$\, \mathcal{K} : \mathcal{A}^{\mathsf{FE}_K(\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathsf{R}^\infty(\cdot,\cdot)} \Rightarrow 1\right]\right|.$$

We also make use of VIL functions with fixed output size. Let $\mathsf{FC} : \{0,1\}^k \times \{0,1\}^{\geq m} \to \{0,1\}^n$ VIL function with output size n. The key is k bits long and m the minimum size of the function input.

Definition 4. *For an adversary* $\mathcal{A}$*, the PRF security of VIL function* $\mathsf{FC} : \{0,1\}^k \times \{0,1\}^{\geq m} \to \{0,1\}^n$ *is defined as*

$$\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{FC}}(\mathcal{A}) = \left|\Pr\left[K \leftarrow\!\$\, \mathcal{K} : \mathcal{A}^{\mathsf{FC}_K(\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathsf{R}^\infty(\cdot,n)} \Rightarrow 1\right]\right|.$$

H-Coefficient Technique. In all of the proofs in this paper, we utilize the H-coefficient technique. The H-coefficient technique [6,20] is a tool used for bounding the advantage of a computationally unbounded adversary $\mathcal{A}$, which is trying to distinguish whether it is interacting with the real or the ideal world. The adversary $\mathcal{A}$ can make oracle queries to either the real construction (in the real world) or its ideal equivalent (in the ideal world). The list of $\mathcal{A}$'s queries and corresponding answers is contained in a *transcript* τ. A transcript τ is called *attainable* if the probability that τ is generated during $\mathcal{A}$'s interaction with the ideal world is greater than 0.

A rough tutorial for the application of the H-coefficient technique goes as follows. We define what the transcript looks like. Then, one defines what it means for a transcript to be *bad*. After that, we need to calculate the probability that some transcript is bad. Finally, one should calculate the interpolation probabilities of some *good* attainable transcript appearing in the real world and it appearing in the ideal world. A transcript is called good if it is not bad. By applying the theorem we give below, one obtains a bound on the adversary's distinguishing advantage.

Letting X_r and X_i denote random variables corresponding to the transcript generated during $\mathcal{A}$'s interaction with the real and ideal world, the H-Coefficient technique is applied using the following theorem.

Theorem 1. *Let $\mathcal{A}$ be a computationally unbounded adversary trying to distinguish between a real world, represented by the game $\mathbf{G}_{\text{real}}$, and an ideal world, represented by the game $\mathbf{G}_{\text{ideal}}$. Let $\mathcal{T}$ be the set of all attainable transcripts and let $\mathcal{T}_{bad}$ be a set of transcripts deemed to be bad. Define $\mathcal{T}_{good} := \mathcal{T} \setminus \mathcal{T}_{bad}$. If there exist ϵ_{bad}, $\epsilon_{\text{ratio}} \geq 0$ such that for all transcripts $\tau' \in \mathcal{T}_{good}$*

$$\frac{\Pr[X_r = \tau']}{\Pr[X_i = \tau']} \geq 1 - \epsilon_{\text{ratio}} \quad \text{and} \quad \Pr[X_i \in \mathcal{T}_{bad}] \leq \epsilon_{\text{bad}},$$

then it holds

$$\left|\Pr\left[\mathcal{A}^{\mathbf{G}_{\text{real}}} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathbf{G}_{\text{ideal}}} \Rightarrow 1\right]\right| \leq \epsilon_{\text{bad}} + \epsilon_{\text{ratio}}.$$

3 RPRPs, its Derivatives and Relations Among Them

The RPRP security notion for VIL tweakable ciphers over a split domain was introduced by Degabriele and Karadžić [13]. The RPRP security game they present offers the adversary access to the decipher algorithm via two oracles. One is a *"restricted" decipher* oracle De, and the other is an oracle Gu they call *guess* oracle. The game in question is given in Figure 1 together with games RPRPd and RPRPg, which are our contributions. We present two subvariants of the RPRP game, namely these RPRPd and RPRPg games. In the RPRPd game, the adversary has access to En and De oracles, while in the RPRPg game, the adversary has access to En and Gu oracles. The restrictions imposed by the RPRP game are also present in the subvariant games. We aim to investigate the relations between the RPRP security notion and the security notions corresponding to the subvariants. For completeness, we reiterate the definition of RPRP advantage in the following and present analogous advantage definitions for RPRPd and RPRPg notions.

Definition 5 (RPRP / RPRPg **Advantage).** *Let $\widetilde{\mathsf{EE}}$ be a tweakable cipher over a split domain $(\mathcal{X}_L \times \mathcal{X}_R)$. Then for a positive integer v and an adversary $\mathcal{A}$ attacking the* RPRP / RPRPg *security of $\widetilde{\mathsf{EE}}$ the corresponding advantage is defined as*

$$\mathbf{Adv}^{\text{RPRP/RPRPg}}_{\widetilde{\mathsf{EE}}}(\mathcal{A}, v) = \left|2\Pr\left[\mathsf{RPRP}^{\mathcal{A},v}_{\widetilde{\mathsf{EE}}} \,/\, \mathsf{RPRPg}^{\mathcal{A},v}_{\widetilde{\mathsf{EE}}} \Rightarrow 1\right] - 1\right|.$$

Definition 6 (RPRPd **Advantage).** *Let $\widetilde{\mathsf{EE}}$ be a tweakable cipher over a split domain $(\mathcal{X}_L \times \mathcal{X}_R)$. Then for an adversary $\mathcal{A}$ attacking the* RPRPd *security of $\widetilde{\mathsf{EE}}$ the corresponding advantage is defined as*

$$\mathbf{Adv}^{\text{RPRPd}}_{\widetilde{\mathsf{EE}}}(\mathcal{A}) = \left|2\Pr\left[\mathsf{RPRPd}^{\mathcal{A}}_{\widetilde{\mathsf{EE}}} \Rightarrow 1\right] - 1\right|.$$

3.1 Relations between RPRP notions

Now that we have defined the RPRP subvariants, we can continue showing the relations between RPRP, RPRPd, and RPRPg notions. It is obvious that RPRP security implies both RPRPd and RPRPg notions since in the games of the latter notions, the adversary has one oracle access less than in the RPRP game. Therefore, if it cannot distinguish while having access to all three oracles, it cannot distinguish having access to just two.

The interesting relations are those between RPRPg and RPRPd notions. As we will show next, neither implies the other notion. We show the RPRPg $\not\Rightarrow$ RPRPd separation in a general way. In contrast, for the other way around, we show the separation with the help of a concrete construction. In Figure 2 we give an overview of the established relations.

3.1.1 RPRPg $\not\Rightarrow$ RPRPd. Let $\widetilde{\mathsf{EE}}$ be a RPRPg-secure tweakable cipher and assume $k = n$. We construct a tweakable cipher $\widetilde{\mathsf{EE}}'$ that is not RPRPd secure. The cipher $\widetilde{\mathsf{EE}}'$ has the same key and tweak space, domain and range, and is defined as follows.

$$\widetilde{\mathsf{EE}}'_K(T, X_L, X_R) = \begin{cases} (0^n, 0^n), & \text{if } (T, X_L, X_R) = (0^n, K, 0^n) \\ \widetilde{\mathsf{EE}}_K(0^n, K, 0^n), & \text{if } (X_L, X_R) = \widetilde{\mathsf{EE}}_K^{-1}(0^n, 0^n, 0^n) \wedge T = 0^n \\ \widetilde{\mathsf{EE}}_K(T, X_L, X_R), & \text{otherwise.} \end{cases}$$

Now in the RPRPd game, an adversary can correctly guess the bit b by first querying De$(0^n, 0^n, 0^n)$ and taking the left output X_L as a key guess. It then checks if it is interacting with the real world by making some enciphering queries and checking if the answers are equal to the outputs it could calculate itself with the key guess.

The attack can easily be adapted to the cases where $k < n$ or $k > n$.

```
Game $\mathsf{RPRP}^{\mathcal{A},v}_{\widetilde{\mathsf{EE}}}$ / $\boxed{\mathsf{RPRPd}^{\mathcal{A}}_{\widetilde{\mathsf{EE}}}}$ / $\mathsf{RPRPg}^{\mathcal{A},v}_{\widetilde{\mathsf{EE}}}$
  $K \leftarrow\!\!\$\ \mathcal{K}$
  $b \leftarrow\!\!\$\ \{0,1\}$
  $\mathcal{F}, \mathcal{R}, \mathcal{U} \leftarrow \emptyset, \emptyset, \emptyset$
  $\widetilde{\Pi} \leftarrow\!\!\$\ \mathsf{IC}(\mathcal{T}, \mathcal{X}_L \times \mathcal{X}_R)$
  $b' \leftarrow \mathcal{A}^{\text{En,De,Gu}}$ / $\boxed{\mathcal{A}^{\text{En,De}}}$ / $\mathcal{A}^{\text{En,Gu}}$
  return $b = b'$

De$(T, Y_L, Y_R)$
  if $Y_L \in \mathcal{F} \cup \mathcal{R}$
    return $\lightning$
  if $b = 0$
    $(X_L, X_R) \leftarrow \widetilde{\Pi}^{-1}(T, Y_L, Y_R)$
  else
    $(X_L, X_R) \leftarrow \widetilde{\mathsf{EE}}_K^{-1}(T, Y_L, Y_R)$
  $\mathcal{R} \xleftarrow{\cup} \{Y_L\}$; $\mathcal{U} \xleftarrow{\cup} \{(T, Y_L, Y_R)\}$
  return $(X_L, X_R)$

En$(T, X_L, X_R)$
  if $b = 0$
    $(Y_L, Y_R) \leftarrow \widetilde{\Pi}(T, X_L, X_R)$
  else
    $(Y_L, Y_R) \leftarrow \widetilde{\mathsf{EE}}_K(T, X_L, X_R)$
  $\mathcal{F} \xleftarrow{\cup} \{Y_L\}$ ; $\mathcal{U} \xleftarrow{\cup} \{(T, Y_L, Y_R)\}$
  return $(Y_L, Y_R)$

Gu$(T, Y_L, Y_R, \boldsymbol{V})$
  if $((T, Y_L, Y_R) \in \mathcal{U}) \vee (|\boldsymbol{V}| > v)$
    return $\lightning$
  if $b = 0$
    return false
  else
    $(X_L, X_R) \leftarrow \widetilde{\mathsf{EE}}_K^{-1}(T, Y_L, Y_R)$
    return $X_L \in \boldsymbol{V}$
```

Fig. 1: The games used to define RPRP, RPRPd and RPRPg security for a tweakable cipher $\widetilde{\mathsf{EE}}$.

$$\begin{tikzcd}
& \text{RPRP} \arrow[dl] \arrow[dr] & \\
\text{RPRPd} \arrow[rr, dotted, bend left, "/" description] & & \text{RPRPg} \arrow[ll, dotted, bend left, "/" description]
\end{tikzcd}$$

Fig. 2: Relations between RPRP notions. Solid arrows indicate trivial implications. Dotted, stroke-out arrows indicate separations.

$\widetilde{\mathsf{EE}}'$ *is* RPRPg *secure.* We argue informally why this reduction holds. Our "rewired" $\widetilde{\mathsf{EE}}'$ differs from $\widetilde{\mathsf{EE}}$ only for two values. Problematic queries are the ones where the cipher $\widetilde{\mathsf{EE}}'$ would be queried on these differing values. If the adversary does not make problematic queries, the reduction is obvious. If the adversary makes a problematic query, it could break the security of $\widetilde{\mathsf{EE}}'$. However, the probability of the adversary making a problematic query is small.

The probability that the adversary queries the encipher oracle with $(0^n, K, 0^n)$ is equal to the probability that it guesses a secret random key. The probability that the adversary queries the encipher oracle with $(0^n, X_L, X_R)$, where $(X_L, X_R) = \widetilde{\mathsf{EE}}_K^{-1}(0^n, 0^n, 0^n)$ is also small, since $\widetilde{\mathsf{EE}}_K$ is by assumption indistinguishable from an ideal cipher.

As for the guess oracle, the problematic queries would be $\text{Gu}(0^n, 0^n, 0^n, \{K\})$, and $\text{Gu}(0^n, Y_L, Y_R, \{X_L\})$, where $(Y_L, Y_R) = \widetilde{\mathsf{EE}}_K(0^n, K, 0^n)$ and $(X_L, X_R) = \widetilde{\mathsf{EE}}_K^{-1}(0^n, 0^n, 0^n)$. Since it is by assumption hard to guess the left deciphering output in $\widetilde{\mathsf{EE}}$, the probability of the adversary making successful guess queries will be small.

Hence, the RPRPg security of $\widetilde{\mathsf{EE}}'$ reduces to the RPRPg security of $\widetilde{\mathsf{EE}}$, except for the small probability of these problematic queries occurring. □

3.1.2 RPRPd $\not\Rightarrow$ RPRPg. In proving the separation in the other direction we do not have the generality we had in the previous case. Here we give a concrete construction and show it is RPRPd secure, but not RPRPg secure. The construction in question is an unbalanced three-round Feistel construction. We present it, together with the separation result, in the following Section 4.

```
$\widetilde{\mathsf{EE}}_K(T, X_L, X_R)$
    $(K_1, K_2, K_3) \leftarrow K$
    $I \leftarrow X_R \oplus \mathsf{FE}_{K_1}(X_L, |X_R|)$
    $Y_L \leftarrow X_L \oplus \mathsf{FC}_{K_2}(T, I)$
    $Y_R \leftarrow I \oplus \mathsf{FE}_{K_3}(Y_L, |X_R|)$
    return $(Y_L, Y_R)$
```

```
$\widetilde{\mathsf{EE}}_K^{-1}(T, Y_L, Y_R)$
    $(K_1, K_2, K_3) \leftarrow K$
    $I \leftarrow Y_R \oplus \mathsf{FE}_{K_3}(Y_L, |Y_R|)$
    $X_L \leftarrow Y_L \oplus \mathsf{FC}_{K_2}(T, I)$
    $X_R \leftarrow I \oplus \mathsf{FE}_{K_1}(X_L, |Y_R|)$
    return $(X_L, X_R)$
```

Fig. 3: Pseudocode description of 3-round Feistel construction ECE.

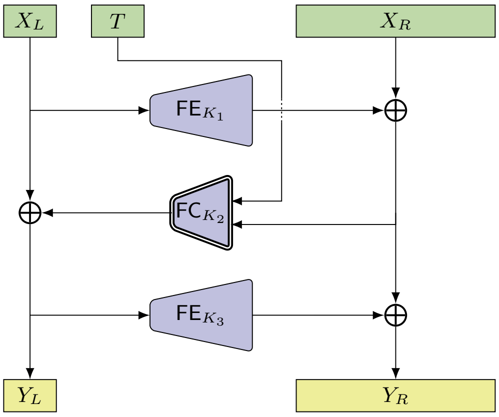

Fig. 4: Graphical representation of 3-round Feistel construction ECE, realized from expanding PRF FE and compressing PRF FC.

4 3-Round Feistel Construction

For an unbalanced 3-round Feistel construction, it is natural to consider two variants. The first one is **E**xpand-**C**ompress-**E**xpand (ECE) variant, where in the first and third round, the left part is expanded and added to the right part, and in the second round, the right part is compressed and added to the left part.

The expanding and compressing are realised using a VOL PRF FE : $\{0,1\}^k \times \{0,1\}^n \times \{0,1\}^l \rightarrow \{0,1\}^{\geq m}$ and VIL PRF FC : $\{0,1\}^k \times \{0,1\}^{\geq m} \rightarrow \{0,1\}^n$, respectively. We sometimes call FE an *expanding* PRF, and FC a *compressing* PRF. The graphical representation of the ECE encipher algorithm is given in Figure 4 and pseudocode description of its encipher and decipher algorithms in Figure 3. The second variant of an unbalanced 3-round Feistel we consider is **C**ompress-**E**xpand-**C**ompress (CEC) construction, where the first and third rounds are compressing, and the second one is expanding. The graphical representation of the CEC encipher algorithm is given in Figure 6 and pseudocode description of its encipher and decipher algorithms in Figure 5 . The expanding PRF admits, in this case, three inputs, where the first two are the values the PRF should be evaluated on, and the third one is the output length[3].

One may wonder why only the second rounds in the constructions admit the tweak T. The reason is that in both ECE and CEC constructions, tweaking just the second round is enough to make them RPRPd and RPRPg secure, respectively. Going further, we show, as a negative result, that a three-round Feistel cipher is not a RPRP. Specifically, in the following, we present an attack against RPRPg security of the ECE variant. The same attack works in the RPRP game, where one does not use the deciphering oracle. This attack makes the first step of showing the RPRPd $\not\Rightarrow$ RPRPg separation.

ECE is not RPRPg Secure. To break the RPRPg security of ECE, the adversary $\mathcal{A}$ executes the following steps.

1. Query $(Y_L^1, Y_R^1) \leftarrow \textsc{En}(T, X_L, X_R)$, with $X_L \neq X_R$.
2. Query $(Y_L^2, Y_R^2) \leftarrow \textsc{En}(T, X_L, Y_R^1)$

[3] One can equivalently write $\mathsf{FE}_{K_2}(T, I, |X_R|)$ as $\mathsf{FE}_{K_2}(T\|I, |X_R|)$

```
$\widetilde{\mathsf{EE}}_K(T, X_L, X_R)$
    $(K_1, K_2, K_3) \leftarrow K$
    $I \leftarrow X_R \oplus \mathsf{FC}_{K_1}(X_R)$
    $Y_R \leftarrow X_R \oplus \mathsf{FE}_{K_2}(T, I, |X_R|)$
    $Y_L \leftarrow I \oplus \mathsf{FC}_{K_3}(Y_L)$
    return $(Y_L, Y_R)$
```

```
$\widetilde{\mathsf{EE}}_K^{-1}(T, Y_L, Y_R)$
    $(K_1, K_2, K_3) \leftarrow K$
    $I \leftarrow Y_L \oplus \mathsf{FC}_{K_3}(Y_R)$
    $X_R \leftarrow Y_R \oplus \mathsf{FE}_{K_2}(T, I, |Y_R|)$
    $X_L \leftarrow I \oplus \mathsf{FC}_{K_1}(X_R)$
    return $(X_L, X_R)$
```

Fig. 5: Pseudocode description of 3-round Feistel construction CEC.

3. Query $o \leftarrow \textsc{Gu}(T, Y_L^1, X_R, \{Y_L^1 \oplus Y_L^2 \oplus X_L\})$
4. **output** 1 if $o =$ **true**, otherwise **output** 0.

In the following calculation we omit second inputs to the expanding PRFs, $|X_R|$ or $|Y_R|$, for the sake of readability. The result of its first query is

$$Y_L^1 = X_L \oplus \mathsf{FC}_{K_2}(T, X_R \oplus \mathsf{FE}_{K_1}(X_L))$$

and

$$Y_R^1 = X_R \oplus \mathsf{FE}_{K_1}(X_L) \oplus \mathsf{FE}_{K_3}(X_L \oplus \mathsf{FC}_{K_2}(T, X_R \oplus \mathsf{FE}_{K_1}(X_L)))$$

Similarly, the output of its second query is

$$Y_L^2 = X_L \oplus \mathsf{FC}_{K_2}(T, Y_R^1 \oplus \mathsf{FE}_{K_1}(X_L))$$

and

$$Y_R^2 = Y_R^1 \oplus \mathsf{FE}_{K_1}(X_L) \oplus \mathsf{FE}_{K_3}(Y_L^2)$$

The last query $\mathcal{A}$ makes is a guess oracle query, and $\mathcal{A}$ outputs that result as its final guess (real or ideal world). Suppose the adversary has access to the real cipher, and let us look at the guess oracle query. Left part of the deciphered input inside the Gu oracle would be

$$X_L^3 = Y_L^1 \oplus \mathsf{FC}_{K_2}(T, X_R \oplus \mathsf{FE}_{K_3}(Y_L^1)).$$

On the other hand, the guessed value is equal to

$$\underbrace{X_L \oplus \mathsf{FC}_{K_2}(T, X_R \oplus \mathsf{FE}_{K_1}(X_L))}_{Y_L^1} \oplus \underbrace{X_L \oplus \mathsf{FC}_{K_2}(T, Y_R^1 \oplus \mathsf{FE}_{K_1}(X_L))}_{Y_L^2} \oplus X_L$$
$$= Y_L^1 \oplus \mathsf{FC}_{K_2}(T, Y_R^1 \oplus \mathsf{FE}_{K_1}(X_L)) = Y_L^1 \oplus \mathsf{FC}_{K_2}(T, X_R \oplus \mathsf{FE}_{K_3}(Y_L^1)),$$

which is exactly equal to X_L^3. Therefore, the adversary $\mathcal{A}$ always outputs 1 if the bit b in the RPRP game is 1. On the other hand, in the ideal world ($b = 0$), the guess oracle returns **true** with very small probability. Overall, $\mathcal{A}$ wins the RPRPg game with high probability.

ECE is RPRPd Secure. The other part of the separation comes next. In Theorem 2, we give the result for RPRPd security of ECE. The proof utilizes the H-coefficient technique, focusing on finding collisions in the input of inner PRFs. There already exist proofs for 3-round Feistel being a secure PRP. However, our proof required a different analysis since we are trying to prove a stronger notion (and a tweakable one at that). There is no reference proof for the 3-round Feistel that considers decipher queries, and that is what we needed to take care of in our analysis. We give the full, detailed proof in Appendix B.

Theorem 2. *Let* ECE *be the construction defined in Figure 3 over the domain $\{0,1\}^n \times \{0,1\}^{\geq m}$. For an adversary $\mathcal{A}$ making q_{en} encipher and q_{de} decipher queries, there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ such that*

$$\mathbf{Adv}^{\mathrm{RPRPd}}_{\mathsf{ECE}}(\mathcal{A}) \leq 2\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{FE}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{FC}}(\mathcal{C}) + \frac{q^2}{2^{n+m}} + \frac{q^2}{2^{m+1}} + \frac{q_{\mathrm{en}}^2}{2^n} + \frac{q_{\mathrm{en}}q_{\mathrm{de}}}{2^{n-1}},$$

under the assumption that $q_{\mathrm{en}} + q_{\mathrm{de}} \leq \frac{1}{2^{n+m-1}}$, and where $q = q_{\mathrm{en}} + q_{\mathrm{de}}$. The resulting PRF adversary $\mathcal{B}$ makes at most $q_{\mathrm{en}} + q_{\mathrm{gu}}$ queries, whereas the resulting PRF adversary $\mathcal{C}$ makes at most $q_{\mathrm{en}} + q_{\mathrm{gu}}$ queries.

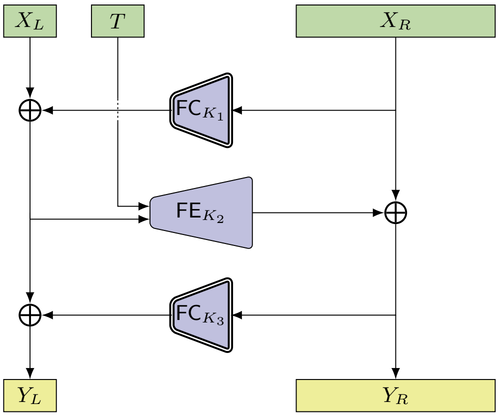

Fig. 6: Graphical representation of 3-round Feistel construction CEC, realized from compressing PRF FC and expanding PRF FE.

As it can be seen from the bound, in order for ECE to have meaningful security, the minimal size of the right input m needs to be large enough (i.e., $m \geq n$).

We can now continue with analyzing the security of the CEC construction. The results for the CEC construction are the opposite of those for ECE. As we will show, CEC is not RPRPd secure but satisfies RPRPg security.

CEC is not RPRPd Secure. To break the RPRPd security of CEC, the adversary $\mathcal{A}$ executes the following steps.

1. Query $(Y_L^1, Y_R^1) \leftarrow \textsc{En}(T, X_L, X_R)$, with $X_L \neq X_R$.
2. Query $(Y_L^2, Y_R^2) \leftarrow \textsc{En}(T, Y_L^1, X_R)$
3. Query $(X_L^3, X_R^3) \leftarrow \textsc{De}(T, X_L, Y_R^1)$
4. **output** 1 if $X_R^3 = X_R \oplus Y_R^1 \oplus Y_R^2$, otherwise **output** 0.

In the following calculation we omit third inputs to the expanding PRF, $|X_R|$ or $|Y_R|$, denoting the needed output length, for the sake of readability. The result of its first query is

$$Y_L^1 = X_L \oplus \mathsf{FC}_{K_1}(X_R) \oplus \mathsf{FC}_{K_3}(X_R \oplus \mathsf{FE}_{K_2}(T, X_L \oplus \mathsf{FC}_{K_1}(X_R)))$$

and

$$Y_R^1 = X_R \oplus \mathsf{FE}_{K_2}(T, X_L \oplus \mathsf{FC}_{K_1}(X_R))$$

Similarly, the output of its second query is

$$Y_L^2 = Y_L^1 \oplus \mathsf{FC}_{K_1}(X_R) \oplus \mathsf{FC}_{K_3}(Y_R^2)$$

and

$$Y_R^2 = X_R \oplus \mathsf{FE}_{K_2}(T, Y_L^1 \oplus \mathsf{FC}_{K_1}(X_R))$$

The last query $\mathcal{A}$ makes leads to the right value X_R^3 of

$$X_R^3 = Y_R^1 \oplus \mathsf{FE}_{K_2}(T, X_L \oplus \mathsf{FC}_{K_3}(Y_R^1)).$$

The question now if X_R^3 is equal to $X_R \oplus Y_R^1 \oplus Y_R^2$.

$$\begin{aligned} & Y_R^1 \oplus \mathsf{FE}_{K_2}(T, X_L \oplus \mathsf{FC}_{K_3}(Y_R^1)) \stackrel{?}{=} X_R \oplus Y_R^1 \oplus Y_R^2 \\ \iff \quad & \mathsf{FE}_{K_2}(T, X_L \oplus \mathsf{FC}_{K_3}(Y_R^1)) \stackrel{?}{=} X_R \oplus Y_R^2 \\ \iff \quad & \mathsf{FE}_{K_2}(T, X_L \oplus \mathsf{FC}_{K_3}(Y_R^1)) \stackrel{?}{=} \mathsf{FE}_{K_2}(T, Y_L^1 \oplus \mathsf{FC}_{K_1}(X_R)) \\ \iff \quad & X_L \oplus \mathsf{FC}_{K_3}(Y_R^1) \stackrel{?}{=} Y_L^1 \oplus \mathsf{FC}_{K_1}(X_R). \end{aligned}$$

The equality above holds in the real world, that is, if $b = 1$. Our adversary also outputs 1 in this case, and thus correctly guesses. On the other hand, in the ideal world ($b = 0$), the probability that the equality above holds is very small. In total, $\mathcal{A}$ wins the RPRPd game with high probability.

```
$\widetilde{\mathsf{EE}}_{K,h}(T, X_L, X_R)$
    $LL \leftarrow X_L \oplus \mathsf{H}_h(T, X_R)$
    $Y'_L \leftarrow \mathsf{E}_K(LL)$
    $IV \leftarrow LL \oplus Y'_L\,;\ k \leftarrow \lceil |X_R|/n \rceil$
    $S \leftarrow \lfloor \mathsf{E}_K(IV \oplus 1) \| \cdots \| \mathsf{E}_K(IV \oplus k) \rfloor_{|X_R|}$
    $Y_R \leftarrow X_R \oplus S$
    $Y_L \leftarrow Y'_L \oplus K_C$
    return $(Y_L, Y_R)$
```

```
$\widetilde{\mathsf{EE}}^{-1}_{K,h}(T, Y_L, Y_R)$
    $Y'_L \leftarrow Y_L \oplus K_C$
    $LL \leftarrow \mathsf{E}_K^{-1}(Y'_L)$
    $IV \leftarrow LL \oplus Y'_L\,;\ k \leftarrow \lceil |Y_R|/n \rceil$
    $S \leftarrow \lfloor \mathsf{E}_K(IV \oplus 1) \| \cdots \| \mathsf{E}_K(IV \oplus k) \rfloor_{|Y_R|}$
    $X_R \leftarrow Y_R \oplus S$
    $X_L \leftarrow LL \oplus \mathsf{H}_h(T, X_R)$
    return $(X_L, X_R)$
```

Fig. 7: Pseudocode description of HEC[H, E].

CEC is RPRPg Secure. We present the result for RPRPg security of CEC in Theorem 3. The proof utilizes the H-coefficient technique, and it was challenging to incorporate the analysis of guess oracle queries. The peculiarities of the guess oracle, namely the fact that the only thing leaked to the adversary is whether $X_L \in \boldsymbol{V}$, contrast the conventional approach in the H-coefficient technique where the whole output of the enciphering or deciphering needs to be included in a query transcript. In the case of a guess query, that would mean the internally deciphered (X_L, X_R) needs to be a part of the query transcript. We "circumvent" this challenge by defining a more complex sampling procedure that builds the transcript in the ideal world of the H-coefficient technique. The full, detailed proof is given in Appendix C.

Theorem 3. *Let* CEC *be the construction defined in Figure 6 over the domain* $\{0,1\}^n \times \{0,1\}^{\geq m}$. *For an adversary* $\mathcal{A}$ *making* q_{en} *encipher and* q_{gu} *guess queries, there exist adversaries* $\mathcal{B}$ *and* $\mathcal{C}$ *such that*

$$\mathbf{Adv}^{\mathrm{RPRPg}}_{\mathsf{CEC}}(\mathcal{A}, v) \leq 2\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{FC}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{FE}}(\mathcal{C}) + \frac{q_{\mathrm{en}}^2}{2^{n+m+1}} + \\ + \frac{q_{\mathrm{en}}^2 + q_{\mathrm{gu}}^2}{2^{n+1}} + \frac{q_{\mathrm{en}} q_{\mathrm{gu}} + q_{\mathrm{gu}} v}{2^n} + \frac{q_{\mathrm{en}}^2 + q_{\mathrm{gu}}^2}{2^{m+1}} + \frac{3 q_{\mathrm{en}} q_{\mathrm{gu}}}{2^m}.$$

The resulting PRF adversary $\mathcal{B}$ *makes at most* $q_{\mathrm{en}} + q_{\mathrm{gu}}$ *queries, whereas the resulting PRF adversary* $\mathcal{C}$ *makes at most* $q_{\mathrm{en}} + q_{\mathrm{gu}}$ *queries.*

As the security bound shows, in order for CEC to have meaningful security, the minimal size of the right input m needs to be large enough (i.e., $m \geq n$).

4.1 Instantiating ECE and CEC

Instantiating the constructions ECE and CEC reduces to how one instantiates the expanding and compressing round functions. For the compressing PRF FC, one could use the Hash-then-PRF paradigm and instantiate the function with an efficient almost-universal hash function together with a fixed-input-size PRF that could be AES or the ChaCha20 block function. The FE could be instantiated using AES in Counter mode or the stream cipher ChaCha20.

Another option would be to instantiate FC with Xoofff [10], a so-called deck function. A deck function is a variable-input and variable-output length PRF, so it is also an excellent candidate for instantiating the expanding FE. Using Xoofff for both FE and FC enables us to instantiate our 3-round Feistel schemes with a single permutation-based primitive, which would also offer very competitive performance [4, 10].

5 HEC

We now present one of our main contributions, a construction called HEC (**H**ash–**E**ncipher–**C**ounter), which we base on a tweakable VIL cipher HCTR [24], originally proven to be STPRP secure. Our goal is to construct a cipher satisfying the weaker notion of RPRP security and a natural step in achieving that is to try and reduce the complexity of HCTR. The original HCTR construction consists of three layers, the first and the third one being an AXU hash function "compressing" layers that process the right part of the plaintext. The middle "expanding" layer is a simple counter mode. We modify HCTR in two ways to arrive at our construction HEC.

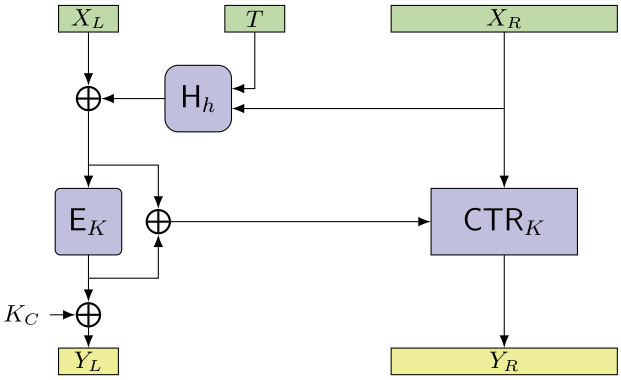

Fig. 8: Graphical representation of the HEC enciphering algorithm.

The first step is removing the lower hash layer. The second step is introducing a n-bit key K_C that is used for masking the left output value Y_L. The pseudocode of the HEC construction encipher and decipher algorithm is presented in Figure 7. A graphical representation of it is given in Figure 8.

Just removing the lower layer in HCTR is not enough to achieve RPRP security, the masking key needs to be present. We show in Appendix D an attack against RPRPd security of the variant that does not have the masking key K_C, therefore showing such construction would not satisfy RPRP security as well. The attack exploits the fact that one can make such a query to the construction's De oracle so that the decipher algorithm of the underlying blockcipher is queried with a value that has already been output by its encipher algorithm.

Note that the alteration we made to the HCTR construction to arrive at the HEC construction makes HEC insecure against an STPRP adversary. Querying (T, Y_L, Y_R) and (T, Y_L, Y'_R) to the decipher oracle, for $Y_R \neq Y'_R$, leads to the respective outputs (X_L, X_R) and (X'_L, X'_R). It will hold $X_R \oplus Y_R = X'_R \oplus Y'_R$, which would be true in the ideal world with a very small probability.

HEC *Security.* Continuing, we prove HEC is a secure RPRP. We use the H-coefficient technique in our proof, and the proof takes care of inputs of all valid lengths, i.e., inputs with a length that is not a multiple of blocksize n. Other relevant works on tweakable cipher constructions prove the security for inputs that end on a full block. Compared to the proof of other known RPRP scheme, namely that of the UIV scheme, the proof we give is much more involved since HEC construction is concrete, as opposed to the more abstract UIV. The HEC security theorem and the corresponding proof follow.

Theorem 4. *Let* HEC *be the construction defined in Figure 7 over the domain $\{0,1\}^n \times \{0,1\}^{\geq m}$, with H being an ϵ_1-AXU hash function. For any positive integer v and an adversary $\mathcal{A}$ making q_{en} encipher queries, q_{de} decipher queries and q_{gu} guess queries, such that every query input is at most ln bits long, there exists an adversary $\mathcal{B}$ such that*

$$\mathbf{Adv}^{\mathrm{RPRP}}_{\mathtt{HEC}}(\mathcal{A}, v) \leq \mathbf{Adv}^{\mathrm{SPRP}}_{\mathsf{E}}(\mathcal{B}) + \frac{q^2}{2^{n+m}} + q_1 q \epsilon_1 + \frac{q_1 q l^2}{2^{n-2}} + \frac{q_1 q l}{2^{n-1}} + \frac{q_1^2}{2^n} + 2 q_{\text{gu}} v \max\{\frac{1}{2^{n-1}}, \epsilon_1\},$$

where $q = q_{\text{en}} + q_{\text{de}} + q_{\text{gu}}$, $q_1 = q_{\text{en}} + q_{\text{de}}$ and under the assumption $q \leq 2^{n+m-1}$. The resulting SPRP *adversary $\mathcal{B}$ makes at most ql oracle queries in total to its own encipher and decipher oracle.*

Proof. Without loss of generality, we assume that the adversary does not make redundant queries. That is, the adversary does not repeat queries to either of the oracles or make queries that the game will restrict.

Our starting game is the real world ($b = 1$) of the RPRP game. Using the standard argument, we first replace the blockcipher E in the construction with a random permutation Π. This adds a SPRP advantage term of E to the bound. We have

$$\mathbf{Adv}^{\mathrm{RPRP}}_{\mathtt{HEC}}(\mathcal{A}, v) \leq \mathbf{Adv}^{\mathrm{SPRP}}_{\mathsf{E}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{RPRP}}_{\mathtt{HEC}^*}(\mathcal{A}, v), \quad (1)$$

```
$\textsc{En}(T, X_L, X_R)$
if $\widetilde{\Pi}(T, X_L, X_R) \neq \bot$ then
    $(Y_L, Y_R) \leftarrow \widetilde{\Pi}(T, X_L, X_R)$
else
    $\mathcal{S} \leftarrow \mathsf{rng}(\widetilde{\Pi}(T, \cdot, \cdot))$
    $(Y_L, Y_R) \leftarrow\!\!\$\ \{0,1\}^{n+|X_R|} \setminus \mathcal{S}$
    $\widetilde{\Pi}(T, X_L, X_R) \leftarrow (Y_L, Y_R)$
    $\widetilde{\Pi}^{-1}(T, Y_L, Y_R) \leftarrow (X_L, X_R)$
return $(Y_L, Y_R)$

$\textsc{De}(T, Y_L, Y_R)$
if $\widetilde{\Pi}^{-1}(T, Y_L, Y_R) \neq \bot$ then
    $(X_L, X_R) \leftarrow \widetilde{\Pi}^{-1}(T, Y_L, Y_R)$
else
    $\mathcal{S} \leftarrow \mathsf{dom}(\widetilde{\Pi}(T, \cdot, \cdot))$
    $(X_L, X_R) \leftarrow\!\!\$\ \{0,1\}^{n+|Y_R|} \setminus \mathcal{S}$
    $\widetilde{\Pi}(T, X_L, X_R) \leftarrow (Y_L, Y_R)$
    $\widetilde{\Pi}^{-1}(T, Y_L, Y_R) \leftarrow (X_L, X_R)$
return $(X_L, X_R)$

$\textsc{Gu}(T, Y_L, Y_R, \boldsymbol{V})$
return false
```

Fig. 9: The ideal world for H-coefficient technique application in Theorem 4 (HEC is RPRP).

where $\mathtt{HEC}^*$ is the HEC construction having a random permutation Π instead of a blockcipher. We now aim to apply the H-coefficient technique and Theorem 1 in order to bound $\mathcal{A}$'s distinguishing advantage between $\mathtt{HEC}^*$ and the ideal world ($b = 0$) of HEC's RPRP game.

The first step in doing that is defining the real and ideal worlds in the H-coefficient technique. In the real world, the adversary interacts with $\mathtt{HEC}^*$ via oracles $\textsc{En}, \textsc{De}$ and $\textsc{Gu}$. In the ideal world, the adversary has access to $\textsc{En}, \textsc{De}$ and $\textsc{Gu}$ oracles given in Figure 9. In words, for each tweak T and input length $n + |X_R|$ (or $n + |Y_R|$), a separate random permutation is lazily sampled with the help of the table $\widetilde{\Pi}$. Note that the ideal world corresponds to the ideal world of the RPRP game.

The transcript τ is structured as follows

$$\tau = (\tau', h, K_C)\,,$$

where τ' contains the queries adversary made during the interaction with the real or ideal world. The h and K_C in the real world correspond to the real hash and masking key appearing in $\mathtt{HEC}^*$. On the other hand, in the ideal world, the two keys are sampled at the end, the exact sampling procedure being explained later.

As for τ', two types of queries are stored there:

1. Queries $(T^i, X_L^i, X_R^i, Y_L^i, Y_R^i, R^i)$, corresponding to the queries to $\textsc{En}$ and $\textsc{De}$ oracles. If the input (or output) length is a multiple of blockcipher size, then $R^i = \varepsilon$. Otherwise, for X_R^i that has k full blocks and r more bits, where $r < n$, R^i in the real world contains the last $n - r$ bits of the last blockcipher (permutation) output in counter mode. That is,

$$R^i := \lceil \Pi(IV \oplus \langle k+1 \rangle_2) \rceil_{n-r}$$

In the ideal world R^i is sampled by the simulator S at the end. We define S shortly.
2. Queries $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i, o^i, X_L^i, X_R^i, R^i)$, corresponding to the queries to $\textsc{Gu}$ oracle. The variable o^i corresponds to the answer of the guess oracle, and for a query from an attainable transcript, its value will always be **false**. In the real world, (X_L^i, X_R^i) corresponds to (X_L, X_R) that would internally be deciphered on input (T^i, Y_L^i, Y_R^i). The value R^i is defined analogously as in the case of encipher and decipher query. As for the ideal world, the simulator defined below samples these values.

The simulator S runs in the ideal world after the adversary has finished its interaction with the oracles, and executes the following steps (in the given order).

i. It uniformly samples the hash key h and the masking key K_C.
ii. It iterates through all $\textsc{En}$ and $\textsc{De}$ queries and for each query $(T^i, X_L^i, X_R^i, Y_L^i, Y_R^i)$, it sets $R^i := \epsilon$ if X_R^i has k full blocks. Otherwise, X_R^i does not end on a full block, but it has k full blocks and r more bits, where $r < n$. The simulator S in that case sets $R^i \leftarrow\!\!\$\ \{0,1\}^{n-r}$.
iii. It iterates through all $\textsc{Gu}$ queries and for each query $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i)$ it first determines if the triple (T^i, Y_L^i, Y_R^i) is fresh.

- (T^i, Y_L^i, Y_R^i) *appears for the first time in a guess query*: The simulator checks if Y_L^i is new. We call Y_L^i new if there is no En or De query (occurring either before or after this i-th guess query) or an earlier Gu query in τ' that contains Y_L^i.

  If Y_L^i is new, then (X_L^i, X_R^i) is sampled according to the permutation $\widetilde{\Pi}^{-1}(T^i, \cdot, \cdot)$. The variable $R^i := \epsilon$ if $r = 0$, otherwise $R^i \leftarrow\!\!\$ \{0,1\}^{n-r}$.

  If Y_L^i is not new, let j-th query be the first En, De query $(T^j, X_L^j, X_R^j, Y_L^j, Y_R^j, R^j)$ or the first Gu query[4] $(T^j, Y_L^j, Y_R^j, \boldsymbol{V}^j, \mathbf{false}, X_L^j, X_R^j, R^j)$ such that $Y_L^i = Y_L^j$. Then set $X_R^i := X_R^j \oplus Y_R^j \oplus Y_R^i$, $X_L^i := X_L^j \oplus \mathsf{H}_h(T^j, X_R^j) \oplus \mathsf{H}_h(T^i, X_R^i)$ and $R^i := R^j$. Note that the term $X_R^j \oplus Y_R^j$ would be the key stream produced by the counter mode if we were in the real world.
- (T^i, Y_L^i, Y_R^i) *does not appear for the first time in a guess query*: The simulator takes the values (X_L^j, X_R^j, R^j) from some previous j-th guess query with the same (T^i, Y_L^i, Y_R^i) and sets

$$(X_L^i, X_R^i, R^i) := (X_L^j, X_R^j, R^j).$$

In the rest of the proof, we assume that l_i denotes the length of the input of the i-th query. We will sometimes write the right value X_R^i of length $kn + r$ as

$$x_1^i \| x_2^i \| \cdots \| x_k^i \| x_{k+1}^i,$$

where $0 \leq r < n$. For $1 \leq j \leq k$, $x_j^i \in \{0,1\}^n$. If $r \neq 0$, then $x_{k+1}^i \in \{0,1\}^r$, otherwise $x_{k+1}^i = \varepsilon$. We do the same for right value Y_R^i and write it as

$$y_1^i \| y_2^i \| \cdots \| y_k^i \| y_{k+1}^i.$$

Defining and bounding the bad transcripts. We now define what it means for an attainable transcript to be *bad*. The intuition for the following bad transcript conditions is as follows. The [B1.*] conditions ensure that for two En/De queries, there will be no collisions in the input or the output of the underlying blockcipher, that is, permutation. The [B2.*] conditions are similar to [B1.*] conditions. They ensure that for one En/De and one Gu query that has new Y_L, there will be no collisions in the input or the output of the underlying blockcipher, that is, permutation. The condition [B3] excludes guess oracle queries that would be deemed successful in the real world.

Definition 7. *A transcript $\tau = (\tau', h, K_C)$ is called bad, if in τ' there exist:*

[B1] *Two En / De queries $(T^i, X_L^i, X_R^i, Y_L^i, Y_R^i, R^i)$ and $(T^j, X_L^j, X_R^j, Y_L^j, Y_R^j, R^j)$, with $|X_R^i| = k_i n + r_i$, $|X_R^j| = k_j n + r_j$ and $0 \leq r_i, r_j < n$, such that one of the following conditions hold:*

[B1.1] $X_L^i \oplus \mathsf{H}_h(T^i, X_R^i) = X_L^j \oplus \mathsf{H}_h(T^j, X_R^j)$, *with $i \neq j$.*

[B1.2] $IV^i \oplus \langle \mathrm{ctr}_i \rangle_2 = IV^j \oplus \langle \mathrm{ctr}_j \rangle_2$ *with* $\mathrm{ctr}_i \in \{1, \ldots, k_i + 1\}$, $\mathrm{ctr}_j \in \{1, \ldots, k_j + 1\}$ *and $i \neq j$.*

[B1.3] $X_L^i \oplus \mathsf{H}_h(T^i, X_R^i) = IV^j \oplus \langle \mathrm{ctr}_j \rangle_2$ *with* $\mathrm{ctr}_j \in \{1, \ldots, k_j + 1\}$.

[B1.4] $Y_L^i \oplus K_C = Y_L^j \oplus K_C$, *with $i \neq j$.*

[B1.5] $Y_L^i \oplus K_C = x_{\mathrm{ctr}_j}^j \oplus y_{\mathrm{ctr}_j}^j$ *with* $\mathrm{ctr}_j \in \{1, \ldots, k_j\}$,
or, if $r_j > 0$, $Y_L^i \oplus K_C = (x_{k_j+1}^j \oplus y_{k_j+1}^j) \| R^j$

[B1.6] $x_{\mathrm{ctr}_i}^i \oplus y_{\mathrm{ctr}_i}^i = x_{\mathrm{ctr}_j}^j \oplus y_{\mathrm{ctr}_j}^j$ *with* $\mathrm{ctr}_i \in \{1, \ldots, k_i\}$ *and* $\mathrm{ctr}_j \in \{1, \ldots k_j\}$,
or, if $r_j > 0$, $x_{\mathrm{ctr}_i}^i \oplus y_{\mathrm{ctr}_i}^i = (x_{k_j+1}^j \oplus y_{k_j+1}^j) \| R^j$ with $\mathrm{ctr}_i \in \{1, \ldots, k_i\}$,
or, if $i \neq j$, $r_i > 0$ and $r_j > 0$, $(x_{k_i+1}^i \oplus y_{k_i+1}^i) \| R^i = (x_{k_j+1}^j \oplus y_{k_j+1}^j) \| R^j$.

[B2] *One En / De query and one Gu query with new Y_L, such that one of the following conditions hold:*

[B2.1] $X_L^i \oplus \mathsf{H}_h(T^i, X_R^i) = X_L^j \oplus \mathsf{H}_h(T^j, X_R^j)$, *with i being En/De query and j being Gu query or vice versa.*

[B2.2] $IV^i \oplus \langle \mathrm{ctr}_i \rangle_2 = IV^j \oplus \langle \mathrm{ctr}_j \rangle_2$ *with* $\mathrm{ctr}_i \in \{1, \ldots, k_i + 1\}$ *and* $\mathrm{ctr}_j \in \{1, \ldots, k_j + 1\}$, *and with i being En/De query and j being Gu query or vice versa.*

[4] In case of a Gu query, it will hold $j < i$.

[B2.3] $X_L^i \oplus \mathsf{H}_h(T^i, X_R^i) = IV^j \oplus \langle \mathrm{ctr}_j \rangle_2$ *with* $\mathrm{ctr}_j \in \{1, \ldots, k_j + 1\}$, *and with* i *being* En/De *query and* j *being* Gu *query or vice versa.*

[B2.4] i *being* En/De *query and* j *being* Gu *query, or vice versa, and:*
$Y_L^i \oplus K_C = x_{\mathrm{ctr}_j}^j \oplus y_{\mathrm{ctr}_j}^j$ *with* $\mathrm{ctr}_j \in \{1, \ldots, k_j\}$,
or, if $r_j > 0$, $Y_L^i \oplus K_C = (x_{k_j+1}^j \oplus y_{k_j+1}^j) \| R^j$.

[B2.5] i *being* En/De *query and* j *being* Gu *query, or vice versa, and:*
$x_{\mathrm{ctr}_i}^i \oplus y_{\mathrm{ctr}_i}^i = x_{\mathrm{ctr}_j}^j \oplus y_{\mathrm{ctr}_j}^j$ *with* $\mathrm{ctr}_i \in \{1, \ldots, k_i\}$ *and* $\mathrm{ctr}_j \in \{1, \ldots k_j\}$,
or, if $r_j > 0$, $x_{\mathrm{ctr}_i}^i \oplus y_{\mathrm{ctr}_i}^i = (x_{k_j+1}^j \oplus y_{k_j+1}^j) \| R^j$ *with* $\mathrm{ctr}_i \in \{1, \ldots, k_i\}$,
or, if $i \neq j$, $r_i > 0$ *and* $r_j > 0$, $(x_{k_i+1}^i \oplus y_{k_i+1}^i) \| R^i = (x_{k_j+1}^j \oplus y_{k_j+1}^j) \| R^j$.

[B3] *One* Gu *query* $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i, o^i, X_L^i, X_R^i, R^i)$ *such that* $X_L^i \in \boldsymbol{V}^i$.

Now, let τ be some attainable transcript in the ideal world. We bound the probabilities of above defined conditions holding true in the ideal world.

[B1.1] We rewrite the condition as

$$\mathsf{H}_h(T^i, X_R^i) \oplus \mathsf{H}_h(T^j, X_R^j) = X_L^i \oplus X_L^j.$$

The equation above holds, by the AXU property of H, with probability at most ϵ_1. Summing over all i and j, with $i \neq j$, we get the term

$$\binom{q_1}{2} \epsilon_1 \leq \frac{q_1^2 \epsilon_1}{2}. \tag{2}$$

[B1.2] Without loss of generality, assume $i < j$. By expanding IV^i and IV^j, the condition becomes

$$X_L^i \oplus \mathsf{H}_h(T^i, X_R^i) \oplus Y_L^i \oplus \langle \mathrm{ctr}_i \rangle_2 = X_L^j \oplus \mathsf{H}_h(T^j, X_R^j) \oplus Y_L^j \oplus \langle \mathrm{ctr}_j \rangle_2$$

We fix some ctr_i and ctr_j. If j-th query was an encipher query, we bound the equation above over the distribution of Y_L^j. Otherwise j-th query was a decipher query and then we bound the equation over the distribution of X_L^j. Assuming $q_1 \leq 2^{n+m-1} \leq 2^{l_j - 1}$, the upper bound for the equation above holding true is

$$\frac{2^{l_j - n}}{2^{l_j} - (j-1)} \leq \frac{2^{l_j - n}}{2^{l_j} - q_1} \leq \frac{2^{l_j - n}}{2^{l_j - 1}} = \frac{1}{2^{n-1}}.$$

Summing up over all i and j, with $i \neq j$, we arrive at the term

$$\binom{q_1}{2} \frac{l^2}{2^{n-1}} \leq \frac{q_1^2 l^2}{2^n}. \tag{3}$$

[B1.3] There are two possibilities here. The first one is, $i \neq j$. The equation, when IV^j is expanded, becomes

$$X_L^i \oplus \mathsf{H}_h(T^i, X_R^i) = X_L^j \oplus \mathsf{H}_h(T^j, X_R^j) \oplus Y_L^j \oplus K_C \oplus \langle \mathrm{ctr}_j \rangle_2.$$

For a fixed ctr_j, the probability of the equation being true is $\frac{1}{2^n}$, taken over the randomness of K_C. Summing up over all i, j and ctr_j, with $i \neq j$, the total probability for condition [B1.3] in this case is at most $\frac{q_1(q_1-1)l}{2^n}$.

The other option is that $i = j$. The condition equation is then reduced to $Y_L^i \oplus \langle \mathrm{ctr}_i \rangle_2 = K_C$. The probability of the equation being true is $\frac{1}{2^n}$ in this case as well, taken over the randomness of K_C. There are q possibilities for i, therefore summing over i and ctr_j, the bound becomes $\frac{q_1 l}{2^n}$.

Adding up the bounds of both cases, the total term for bounding the probability of this condition holding true is

$$\frac{q_1^2 l}{2^n}. \tag{4}$$

[B1.4] Without loss of generality, assume $i < j$. The condition of [B1.4] is equivalent to $Y_L^i = Y_L^j$. We differentiate 4 subcases here.

- *Both queries are encipher queries.* The probability of the condition being true is $\frac{1}{2^{n-1}}$, taken over the draw of Y_L^j and assuming $q_1 \leq 2^{n+m-1}$.

- *Both queries are decipher queries.* The probability of the condition being true is 0, since the adversary would not make j-th query with Y_L^j repeating.
- *i-th query is encipher query, j-th query is decipher query.* The probability of the condition being true is 0, since the adversary would not make j-th query with Y_L^j repeating.
- *i-th query is decipher query, j-th query is encipher query.* The probability of the condition being true is $\frac{1}{2^{n-1}}$, taken over the draw of Y_L^j and assuming $q_1 \leq 2^{n+m-1}$.

Summing up over all i and j, the total bound for condition [B1.4] becomes

$$\binom{q_1}{2}\frac{1}{2^{n-1}} \leq \frac{q_1^2}{2^n}. \tag{5}$$

[B1.5] We differentiate here two subcases.

- $l_j = k_j n$. For a fixed ctr_j, the probability of the equation being true is $\frac{1}{2^n}$, taken over the randomness of K_C.
- $l_j = k_j n + r_j$, *for* $r_j > 0$. The probability of bad condition occurring can be rewritten as

  $$\Pr\Big[\lfloor Y_L^i \oplus K_C \rfloor_{r_j} = (x^j_{k_j+1} \oplus y^j_{k_j+1}) \wedge \lceil Y_L^i \oplus K_C \rceil_{n-r_j} = R^j\Big].$$

  The probability of the first equation holding true can be bounded by $\frac{1}{2^{r_j}}$, taken over randomness of K_C, and the probability of second equation being true is $\frac{1}{2^{n-r_j}}$, since R^j is sampled uniformly at random. In total, the probability is bounded by

  $$\frac{1}{2^n}.$$

Summing up over all i, j and ctr_j, the total bound for condition [B1.5] holding true is

$$\frac{q_1^2 l}{2^n}. \tag{6}$$

[B1.6] We differentiate three subcases here.

- $r_i = r_j = 0$. Assume first that $i = j$. In that case the, probability of a condition being true for some fixed $\text{ctr}_{i_1} \neq \text{ctr}_{i_2}$ is at most $\frac{1}{2^{n-1}}$, taken over the sampling of y^i's in case the query was an encipher query, or x^i's in case the query was a decipher query.

  In case of $i \neq j$, the probability is calculated analogously and one gets the same bound $\frac{1}{2^{n-1}}$.
- $r_j > 0$. We fix some ctr_i. The probability of bad condition occurring can be rewritten as

  $$\Pr\Big[\lfloor x^i_{\text{ctr}_i} \oplus y^i_{\text{ctr}_i} \rfloor_{r_j} = (x^j_{k_j+1} \oplus y^j_{k_j+1}) \wedge \lceil x^i_{\text{ctr}_i} \oplus y^i_{\text{ctr}_i} \rceil_{n-r_j} = R^j\Big].$$

  The equation above is bounded by $\frac{1}{2^{r_j-1}}\frac{1}{2^{n-r_j}} = \frac{1}{2^{n-1}}$, taken over the distribution of R^j and $x^i_{\text{ctr}_i}/y^i_{\text{ctr}_i}$ or $x^j_{k_j+1}/y^j_{k_j+1}$.
- *Both* $r_i > 0$ *and* $r_j > 0$. Without loss of generality assume $r_i \leq r_j$. The probability of bad condition occurring can be rewritten as

  $$\Pr\Big[\lfloor x^i_{k_i+1} \oplus y^i_{k_i+1} \rfloor_{r_i} = \lfloor x^j_{k_j+1} \oplus y^j_{k_j+1} \rfloor_{r_i} \wedge R^i = \lceil x^j_{k_j+1} \oplus y^j_{k_j+1} \rceil_{n-r_i}\Big].$$

  This is bounded by $\frac{1}{2^{r_i-1}}\frac{1}{2^{n-r_i}} = \frac{1}{2^{n-1}}$, where the calculation is analogous to the calculation from the previous subcase.

Summing up over all i, j, ctr_i and ctr_j, the final bound for condition [B1.6] occurring is

$$\frac{q_1^2 l^2}{2^{n-1}}. \tag{7}$$

[B2.1] This condition holds true with probability at most

$$q_1 q_{\text{gu}} \epsilon_1, \tag{8}$$

where the probability is calculated similarly as in condition [B1.1].

[B2.2] This condition holds true with probability at most

$$\frac{q_1 q_{\mathrm{gu}} l^2}{2^{n-1}}, \tag{9}$$

where the probability is calculated similarly as in condition [B1.2] and assuming $q \leq 2^{n+m-1}$.

[B2.3] This condition holds true with probability at most

$$\frac{q_1 q_{\mathrm{gu}} l}{2^n}, \tag{10}$$

where the probability is calculated similarly as in condition [B1.3].

[B2.4] This condition holds true with probability at most

$$\frac{q_1 q_{\mathrm{gu}} l}{2^n}, \tag{11}$$

where the probability is calculated similarly as in condition [B1.5].

[B2.5] This condition holds true with probability at most

$$\frac{q_1 q_{\mathrm{gu}} l^2}{2^{n-1}}, \tag{12}$$

where the probability is calculated similarly as in condition [B1.6] and assuming $q \leq 2^{n+m-1}$.

[B3] Let us fix some $X_L^* \in \boldsymbol{V}^i$. We immediately differentiate two cases. The first one is when Y_L^i is new. In that case, (X_L^i, X_R^i) is sampled according to $\widetilde{\Pi}$ and it holds

$$\Pr\big[X_L^i = X_L^*\big] \leq \frac{2^{l_i - n}}{2^{l_i} - (q_1 + q_{\mathrm{gu}})} \leq \frac{1}{2^{n-1}},$$

assuming $q \leq 2^{n+m-1} \leq 2^{l_i - 1}$. The second case is when Y_L^i is *not* new. Then it holds $X_L^i = X_L^j \oplus \mathsf{H}_h(T^j, X_R^j) \oplus \mathsf{H}_h(T^i, X_R^i)$, where the j-th query is the one in which Y_L^i appears in for the first time. If $(T^i, X_R^i) \neq (T^j, X_R^j)$, we can reduce the probability of the equation $X_L^* = X_L^i$ holding true to ϵ_1. Otherwise $(T^i, X_R^i) = (T^j, X_R^j)$ and the equation reduces to $X_L^* = X_L^j$, which can again be bounded by $\frac{1}{2^{n-1}}$. The bound, for the case when Y_L^i is not new, is then $\max\{\frac{1}{2^{n-1}}, \epsilon_1\}$. Summing up over all X_L^* in $\boldsymbol{V}^i$ and then over all guess oracle queries, we have that the probability of the condition [B3] being true is at most

$$2 q_{\mathrm{gu}} v \max\{\frac{1}{2^{n-1}}, \epsilon_1\}. \tag{13}$$

Adding up the bounds in (2) − (13) we have that the probability of an attainable transcript τ in the ideal world being bad is bounded by

$$\epsilon_{\mathrm{bad}} \leq q_1 q \epsilon_1 + \frac{q_1 q l^2}{2^{n-2}} + \frac{q_1 q l}{2^{n-1}} + \frac{q_1^2}{2^n} + 2 q_{\mathrm{gu}} v \max\{\frac{1}{2^{n-1}}, \epsilon_1\}.$$

Bounding the ratio of good transcripts. Fix some good and an attainable transcript (τ', h, K_C). We split the encipher, decipher and guess queries that have new Y_L in τ' into two disjoint sets τ_1' and τ_2'. The set τ_1' contains queries whose length is a multiple of n and τ_2' contains all other En, De and Gu queries (with new Y_L). We note here that we defined the term of "new Y_L" in the ideal world, but the "new Y_L" has the same meaning in the real world. Furthermore, each of τ_1' and τ_2' is further "decomposed" into smaller disjoint subsets that only contain queries of the same length. That is, for $l_{1,1}, l_{1,2}, ..., l_{1,c_1}$, where every $l_{1,i}$ is a multiple of n, we have disjoint sets $\tau_{1,1}, \ldots, \tau_{1,c_1}$, with $\tau_{1,i}$ containing queries of length $l_{1,i}$. Therefore, it holds

$$\tau_1' = \tau_{1,1} \cup \cdots \cup \tau_{1,c_1}.$$

Similarly, for $l_{2,1}, \ldots, l_{2,c_2}$, where every $l_{2,i}$ is not a multiple of n, we have disjoint sets $\tau_{2,1}, \ldots, \tau_{2,c_2}$, with $\tau_{2,i}$ containing queries of length $l_{2,i}$. It holds

$$\tau_2' = \tau_{2,1} \cup \cdots \cup \tau_{2,c_2}.$$

In addition, for queries in τ'_1 we let $k'_{1,i}$ denote the number of blocks in the whole input[5], i.e. $l_{1,i} = k'_{1,i}n$. With $k'_{2,i}$ we denote the number of full blocks for a query in τ'_2 with length $l_{2,i}$, i.e. $l_{2,i} = k'_{2,i}n + r_{2,i}$. We denote the cardinality of set $\tau_{b,i}$ with $t_{b,i}$. We also introduce an equivalence relation $\sim_T$, where two queries from set $\tau_{b,i}$ are related if and only if they have the same tweak T. This equivalence relation partitions the set $\tau_{b,i}$ into equivalence classes by the tweak T, and there will be $w[b,i]$ classes with j-th equivalence class having $t_{b,i,j}$ number of queries in it. It then holds $t_{b,i} = t_{b,i,1} + \cdots + t_{b,i,w[b,i]}$, for $w[b,i]$ being the number of queried tweaks for queries in $\tau_{b,i}$. Finally, with uyl we denote the number of guess oracle queries that contain new Y_L and we let $\mathcal{H}$ denote the key space of the HEC's AXU hash function H.

Ideal world. The interpolation probability for the hash key h and the masking key K_C is $\frac{1}{|\mathcal{H}|}\frac{1}{2^n}$. The interpolation probabilities of queries in τ'_1 and τ'_2 are

$$\prod_{i=1}^{c_1} \frac{1}{(2^{l_{1,i}})_{t_{1,i,1}} \cdots (2^{l_{1,i}})_{t_{1,i,w[1,i]}}} \text{ and } \prod_{i=1}^{c_2} \frac{1}{(2^{l_{2,i}})_{t_{2,i,1}} \cdots (2^{l_{2,i}})_{t_{2,i,w[2,i]}}} \frac{1}{2^{(n-r_{2,i})t_{2,i}}},$$

respectively. As for the interpolation probability of guess oracle queries that do not contain a new Y_L, we fix some such query $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i, \mathbf{false}, X_L^i, X_R^i, R^i)$. Since Y_L is not new, there exists some En, De or Gu query with the same Y_L. The variables X_L^i, X_R^i and R^i then have the following value

$$X_R^i = X_R^j \oplus Y_R^j \oplus Y_R^i,\ X_L^i = X_L^j \oplus \mathsf{H}_h(T^j, X_R^j) \oplus \mathsf{H}_h(T^i, X_R^i),\ R^i = R^j.$$

The values in the right-hand side of the three equations above are already fixed, so the interpolation probability for the triple (X_R^i, X_L^i, R^i) is equal to 1.

In total, the interpolation probability for a transcript τ in the ideal world $\Pr[X_i = \tau]$ is

$$\frac{1}{|\mathcal{H}|}\frac{1}{2^n} \times \prod_{i=1}^{c_1} \frac{1}{(2^{l_{1,i}})_{t_{1,i,1}} \cdots (2^{l_{1,i}})_{t_{1,i,w[1,i]}}} \times \prod_{i=1}^{c_2} \frac{1}{(2^{l_{2,i}})_{t_{2,i,1}} \cdots (2^{l_{2,i}})_{t_{2,i,w[2,i]}}} \frac{1}{2^{(n-r_{2,i})t_{2,i}}} \times 1^{q_{\mathrm{gu}}-\mathsf{uyl}}.$$

Real world. The interpolation probability for the hash key h and the masking key K_C is $\frac{1}{|\mathcal{H}|}\frac{1}{2^n}$ in the real world as well. For queries in τ'_1 and τ'_2 we know there are no input and output collisions to the underlying blockcipher (permutation). Then, for example for some j-th query in τ'_1 that has $k'_{1,i}$ blocks, the interpolation probability that its input maps to its output is

$$\frac{1}{(2^n - \sigma)(2^n - \sigma - 1)\cdots(2^n - \sigma - (k'_{1,i} - 1))},$$

where σ represents the number of blocks processed in all the queries preceding the i-th query. By the above, the interpolation probability in total for queries in τ'_1 and τ'_2 is

$$\frac{1}{(2^n)_{t_{1,1}k'_{1,1}+\cdots+t_{1,c_1}k'_{1,c_1}+t_{2,1}(k'_{2,1}+1)+\cdots+t_{2,c_2}(k'_{2,c_2}+1)}}.$$

As for the guess oracle queries that do not have a new Y_L, considering we have already "fixed" the values related to this Y_L (e.g., the $IV = \Pi^{-1}(Y_L \oplus K_C) \oplus Y_L \oplus K_C$ and with that the keystream produced by the counter mode), the interpolation probability for X_R^i appearing in that guess query transcript will be 1. Similarly for $X_L^i = \Pi^{-1}(Y_L \oplus K_C) \oplus \mathsf{H}_h(X_R^i)$, everything on the right-hand side has already been fixed and therefore the X_L^i appears in that transcript with probability 1. In total, the interpolation probability for a transcript τ in the real world $\Pr[X_r = \tau]$ is

$$\frac{1}{|\mathcal{H}|}\frac{1}{2^n} \times \frac{1}{(2^n)_{t_{1,1}k'_{1,1}+\cdots+t_{1,c_1}k'_{1,c_1}+t_{2,1}(k'_{2,1}+1)+\cdots+t_{2,c_2}(k'_{2,c_2}+1)}} \times 1^{q_{\mathrm{gu}}-\mathsf{uyl}}.$$

[5] Following the previous notation, it holds $k'_{1,i} = k_{1,i} + 1$, where $k_{1,i}$ is the number of full blocks in the right part of the input.

Interpolation ratio. Finally, the interpolation ratio $\frac{\Pr[X_r=\tau]}{\Pr[X_i=\tau]}$ for a good transcript τ is

$$\frac{\prod_{i=1}^{c_1}(2^{l_{1,i}})_{t_{1,i,1}}\cdots(2^{l_{1,i}})_{t_{1,i,w[1,i]}}\times\prod_{i=1}^{c_2}(2^{l_{2,i}})_{t_{2,i,1}}\cdots(2^{l_{2,i}})_{t_{2,i,w[2,i]}}2^{(n-r_{2,i})t_{2,i}}}{(2^n)_{t_{1,1}k'_{1,1}+\cdots+t_{1,c_1}k'_{1,c_1}+t_{2,1}(k'_{2,1}+1)+\cdots+t_{2,c_2}(k'_{2,c_2}+1)}}.$$

Going further, by applying the Lemma 2 that can be found in Appendix D both in the enumerator and the denominator, it follows that the term above is greater or equal than

$$\frac{\prod_{i=1}^{c_1}(2^{l_{1,i}})_{t_{1,i}}}{(2^n)_{t_{1,1}k'_{1,1}+\cdots+t_{1,c_1}k'_{1,c_1}}}\times\frac{\prod_{i=1}^{c_2}(2^{l_{2,i}})_{t_{2,i}}2^{(n-r_{2,i})t_{2,i}}}{(2^n)_{t_{2,1}(k'_{2,1}+1)+\cdots+t_{2,c_2}(k'_{2,c_2}+1)}}.$$

Applying Lemma 3 to the left term and then Lemma 2 again to the right term tells us the expression above is greater or equal than

$$\prod_{i=1}^{c_2}\frac{(2^{k_{2,i}n+r_{2,i}})_{t_{2,i}}2^{(n-r_{2,i})t_{2,i}}}{(2^n)_{t_{2,i}(k_{2,i}+1)}}.$$

Finally, with some more calculation and applying the Weierstrass inequality we get

$$\begin{aligned}\frac{\Pr[X_r=\tau]}{\Pr[X_i=\tau]}&\geq\prod_{i=1}^{c_2}\frac{(2^{k_{2,i}n+r_{2,i}}-q)^{t_{2,i}}2^{(n-r_{2,i})t_{2,i}}}{2^{nt_{2,i}(k_{2,i}+1)}}=\prod_{i=1}^{c_2}\left(\frac{(2^{k_{2,i}n+r_{2,i}}-q)2^{n-r_{2,i}}}{2^{n(k_{2,i}+1)}}\right)^{t_{2,i}}\\&=\prod_{i=1}^{c_2}\left(1-\frac{q}{2^{k_{2,i}n+r_{2,i}}}\right)^{t_{2,i}}\geq 1-q\sum_{i=1}^{c_2}\frac{t_{2,i}}{2^{k_{2,i}n+r_{2,i}}}\geq 1-q\sum_{i=1}^{c_2}\frac{t_{2,i}}{2^{n+m}}\geq 1-\frac{q^2}{2^{n+m}},\end{aligned}$$

therefore $\epsilon_{\text{ratio}}=\frac{q^2}{2^{n+m}}$.

Summing up (1), ϵ_{bad} and ϵ_{ratio} one achieves the bound from the theorem statement. □

6 RPRP Domain Extension

In the case of UIV and HEC, the size of the left domain $\mathcal{X}_L$ is inherently equal to the size of the underlying (tweakable) blockcipher. Typical (tweakable) blockciphers have a block size of at most 128 bits, as is the case for AES, for instance. This can be a limiting factor in some RPRP applications, namely, in using RPRPs as a building block to arrive at the order-resilient secure channel. Namely, if one considers the order-resilient secure channel construction from [13, Section 6], instantiated with the nonce-set AEAD scheme AwN, the overall security of the channel reduces to the security of the underlying RPRP scheme. The RPRP advantage term of the UIV scheme is bounded by

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{E}}^*}(\mathcal{B})+\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{F}}(\mathcal{C})+\frac{q_{\mathrm{gu}}v}{2^{n-1}}+\frac{q(q-1)}{2^{n+1}}+\frac{q_{\mathrm{en}}(q_{\mathrm{en}}-1)}{2^{n+1}}+\frac{q_1(q_1-1)}{2^{n+m+1}},$$

where $\widetilde{\mathsf{E}}^*$ and F are the underlying tweakable blockcipher and PRF, respectively.

The term $\frac{q_{\mathrm{gu}}v}{2^{n-1}}$ in that bound corresponds to the integrity term of the order-resilient secure channel, where the q_{gu} would be the number of forgery attempts the channel adversary makes. The product $q_{\mathrm{gu}}v$ can grow quickly in specific use cases. Firstly, certain application will "embed" information in the nonce, consequently making the v large (i.e. up to 2^{64}). Secondly, some applications with long-lived channels that cannot be rekeyed easily could need to withstand unlimited adversarial forgery attempts. Because of the two reasons above, the integrity term $\frac{q_{\mathrm{gu}}v}{2^{n-1}}$ can quickly become large, leading to a need to extend the left domain of the underlying RPRP. If one doubles the size of $\mathcal{X}_L$ from $\{0,1\}^n$ to $\{0,1\}^{2n}$, the term above becomes $\frac{q_{\mathrm{gu}}v}{2^{2n-1}}$.

In addition, we also get an interesting "side effect" of the domain extension. Namely, the other three independent terms in the bound improve as well, e.g. the term $\frac{q_{\mathrm{en}}(q_{\mathrm{en}}-1)}{2^{n+1}}$ becomes $\frac{q_{\mathrm{en}}(q_{\mathrm{en}}-1)}{2^{2n+1}}$. Assuming that the STPRP security of $\widetilde{\mathsf{E}}^*$ and PRF security of F can also be strengthened, the overall security of the UIV construction (and thus the order-resilient secure channel it builds) would improve. However, we do not investigate this "side effect" further in this work.

In the following subsections, we present two possible black-box solutions for extending the left domain of the UIV construction [13]. The graphical representations of the UIV enciphering algorithm and these extender constructions are given in Fig. 10. We will call the UIV using one of these extender constructions an *extended UIV*.

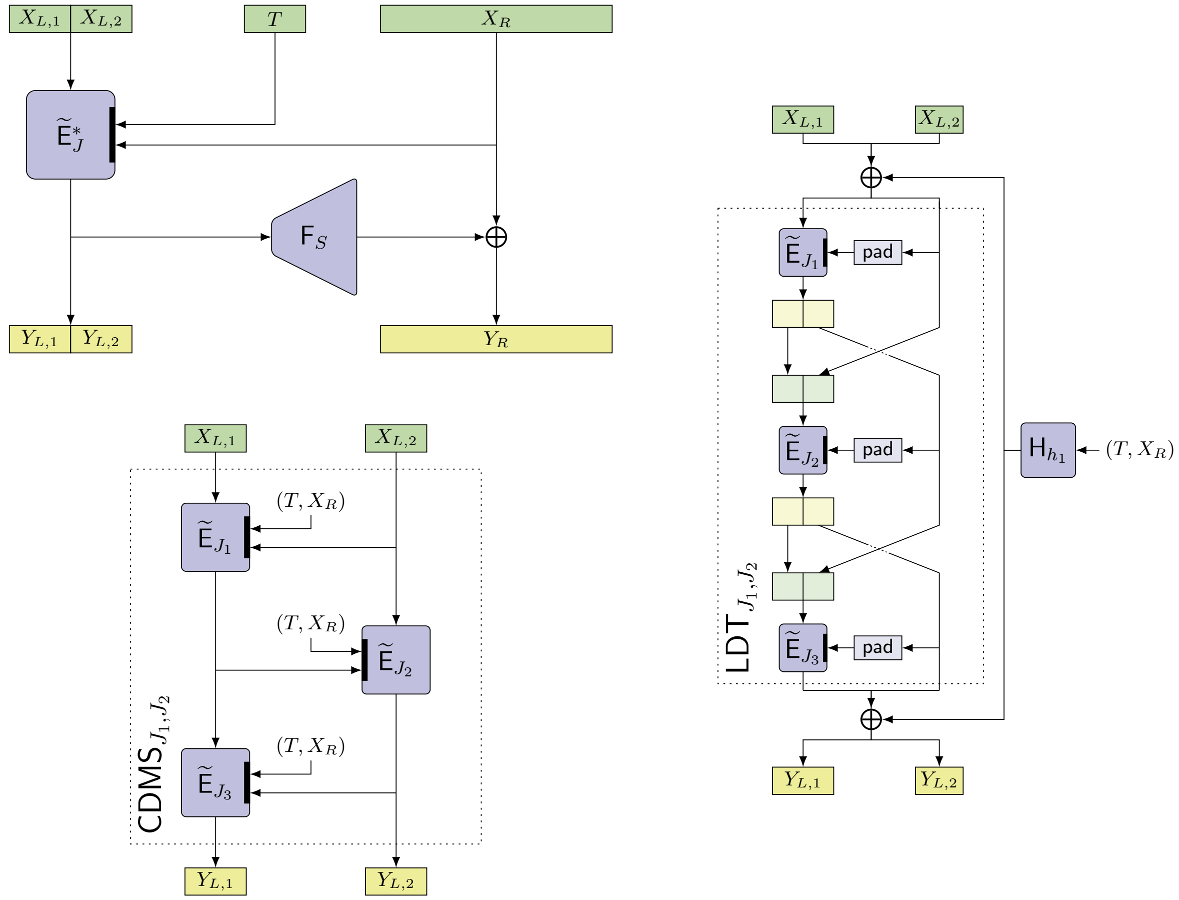
$X_{L,1}$
$X_{L,2}$
T
X_R
$\widetilde{\mathsf{E}}^*_J$
F_S
$Y_{L,1}$
$Y_{L,2}$
Y_R
(T, X_R)
$\widetilde{\mathsf{E}}_{J_1}$
$\widetilde{\mathsf{E}}_{J_2}$
$\widetilde{\mathsf{E}}_{J_3}$
CDMS_{J_1,J_2}
pad
H_{h_1}
LDT_{J_1,J_2}

Fig. 10: **Top left:** Extended UIV construction with a black-box tweakable blockcipher $\widetilde{\mathsf{E}}$; **Bottom left:** CDMS extender; **Right:** LRW2 + LDT extender.

We do not consider domain extenders for `HEC` in this work since the blockcipher used in the left part is also used throughout the whole construction. Therefore, replacing all appearances of it in a black-box manner would damage the performance. However, we leave finding specific domain extender options for `HEC` as an avenue for future work.

6.1 CDMS Extender

For our first extender, we use the construction of Coron et al. [8] that transforms a n-bit tweakable blockcipher into a $2n$-bit tweakable cipher using a 3-round Feistel scheme. We denote this construction with CDMS. The idea of using CDMS for domain extension inside a VIL cipher is not new. Shrimpton and Terashima utilized the same approach to instantiate their PIV construction [23].

The CDMS construction assumes the tweakable blockcipher admits tweaks of size ω. The size of the "outer" tweak, which is here (T, X_R), is then $\omega - n$. Denoting the underlying tweakable blockcipher with $\widetilde{\mathsf{E}}$, we can express the security of the UIV construction extended with CDMS using the following theorem, which is an adaptation of the original theorem for RPRP security of UIV [13, Theorem 1], using the result of Coron et al. [8] transform.

Theorem 5. *Let extended UIV be the scheme over the domain* $\{0,1\}^{2n} \times \{0,1\}^{\geq m}$ *using the* CDMS *extender. For any positive integer* v *and an adversary* $\mathcal{A}$ *making* q_{en} *encipher queries,* q_{de} *decipher queries and* q_{gu} *guess queries under the constraint that* $q_{\mathrm{gu}} v \leq 2^{2n-1}$*, there exist adversaries* $\mathcal{B}$ *and* $\mathcal{C}$ *such that*

$$\mathbf{Adv}^{\mathrm{RPRP}}_{\mathtt{UIV}[\mathsf{CDMS}]}(\mathcal{A}, v) \leq 3\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{E}}}(\mathcal{B}) + \frac{q^2}{2^{2n}} + \mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{F}}(\mathcal{C}) \\ + \frac{q_{\mathrm{gu}} v}{2^{2n-1}} + \frac{q^2}{2^{2n+1}} + \frac{q_{\mathrm{en}}^2}{2^{2n+1}} + \frac{q_1^2}{2^{2n+m+1}},$$

where $q = q_{\text{en}} + q_{\text{de}} + q_{\text{gu}}$ *and* $q_1 = q_{\text{en}} + q_{\text{de}}$. *The resulting* STPRP *adversary* $\mathcal{B}$ *makes at most* q_{en} *encipher queries and* $q_{\text{de}} + q_{\text{gu}}$ *decipher queries, whereas the resulting PRF adversary* $\mathcal{C}$ *makes at most* $q_{\text{en}} + q_{\text{de}} + q_{\text{gu}}$ *queries.*

6.2 LRW2 + LDT Extender

For our second extender we use the LRW2 [15] instantiation of $\widetilde{\mathsf{E}}^*$ with the 3-round length doubler construction LDT [7] by Chen, Mennink and Nandi serving as the underlying blockcipher. In contrast to the previous extender, the LDT extends the UIV domain to $\{0,1\}^{n+s} \times \{0,1\}^{\geq m}$, for a fixed $s \in [n+1, 2n-1]$. The advantage this extender offers is the variable length extension since the doubling of the left domain could be overabundant in some cases.

The LDT construction can "encipher" and "decipher" a $[n+1, 2n-1]$-bit string, using a n-bit tweakable blockcipher and a swapping function $\mathsf{swap}(X, Y) := (Y, X)$ that takes two inputs X, Y of size $1 \leq s \leq n-1$. In our case, LDT has fixed input size, i.e. fixed s, so the security for LDT we need is plain SPRP security, in contrast to the VSPRP (variable-input SPRP) notion used in [7].

We can express the security of the UIV construction extended with LRW2+LDT using the following theorem, which is an adaptation of the original theorem for RPRP security of UIV as well, using the result of the LRW2 transform [15].

Theorem 6. *Let extended UIV be the scheme over the domain* $\{0,1\}^{n+s} \times \{0,1\}^{\geq m}$ *using the* LRW2+LDT *extender and let* H *be a* ϵ_1*-AXU hash function with output space* $\{0,1\}^{n+s}$. *For any positive integer* v *and an adversary* $\mathcal{A}$ *making* q_{en} *encipher queries,* q_{de} *decipher queries and* q_{gu} *guess queries under the constraint that* $q_{\text{gu}} v \leq 2^{n+s-1}$, *there exist adversaries* $\mathcal{B}$ *and* $\mathcal{C}$ *such that*

$$\begin{aligned}\mathbf{Adv}^{\text{RPRP}}_{\mathtt{UIV}[\text{LRW2+}\mathsf{LDT}]}(\mathcal{A}, v) \leq \mathbf{Adv}^{\text{SPRP}}_{\mathsf{LDT}}(\mathcal{B}) + 3\epsilon_1 q^2 + \mathbf{Adv}^{\text{PRF}}_{\mathsf{F}}(\mathcal{C}) \\ + \frac{q_{\text{gu}} v}{2^{n+s-1}} + \frac{q^2}{2^{n+s+1}} + \frac{q_{\text{en}}^2}{2^{n+s+1}} + \frac{q_1^2}{2^{n+s+m+1}},\end{aligned}$$

where $q = q_{\text{en}} + q_{\text{de}} + q_{\text{gu}}$ *and* $q_1 = q_{\text{en}} + q_{\text{de}}$. *The resulting* STPRP *adversary* $\mathcal{B}$ *makes at most* q_{en} *encipher queries and* $q_{\text{de}} + q_{\text{gu}}$ *decipher queries, whereas the resulting PRF adversary* $\mathcal{C}$ *makes at most* $q_{\text{en}} + q_{\text{de}} + q_{\text{gu}}$ *queries.*

Interpreting Corollary 2 from [7], the SPRP advantage of the LDT construction in the bound above gives at least $\frac{2n}{3}$ bits of security.

One should take care when instantiating the LRW2 AXU hash function H since it needs to have a non-standard output size. One natural approach is concatenating and truncating two independently keyed AXU hash functions. Start with a n-bit AXU H' and construct a $2n$-bit AXU by concatenating two instances of H' keyed with two independent keys. After that, truncate the output to the desired output size $n' \in [n+1, 2n-1]$, which would incur a security loss of $2n - n'$ bits. Examples of concatenating and truncating AXU hash functions can be found in these works [11, 17, 21].

7 Conclusion

In this work, we gave multiple new results on rugged pseudorandom permutations. The first group of results introduced the RPRPd and RPRPd variations of the main RPRP definition. Then, we showed two interesting results about the 3-round unbalanced Feistel scheme. First, that the $\mathtt{ECE}$ scheme satisfies the RPRPd but not the RPRPg security, and second, that the $\mathtt{CEC}$ scheme satisfies the RPRPg but not the RPRPd security.

After that, we presented the $\mathtt{HEC}$ scheme and proved it RPRP secure, making it, together with the UIV scheme, the only other construction proven to be RPRP secure so far. In the end, we showed that the left domain of the UIV construction could be extended using the 3-round CDMS and LDT schemes in a black-box manner, providing better security than the "plain" UIV, which furthermore can be beneficial for order-resilient channels that are instantiated with UIV as presented in [13, Section 6].

Collectively, these findings contribute to a deeper understanding of the RPRP notion and show that it is more natural than it may seem.

Acknowledgments

We thank the anonymous ASIACRYPT 2023 reviewers for their constructive comments. This research was supported by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

1. Farzaneh Abed, Christian Forler, Eik List, Stefan Lucks, and Jakob Wenzel. RIV for robust authenticated encryption. In Thomas Peyrin, editor, *FSE 2016*, volume 9783 of *LNCS*, pages 23–42. Springer, Heidelberg, March 2016. doi:10.1007/978-3-662-52993-5_2.
2. Elena Andreeva, Amit Singh Bhati, Bart Preneel, and Damian Vizár. 1, 2, 3, fork: Counter mode variants based on a generalized forkcipher. *IACR Trans. Symm. Cryptol.*, 2021(3):1–35, 2021. doi:10.46586/tosc.v2021.i3.1-35.
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, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 105–125. Springer, Heidelberg, December 2014. doi:10.1007/978-3-662-45611-8_6.
4. Norica Bacuieti, Joan Daemen, Seth Hoffert, Gilles Van Assche, and Ronny Van Keer. Jammin' on the deck. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part II*, volume 13792 of *LNCS*, pages 555–584. Springer, Heidelberg, December 2022. doi:10.1007/978-3-031-22966-4_19.
5. Mihir Bellare and Phillip Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In Tatsuaki Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 317–330. Springer, Heidelberg, December 2000. doi:10.1007/3-540-44448-3_24.
6. Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 327–350. Springer, Heidelberg, May 2014. doi:10.1007/978-3-642-55220-5_19.
7. Yu Long Chen, Bart Mennink, and Mridul Nandi. Short variable length domain extenders with beyond birthday bound security. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part I*, volume 11272 of *LNCS*, pages 244–274. Springer, Heidelberg, December 2018. doi:10.1007/978-3-030-03326-2_9.
8. Jean-Sébastien Coron, Yevgeniy Dodis, Avradip Mandal, and Yannick Seurin. A domain extender for the ideal cipher. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 273–289. Springer, Heidelberg, February 2010. doi:10.1007/978-3-642-11799-2_17.
9. Paul Crowley, Nathan Huckleberry, and Eric Biggers. Length-preserving encryption with HCTR2. Cryptology ePrint Archive, Report 2021/1441, 2021. https://eprint.iacr.org/2021/1441.
10. Joan Daemen, Seth Hoffert, Gilles Van Assche, and Ronny Van Keer. The design of Xoodoo and Xoofff. *IACR Trans. Symm. Cryptol.*, 2018(4):1–38, 2018. doi:10.13154/tosc.v2018.i4.1-38.
11. Jean Paul Degabriele, Jérôme Govinden, Felix Günther, and Kenneth G. Paterson. The security of ChaCha20-Poly1305 in the multi-user setting. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 1981–2003. ACM Press, November 2021. doi:10.1145/3460120.3484814.
12. Jean Paul Degabriele, Vukašin Karadžić, Alessandro Melloni, Jean-Pierre Münch, and Martijn Stam. Rugged pseudorandom permutations and their applications. Presented at the IACR Real World Crypto Symposium 2022.
13. Jean Paul Degabriele and Vukašin Karadžić. Overloading the nonce: Rugged PRPs, nonce-set AEAD, and order-resilient channels. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 264–295. Springer, Heidelberg, August 2022. doi:10.1007/978-3-031-15985-5_10.
14. Avijit Dutta and Mridul Nandi. Tweakable HCTR: A BBB secure tweakable enciphering scheme. In Debrup Chakraborty and Tetsu Iwata, editors, *INDOCRYPT 2018*, volume 11356 of *LNCS*, pages 47–69. Springer, Heidelberg, December 2018. doi:10.1007/978-3-030-05378-9_3.
15. Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. *Journal of Cryptology*, 24(3):588–613, July 2011. doi:10.1007/s00145-010-9073-y.
16. Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2), 1988.
17. David A. McGrew and John Viega. The security and performance of the galois/counter mode of operation (full version). Cryptology ePrint Archive, Report 2004/193, 2004. https://eprint.iacr.org/2004/193.
18. Kazuhiko Minematsu and Tetsu Iwata. Building blockcipher from tweakable blockcipher: Extending FSE 2009 proposal. In Liqun Chen, editor, *13th IMA International Conference on Cryptography and Coding*, volume 7089 of *LNCS*, pages 391–412. Springer, Heidelberg, December 2011.
19. National Institute of Standards and Technology (NIST). The Third NIST Workshop on Block Cipher Modes of Operation, 2023. https://csrc.nist.gov/Events/2023/third-workshop-on-block-cipher-modes-of-operation.

20. Jacques Patarin. The "coefficients H" technique (invited talk). In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *SAC 2008*, volume 5381 of *LNCS*, pages 328–345. Springer, Heidelberg, August 2009. doi:10.1007/978-3-642-04159-4_21.
21. Phillip Rogaway. Bucket hashing and its application to fast message authentication. In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 29–42. Springer, Heidelberg, August 1995. doi:10.1007/3-540-44750-4_3.
22. Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 373–390. Springer, Heidelberg, May / June 2006. doi:10.1007/11761679_23.
23. Thomas Shrimpton and R. Seth Terashima. A modular framework for building variable-input-length tweakable ciphers. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 405–423. Springer, Heidelberg, December 2013. doi:10.1007/978-3-642-42033-7_21.
24. 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*, pages 175–188, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

Appendix

A RRND Security

In [13], the authors also present the RRND notion, a security notion accompanying the RPRP notion. In the RRND game, the adversary must distinguish between $\widetilde{\mathsf{EE}}$ and a tweakable two-sided random function $\widetilde{\mathsf{RR}}$. Compared to the RPRP game, the RRND game has one more set tracking values that need to be restricted by the game. The set in question is $\mathcal{P}$, which prohibits the adversary from forwarding queries from De to En oracle. In Fig. 11, we give the RRND game definition and in addition present one of its subvariants that we use, the RRNDg game. The RRNDg game is a RRND "equivalent" of the RPRPg game given in Section 3. For completeness, we reiterate the RRND advantage definition and present the

```
Game $\mathsf{RRND}_{\widetilde{\mathsf{EE}}}^{\mathcal{A},v}$ / $\mathsf{RRNDg}_{\widetilde{\mathsf{EE}}}^{\mathcal{A},v}$
  $K \leftarrow\!\!\$\ \mathcal{K}$
  $b \leftarrow\!\!\$\ \{0,1\}$
  $\mathcal{F}, \mathcal{R}, \mathcal{U}, \mathcal{P} \leftarrow \emptyset, \emptyset, \emptyset, \emptyset$
  $\widetilde{\mathsf{RR}} \leftarrow\!\!\$\ \text{2-Func}(\mathcal{T}, \mathcal{X}_L \times \mathcal{X}_R)$
  $b' \leftarrow \mathcal{A}^{\textsc{En},\textsc{De},\textsc{Gu}}$ / $\mathcal{A}^{\textsc{En},\textsc{Gu}}$
  return $b = b'$
```

```
De$(T, Y_L, Y_R)$
  if $Y_L \in \mathcal{F} \cup \mathcal{R}$
    return $\lightning$
  if $b = 0$
    $(X_L, X_R) \leftarrow \widetilde{\mathsf{RR}}(-, T, Y_L, Y_R)$
  else
    $(X_L, X_R) \leftarrow \widetilde{\mathsf{EE}}_K^{-1}(T, Y_L, Y_R)$
  $\mathcal{R} \xleftarrow{\cup} \{Y_L\}$; $\mathcal{U} \xleftarrow{\cup} \{(T, Y_L, Y_R)\}$
  $\mathcal{P} \xleftarrow{\cup} \{(T, X_L, X_R)\}$
  return $(X_L, X_R)$
```

```
En$(T, X_L, X_R)$
  if $(T, X_L, X_R) \in \mathcal{P}$
    return $\lightning$
  if $b = 0$
    $(Y_L, Y_R) \leftarrow \widetilde{\mathsf{RR}}(+, T, X_L, X_R)$
  else
    $(Y_L, Y_R) \leftarrow \widetilde{\mathsf{EE}}_K(T, X_L, X_R)$
  $\mathcal{F} \xleftarrow{\cup} \{Y_L\}$ ; $\mathcal{U} \xleftarrow{\cup} \{(T, Y_L, Y_R)\}$
  return $(Y_L, Y_R)$
```

```
Gu$(T, Y_L, Y_R, \boldsymbol{V})$
  if $((T, Y_L, Y_R) \in \mathcal{U}) \vee (|\boldsymbol{V}| > v)$
    return $\lightning$
  if $b = 0$
    return false
  else
    $(X_L, X_R) \leftarrow \widetilde{\mathsf{EE}}_K^{-1}(T, Y_L, Y_R)$
    return $X_L \in \boldsymbol{V}$
```

Fig. 11: The games used to define RRND and RRNDg security for a tweakable cipher $\widetilde{\mathsf{EE}}$.

RRNDg advantage definition.

Definition 8 (RRND / RRNDg **Advantage).** *Let* $\widetilde{\mathsf{EE}}$ *be a tweakable cipher over a split domain* $(\mathcal{X}_L \times \mathcal{X}_R)$*. Then for a positive integer* v *and an adversary* $\mathcal{A}$ *attacking the* RRND / RRNDg *security of* $\widetilde{\mathsf{EE}}$ *the corresponding advantage is defined as*

$$\mathbf{Adv}^{\mathrm{RRND/RRNDg}}_{\widetilde{\mathsf{EE}}}(\mathcal{A}, v) = \left|2\Pr\left[\mathsf{RRND}^{\mathcal{A},v}_{\widetilde{\mathsf{EE}}} \,/\, \mathsf{RRNDg}^{\mathcal{A},v}_{\widetilde{\mathsf{EE}}} \Rightarrow 1\right] - 1\right|.$$

We give an analogous statement of the "RPRP to RRND" switching lemma given in [13] for the RPRPg and RPRPd notions. The proof would go along the lines of the proof of the original lemma and therefore we omit it.

Lemma 1. *Let* $\widetilde{\mathsf{EE}}$ *be a tweakable cipher over a split domain* $(\mathcal{X}_L \times \mathcal{X}_R)$ *where* $\mathcal{X}_L \subseteq \{0,1\}^{\geq n}$ *and* $\mathcal{X}_R \subseteq \{0,1\}^{\geq m}$*. Then for a positive integer* v *and an adversary* $\mathcal{A}$ *making* q_{en} *encipher oracle queries and* q_{gu} *guess oracle queries, it holds that*

$$\mathbf{Adv}^{\mathrm{RPRPg}}_{\widetilde{\mathsf{EE}}}(\mathcal{A}, v) \leq \mathbf{Adv}^{\mathrm{RRNDg}}_{\widetilde{\mathsf{EE}}}(\mathcal{A}, v) + \frac{q_{\mathrm{en}}(q_{\mathrm{en}}-1)}{2^{n+m+1}}.$$

B Proof of Theorem 2 (ECE Construction)

Proof. Without loss of generality, we assume that the adversary does not make redundant queries. That is, the adversary does not repeat queries to either of the oracles or make queries that the game will restrict.

We immediately replace the expanding PRFs FE_{K_1} and FE_{K_3} with two random expanding functions fe_1 and fe_3 via the standard argument. Additionally, we replace the compressing PRF FC_{K_2} with a random compression function fc_2, also via the standard argument. These two switches induce the following term in the bound.

$$2\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{FE}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{FC}}(\mathcal{C}). \tag{14}$$

Next, we apply the H-coefficient technique and Theorem 1. The real world is the ECE construction with ideal primitives fe_1, fc_2 and fe_3. In the ideal world, the encipher and decipher oracle evaluate the input on an ideal cipher $\widetilde{\Pi}$ sampled uniformly at random from $\mathsf{IC}(\mathcal{T}, \mathcal{X}_L \times \mathcal{X}_R)$.

The transcript τ contains a list of tuples that correspond to all queries made by the adversary.

$$\tau := ((T^1, X^1_L, X^1_R, Y^1_L, Y^1_R, O^1_1), \ldots, (T^q, X^q_L, X^q_R, Y^q_L, Y^q_R, O^q_1)),$$

where $q = q_{\mathrm{en}} + q_{\mathrm{de}}$. We will use l_i to denote the length of i-th query (i.e., $l_i = |X^i_L| + |X^i_R|$) throughout the proof.

In the real world, variable O^i_1 gets the value $\mathsf{fe}_1(X^i_L)$, where the function fe_1 is the real expanding random function appearing in the first round of the ECE construction. On the other hand, in the ideal world, the function fe_1 is an expanding random function that is sampled by a simulator after the adversary's interaction, after all the values X^i_L's have been fixed.

We note here that with values $(X^i_L, X^i_R, Y^i_L, Y^i_R, O^i_1)$, one can calculate all the other intermediate values that appear during the enciphering or deciphering (e.g., an input to fc_2).

Defining and bounding the bad transcripts. Next, we define what it means for a transcript to be *bad.*

Definition 9. *A transcript* τ *is called bad, if any one of the following conditions hold:*

[B1] *There exist two queries* $(T^i, X^i_L, X^i_R, Y^i_L, Y^i_R, O^i_1)$ *and* $(T^j, X^j_L, X^j_R, Y^j_L, Y^j_R, O^j_1)$ *with* $T^i = T^j$ *and* $l_i = l_j$*, such that* $O^i_1 \oplus X^i_R = O^j_1 \oplus X^j_R$*.*

[B2] *There exist two encipher queries* $(T^i, X^i_L, X^i_R, Y^i_L, Y^i_R, O^i_1)$ *and* $(T^j, X^j_L, X^j_R, Y^j_L, Y^j_R, O^j_1)$ *with* $l_i = l_j$*, such that* $Y^i_L = Y^j_L$*.*

[B3] *There exist a decipher query* $(T^i, X^i_L, X^i_R, Y^i_L, Y^i_R, O^i_1)$ *and an encipher query* $(T^j, X^j_L, X^j_R, Y^j_L, Y^j_R, O^j_1)$ *with* $l_i = l_j$ *and* $i < j$*, such that* $Y^i_L = Y^j_L$*.*

Informally, the conditions [B1]-[B3] serve the purpose of ensuring the outputs of ECE in the real world will always be uniformly random.

Now, let τ be some attainable transcript in the ideal world. We bound the probabilities of above-defined conditions holding true in the ideal world.

[B1] Assuming without loss of generality that $i < j$, we differentiate three possible subcases occurring.

[B1.1] *Both queries are encipher queries.* If $X_L^i = X_L^j$, it must hold that $X_R^i \neq X_R^j$ (otherwise queries would be the same), hence the collision $O_1^i \oplus X_R^i = O_1^j \oplus X_R^j$ cannot happen. If $X_L^i \neq X_L^j$, we have that

$$\Pr\left[O_1^i \oplus X_R^i = O_1^j \oplus X_R^j\right] \leq \frac{1}{2^m},$$

where the probability is taken over random coins of O_1^j.

[B1.2] *Both queries are decipher queries.* If $X_L^i = X_L^j$, then the probability of the equation $O_1^i \oplus X_R^i = O_1^j \oplus X_R^j$ holding true reduces to the probability of X_R^i being equal to X_R^j. That probability is 0 since (X_L^i, X_R^i) and (X_L^j, X_R^j) are sampled according to a permutation and hence they cannot be equal. If $X_L^i \neq X_L^j$, we have that

$$\Pr\left[O_1^i \oplus X_R^i = O_1^j \oplus X_R^j\right] \leq \frac{1}{2^m},$$

where the probability is taken over random coins of O_1^j.

[B1.3] *One query is encipher query, other one is decipher query.* We furthermore now have 4 possibilities here.

- i-th query was encipher and j-th decipher query, and $X_L^i = X_L^j$. Probability of a collision reduces to the probability that $X_R^i = X_R^j$, which is 0 since otherwise the j-th query would be a forbidden forwarded query from En to De.
- i-th query was encipher and j-th decipher query, and $X_L^i \neq X_L^j$. Probability of a collision reduces to the probability that $O_1^j = X_R^i \oplus X_R^j \oplus O_1^i$, which is bounded by $\frac{1}{2^m}$.
- i-th query was decipher and j-th encipher query, and $X_L^i = X_L^j$. Probability of a collision reduces to the probability that $X_R^i = X_R^j$, which is 0 since otherwise j-th query would be a forbidden forwarded query from De to En.
- i-th query was decipher and j-th encipher query, and $X_L^i \neq X_L^j$. Probability of a collision reduces to the probability that $O_1^j = X_R^i \oplus X_R^j \oplus O_1^i$, which is bounded by $\frac{1}{2^m}$.

In all cases, the probability that the equality holds is at most $\frac{1}{2^m}$.

Summing up the probabilities of [B1.1], [B1.2] and [B1.3] over all i, j, such that $i \neq j$, we have that the probability of [B1] being true is bounded by

$$\binom{q}{2}\frac{1}{2^m} \leq \frac{q^2}{2^{m+1}}. \tag{15}$$

[B2] A collision in inputs to fe_3 happens if $Y_L^i = Y_L^j$. Assume without loss of generality that $i < j$. Since both queries are encipher queries, it holds

$$\Pr\left[Y_L^i = Y_L^j\right] \leq \frac{2^{l_j - n} - (j-1)}{2^{l_j} - (j-1)} \leq \frac{2^{l_j - n}}{2^{l_j} - (q_{\text{en}} + q_{\text{de}})} \leq \frac{2^{l_j - n}}{2^{l_j - 1}} = \frac{1}{2^{n-1}},$$

assuming $q_{\text{en}} + q_{\text{de}} \leq 2^{n+m-1} \leq 2^{l_j - 1}$. Summing up over all i, j, we have that the probability of [B2] being true is bounded by

$$\binom{q_{\text{en}}}{2}\frac{1}{2^{n-1}} \leq \frac{q_{\text{en}}^2}{2^n}. \tag{16}$$

[B3] A collision in fe_3 happens if $Y_L^i = Y_L^j$. Since j-th query is an encipher query and $i < j$, through a calculation similar to one in [B2], it holds that

$$\Pr\left[Y_L^i = Y_L^j\right] \leq \frac{1}{2^{n-1}}.$$

Summing up over all $i < j$, we have that the probability of [B3] being true is bounded by

$$\frac{q_{\text{en}} q_{\text{de}}}{2^{n-1}}. \tag{17}$$

Adding up the bounds in (15), (16) and (17), we have that the probability of an attainable transcript τ in the ideal world being bad is bounded by

$$\epsilon_{\text{bad}} = \frac{q^2}{2^{m+1}} + \frac{q_{\text{en}}^2}{2^n} + \frac{q_{\text{en}} q_{\text{de}}}{2^{n-1}}.$$

Bounding the ratio of good transcripts. Fix some good, attainable, transcript τ. Let the transcript contain queries of length $l_1, \dots, l_c$. We split the transcript into disjoint subsets $\tau_1, \dots, \tau_c$, where the set τ_i contains all queries of length l_i. We also introduce an equivalence relation $\sim_T$ across sets τ_i, where two queries from set τ_i are related if and only if they have the same tweak. This equivalence relation partitions the set τ_i into equivalence classes by the tweak T. We denote the number of queries in some j-th equivalence class with $t_{i,j}$. We consequently denote with t_i the number of queries in the set τ_i. It will hold $t_i = t_{i,1} + \cdots + t_{i,\text{cnt}_i}$, where cnt_i is the number of distinct tweaks that appear in the queries in set τ_i. Also, $q = t_1 + \cdots + t_c$.

Ideal world. The interpolation in the ideal world then is

$$\Pr[X_i = \tau] = \prod_{i=1}^{c} \prod_{j=1}^{\text{cnt}_i} (2^{l_i})_{t_{i,j}} \times \mathcal{P}_{O_1},$$

where $\mathcal{P}_{O_1}$ is the interpolation for O_1^i variables appearing in the query transcripts. Since the values for O_1^i are sampled in exactly the same way in both worlds, there is no need to calculate what $\mathcal{P}_{O_1}$ is equal to. The values will cancel themselves out when calculating the ratio of interpolations.

Real world. Because the transcript is good, the input values to fc_2 do not collide (condition [B1]), making the left outputs (Y_L^i if the query was an encipher query, X_L^i if the query was a decipher query) always uniformly random. As for the right output in case of an encipher query, the input to fe_3 will never repeat since conditions [B2] and [B3] exclude that possibility. Therefore, Y_R^i will always be uniformly random. If the query is a decipher query, the left input Y_L^i is guaranteed to be fresh by the RPRPd game restriction. Thus, the output of $\mathsf{fe}_3(Y_L^i)$ will make the right output X_R^i uniformly random. From the above and the fact that the adversary does not make redundant queries, we see that the encipher and decipher algorithms always give uniformly random output in the real world. Therefore, given a transcript query $(T^i, X_L^i, X_R^i, Y_L^i, Y_R^i, O_1^i)$, its input will map to its output, or vice-versa, with probability $\frac{1}{2^{l_i}}$. As for the O_1^i variables interpolation, its value is the same as in the ideal world, $\mathcal{P}_{O_1}$.

Taking all queries from the transcript into account, the interpolation probability in the real world is

$$\Pr[X_r = \tau] = \prod_{i=1}^{c} \left(\frac{1}{2^{l_i}}\right)^{t_i} \times \mathcal{P}_{O_1}.$$

Interpolation ratio. Finally, let us calculate the interpolation ratio for a good transcript τ.

$$\begin{aligned}\frac{\Pr[X_r = \tau]}{\Pr[X_i = \tau]} &= \frac{\prod_{i=1}^{c} \prod_{j=1}^{\text{cnt}_i} (2^{l_i})_{t_{i,j}} \times \mathcal{P}_{O_1}}{\prod_{i=1}^{c} 2^{l_i t_i} \times \mathcal{P}_{O_1}} \geq \prod_{i=1}^{c} \frac{\prod_{j=1}^{\text{cnt}_i} (2^{l_i} - q)^{t_{i,j}}}{2^{l_i t_i}} \\ &= \prod_{i=1}^{c} \left(\frac{2^{l_i} - q}{2^{l_i}}\right)^{t_i} \geq \left(1 - \frac{q}{2^{l_1}}\right)^{t_1} \cdots \left(1 - \frac{q}{2^{l_c}}\right)^{t_c}.\end{aligned}$$

Applying the Weierstrass inequality we obtain that the expression above is greater or equal than

$$1 - q\left(\frac{t_1}{2^{l_1}} + \cdots + \frac{t_c}{2^{l_c}}\right) \geq 1 - q\frac{t_1 + \cdots + t_c}{2^{n+m}} = 1 - \frac{q^2}{2^{n+m}},$$

hence $\epsilon_{\text{ratio}} = \dfrac{q^2}{2^{n+m}}$.

Summing up (14), ϵ_{bad} and ϵ_{ratio} one arrives at the bound from the theorem statement. □

C Proof of Theorem 3 (CEC Construction)

Proof. Without loss of generality, we assume that the adversary does not make redundant queries. That is, the adversary does not repeat queries to either of the oracles or make queries that the game will restrict.

We make use of the RRNDg game and Lemma 1, both of which are given in Appendix A. Applying Lemma 1 and making the switch from the RPRPg game to the RRNDg game induces the term

$$\frac{q_{\text{en}}(q_{\text{en}}-1)}{2^{n+m+1}} \leq \frac{q_{\text{en}}^2}{2^{n+m+1}} \tag{18}$$

in the bound. The adversary now is playing the RRNDg game and let the starting point be the real world of the RRNDg game ($b = 1$).

We replace the compressing PRF FC_{K_1} and FC_{K_3} with two random compressing functions fc_1 and fc_3 via the standard argument. Additionally, we replace the expanding PRF FE_{K_2} with a random expanding function fe_2, also via the standard argument. These two switches yield the following term in the bound.

$$2\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{FC}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{FE}}(\mathcal{C}). \tag{19}$$

Next, we apply the H-coefficient technique and Theorem 1. The real world is the CEC construction with ideal primitives fc_1, fe_2 and fc_3. In the ideal world, the adversary has access to the oracles given in Figure 12. Note that the ideal world corresponds to the ideal world of the RRNDg game. The transcript τ contains a list of tuples that correspond to all queries made by the adversary. There are two types of queries stored in τ:

```
En($T, X_L, X_R$)
if $\widetilde{\mathsf{RR}}(+, T, X_L, X_R) \neq \perp$ then
    $(Y_L, Y_R) \leftarrow \widetilde{\mathsf{RR}}(+, T, X_L, X_R)$
else
    $(Y_L, Y_R) \leftarrow\$ \{0,1\}^{n+|X_R|}$
    $\widetilde{\mathsf{RR}}(+, T, X_L, X_R) \leftarrow (Y_L, Y_R)$
return $(Y_L, Y_R)$
```

```
Gu($T, Y_L, Y_R, \boldsymbol{V}$)
return false
```

Fig. 12: The ideal world for H-coefficient technique application in Theorem 3 (CEC is RPRPg).

1. Queries $(T^i, X_L^i, X_R^i, Y_L^i, Y_R^i, O_1^i)$, corresponding to the queries to En oracle. The variable O_1^i takes in the real world the following value
$$O_1^i := \mathsf{fc}_1(X_R^i),$$
where the function fc_1 is the compressing random function present in the first round of the construction. In the ideal world, O_1^i is set by the simulator that we define later.
2. Queries $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i, o^i, X_L^i, X_R^i, O_1^i)$, corresponding to the queries to Gu oracle. The variable o^i corresponds to the answer of the guess oracle, and for a query from an attainable transcript, its value will always be **false**. In the real world, the pair (X_L^i, X_R^i) corresponds to the real output that would be evaluated using $\mathsf{fc}_1, \mathsf{fe}_2$ and fc_3 on the input (T^i, Y_L^i, Y_R^i). In the ideal world, the pair (X_L^i, X_R^i) is sampled by the simulator that we define shortly.

The simulator S runs in the ideal world after the adversary has finished its interaction with the oracles, and executes the following steps (in the given order).

i. It iterates through all En queries and for each query $(T^i, X_L^i, X_R^i, Y_L^i, Y_R^i)$ it sets $O_1^i := \mathsf{fc}_1(X_R^i)$, where the function fc_1 is a random compressing function it samples.

ii. It iterates through all Gu oracle queries and for each query $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i)$ first determines if the triple (T^i, Y_L^i, Y_R^i) is fresh.

- (T^i, Y_L^i, Y_R^i) *is fresh.* Set $(X_L^i, X_R^i) \xleftarrow{\$} \{0,1\}^{n+|Y_R|}$. For defining O_1^i we first need to know if Y_R^i is *new*. We call Y_R^i new if there is no earlier En or Gu query in τ that contains Y_R^i.

  If Y_R^i is new, then $O_1^i := \mathsf{fc}_1(X_R^i)$.

  Otherwise Y_R^i is not new, and we can fix j-th En or Gu query, with $j < i$, such that $Y_R^j = Y_R^i$. Then $O_1^i := O_1^j \oplus Y_L^j \oplus X_L^j \oplus X_L^i \oplus Y_L^i$. Note that the term $O_1^j \oplus Y_L^j \oplus X_L^j$ would be an implicit output of the function $\mathsf{fc}_3(Y_R^i)$ if we were in the real world.
- (T^i, Y_L^i, Y_R^i) *is not fresh.* The simulator takes the values (X_L^j, X_R^j, O_1^j) from some previous j-th guess query with the same (T^i, Y_L^i, Y_R^i) and sets

$$(X_L^i, X_R^i, O_1^i) := (X_L^j, X_R^j, O_1^j).$$

We will use l_i to denote the length of i-th query (i.e., $l_i = |X_L^i| + |X_R^i|$) throughout the proof.

Defining and bounding the bad transcripts. We now define what it means for an attainable transcript to be *bad*. The intuition for the following bad transcript conditions is as follows. The [B1], [B2], [B4] and [B5] conditions ensure output of CEC encipher algorithm in the real world will always be uniformly random. The [B2], [B3], [B6] and [B7] conditions ensure the (X_L, X_R) from the guess oracle query transcript in the real world will always be indistinguishable from their counterparts in the ideal world. The condition [B8] excludes guess oracle queries that would be deemed successful in the real world.

Definition 10. *A transcript τ is called bad, if any one of the following conditions hold:*

[B1] *There exist two encipher queries $(T^i, X_L^i, X_R^i, Y_L^i, Y_R^i, O_1^i)$ and $(T^j, X_L^j, X_R^j, Y_L^j, Y_R^j, O_1^j)$ with $T^i = T^j$ and $l_i = l_j$, such that $O_1^i \oplus X_L^i = O_1^j \oplus X_L^j$.*

[B2] *There exist an encipher query $(T^i, X_L^i, X_R^i, Y_L^i, Y_R^i, O_1^i)$ and a guess query $(T^j, Y_L^j, Y_R^j, \boldsymbol{V}^j, \mathbf{false}, X_L^j, X_R^j, O_1^j)$ with $l_i = l_j$, such that $X_L^i \oplus O_1^i = X_L^j \oplus O_1^j$.*

[B3] *There exist two guess queries $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i, \mathbf{false}, X_L^i, X_R^i, O_1^i)$ and $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i, \mathbf{false}, X_L^j, X_R^j, O_1^j)$ with $T^i = T^j$ and $(Y_L^i, Y_R^i) \neq (Y_L^j, Y_R^j)$ with $l_i = l_j$, such that $O_1^i \oplus X_L^i = O_1^j \oplus X_L^j$.*

[B4] *There exist two encipher queries $(T^i, X_L^i, X_R^i, Y_L^i, Y_R^i, O_1^i)$ and $(T^j, X_L^j, X_R^j, Y_L^j, Y_R^j, O_1^j)$ with $l_i = l_j$, such that $Y_R^i = Y_R^j$.*

[B5] *There exist a guess query $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i, o^i, X_L^i, X_R^i, O_1^i)$ and an encipher query $(T^j, X_L^j, X_R^j, Y_L^j, Y_R^j, O_1^j)$ with $l_i = l_j, i < j$ and $T^i = T^j$, such that $Y_R^i = Y_R^j$.*

[B6] *There exist an encipher query $(T^i, X_L^i, X_R^i, Y_L^i, Y_R^i, O_1^i)$ and a guess query $(T^j, Y_L^j, Y_R^j, \boldsymbol{V}^j, \mathbf{false}, X_L^j, X_R^j, O_1^j)$ with $l_i = l_j$, such that $X_R^i = X_R^j$.*

[B7] *There exist two guess queries $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i, \mathbf{false}, X_L^i, X_R^i, O_1^i)$ and $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i, \mathbf{false}, X_L^j, X_R^j, O_1^j)$ with $l_i = l_j$ and $(T^i, Y_L^i, Y_R^i) \neq (T^j, Y_L^j, Y_R^j)$, such that $X_R^i = X_R^j$.*

[B8] *There exists a guess query $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i, \mathbf{false}, X_L^i, X_R^i, O_1^i)$ such that $X_L^i \in \boldsymbol{V}^i$.*

Now, let τ be some attainable transcript in the ideal world. We bound the probabilities of above-defined conditions holding true in the ideal world.

[B1] Assume without loss of generality that $i < j$. If $X_R^i = X_R^j$, then we know it holds $X_L^i \neq X_L^j$ since otherwise j-th query would be a repeated query. In case of $X_R^i \neq X_R^j$ it holds

$$\Pr\left[O_1^i \oplus X_L^i = O_1^j \oplus X_L^j\right] = \frac{1}{2^n},$$

where the probability is taken over O_1^j. Summing over all q_{en} encipher queries, the probability of [B1] being true is at most

$$\binom{q_{\text{en}}}{2} \frac{1}{2^m} \leq \frac{q_{\text{en}}^2}{2^{n+1}}. \tag{20}$$

[B2] We denote the equation $O_1^i \oplus X_L^i = O_1^j \oplus X_L^j$ with E and expand the bad condition using the law of total probability as

$$\begin{aligned}\Pr[E] &= \Pr\left[E \,\middle|\, X_R^i = X_R^j\right] \Pr\left[X_R^i = X_R^j\right] + \Pr\left[E \,\middle|\, X_R^i \neq X_R^j\right] \Pr\left[X_R^i \neq X_R^j\right] \\ &\leq \Pr\left[X_R^i = X_R^j\right] + \Pr\left[E \,\middle|\, X_R^i \neq X_R^j\right] \leq \frac{1}{2^m} + \frac{1}{2^n},\end{aligned}$$

where the final inequality follows from the fact (X_L^j, X_R^j) is uniformly sampled after (X_L^i, X_R^i) have been fixed. Summing over all i, j, the probability of [B2] being true is at most

$$\frac{q_{\text{en}}q_{\text{gu}}}{2^m} + \frac{q_{\text{en}}q_{\text{gu}}}{2^n}. \tag{21}$$

[B3] It holds

$$\Pr\left[O_1^i \oplus X_L^i = O_1^j \oplus X_L^j\right] = \frac{1}{2^n}$$

since X_L^i and X_L^j are uniformly sampled (or were uniformly sampled, in case the guess queries i and j are repeated ones). Summing over all q_{gu} guess queries, the probability of [B3] being true is at most

$$\binom{q_{\text{gu}}}{2}\frac{1}{2^n} \leq \frac{q_{\text{gu}}^2}{2^{n+1}}. \tag{22}$$

[B4] Assume without loss of generality that $i < j$. It holds

$$\Pr\left[Y_L^i = Y_R^i\right] \leq \frac{1}{2^m},$$

where the probability is taken over Y_R^i. Summing over all q_{en} encipher queries, the probability of [B4] being true is bounded by

$$\binom{q_{\text{en}}}{2}\frac{1}{2^m} \leq \frac{q_{\text{en}}^2}{2^{m+1}}. \tag{23}$$

[B5] Calculated over the probability of Y_R^j, it holds

$$\Pr\left[Y_L^i = Y_R^i\right] \leq \frac{1}{2^m}.$$

Summing over all $i < j$, the probability of [B5] being true is bounded by

$$\frac{q_{\text{en}}q_{\text{gu}}}{2^m}. \tag{24}$$

[B6] It holds that

$$\Pr\left[X_R^i = X_R^j\right] \leq \frac{1}{2^m},$$

where the inequality follows from the fact X_R^j is uniformly sampled after X_R^i has been fixed. Summing over all i, j, the probability of [B6] being true is at most

$$\frac{q_{\text{en}}q_{\text{gu}}}{2^m}. \tag{25}$$

[B7] It holds that

$$\Pr\left[X_R^i = X_R^j\right] \leq \frac{1}{2^m}$$

since X_R^i and X_R^j are uniformly sampled (or were uniformly sampled, in case the guess queries i and j are repeated ones). Summing over all i, j, the probability of [B7] being true is at most

$$\binom{q_{\text{gu}}}{2}\frac{1}{2^m} \leq \frac{q_{\text{gu}}^2}{2^{m+1}}. \tag{26}$$

[B8] For a triple (T^i, Y_L^i, Y_R^i), the variable X_L^i is sampled after all guessed sets have been fixed. For a concrete guess set $\boldsymbol{V}^i$, it then holds

$$\Pr\left[X_L^i \in \boldsymbol{V}^i\right] \leq \frac{v}{2^n},$$

v being the maximum cardinality of $\boldsymbol{V}^i$ the adversary is allowed to query. Summing over all q_{gu} guess queries, the probability of [B8] being true is bounded by

$$\frac{q_{\text{gu}}v}{2^n}. \tag{27}$$

Adding up the bounds in (20) – (27) we have that the probability of an attainable transcript τ in the ideal world being bad is bounded by

$$\epsilon_{\text{bad}} \leq \frac{q_{\text{en}}^2 + q_{\text{gu}}^2}{2^{n+1}} + \frac{q_{\text{en}}q_{\text{gu}} + q_{\text{gu}}v}{2^n} + \frac{q_{\text{en}}^2 + q_{\text{gu}}^2}{2^{m+1}} + \frac{3q_{\text{en}}q_{\text{gu}}}{2^m}.$$

Bounding the ratio of good transcripts. Fix some good, attainable, transcript τ. We split the transcript into two disjoint subsets τ_{en} and τ_{gu} that contain encipher and guess oracle query transcripts, respectively. The set τ_{en} contains queries of length $l_1, \dots, l_c$, and let t_i denote the number of encipher queries of length l_i. It holds $q_{\text{en}} = t_1 + \cdots + t_c$. The set τ_{gu} contains guess queries with ugu unique triples (T^i, Y_L^i, Y_R^i) being queried throughout the interaction. Finally, we denote with uyr the number of fresh guess triples (T^i, Y_L^i, Y_R^i) where Y_R^i is new.

Ideal world. We first take a look at encipher queries transcripts. For some fixed query i, the input (T^i, X_L^i, X_R^i) will map to the output (Y_L^i, Y_R^i) with probability $\frac{1}{2^{l_i}}$, since the encipher oracle in the ideal world returns uniformly random bits of the same length as the input (X_L^i, X_R^i). With $\mathcal{P}_{O_1,\text{en}}$ we denote the interpolation probability for O_1^i variables appearing in the encipher oracle queries transcripts. Since these variables are sampled in exactly the same way in both worlds, there is no need to calculate what $\mathcal{P}_{O_1,\text{en}}$ is equal to.

For guess oracle queries, let us fix some i-th query $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i, \mathbf{false}, X_L^i, X_R^i, O_1^i)$, where the triple (T^i, Y_L^i, Y_R^i) is being queried for the first time. The interpolation probability for X_R^i is clearly $\frac{1}{2^{l_i - n}}$. Similarly, for X_L^i it is $\frac{1}{2^n}$. Then for the interpolation probability of O_1^i we differentiate two cases. The first one is where Y_R^i is new. In this case, O_1^i is sampled and X_R^i, which is always unique by conditions [B6] and [B7], "maps" to O_1^i with probability $\frac{1}{2^n}$. The other case is where Y_R^i is not new. Now O_1^i is fixed and it appears in the transcript with probability 1. The interpolation in the ideal world then is

$$\Pr[X_i = \tau] = \prod_{i=1}^{c} \frac{1}{2^{l_i t_i}} \times \mathcal{P}_{O_1,\text{en}} \times \frac{1}{2^{(l_i - n)\cdot \mathsf{ugu}}} \times \frac{1}{2^{n \cdot \mathsf{ugu}}} \times \frac{1}{2^{n \cdot \mathsf{uyr}}}.$$

Real world. We again first look at encipher queries transcripts. Because the transcript is good, the input values to fe_2 do not collide (conditions [B1] and [B2]), making the right output Y_R^i always uniformly random. As for the left output Y_L^i, the input to fe_3 will never repeat since conditions [B4] and [B5] exclude that possibility. Therefore, Y_L^i will always be uniformly random. From the above and the fact that the adversary does not make redundant queries, we see that the encipher algorithm always gives uniformly random outputs in the real world. Therefore, given a transcript query $(T^i, X_L^i, X_R^i, Y_L^i, Y_R^i, O_1^i)$, its input will map to its output with probability $\frac{1}{2^{l_i}}$. As for the O_1^i variables interpolation in the encipher queries, its value is the same as in the ideal world, $\mathcal{P}_{O_1,\text{en}}$.

As for the guess oracle queries, let us analyze a query $(T^i, Y_L^i, Y_R^i, \boldsymbol{V}^i, \mathbf{false}, X_L^i, X_R^i, O_1^i)$, where the triple (T^i, Y_L^i, Y_R^i) is being queried for the first time. Again it holds the input values to fe_2 do not collide (conditions [B2] and [B3]), making the right output X_R^i uniformly random. For the interpolation probability of O_1^i we know its $\frac{1}{2^n}$ since the conditions [B6] and [B7] ensure the X_R^i that "maps" to it never repeats. As for the left output X_L^i, we know it is calculated as $X_L^i := Y_L^i \oplus \mathsf{fc}_3(Y_R^i) \oplus O_1^i$. At this point we also know O_1^i has already been sampled, that is, it is fixed. Finally, we now differentiate two cases. The first one is where Y_R^i is new, making the interpolation probability for X_L^i be $\frac{1}{2^n}$, by the implicit sampling of fc_3. In the other case, where Y_R^i is not new, everything that determines X_L^i is already fixed, so its interpolation probability is 1. Taking all queries from the transcript into account, the interpolation probability in the real world is

$$\Pr[X_r = \tau] = \prod_{i=1}^{c} \frac{1}{2^{l_i t_i}} \times \mathcal{P}_{O_1,\text{en}} \times \frac{1}{2^{(l_i - n)\cdot \mathsf{ugu}}} \times \frac{1}{2^{n \cdot \mathsf{uyr}}} \times \frac{1}{2^{n \cdot \mathsf{ugu}}}$$

Interpolation ratio. At last, the interpolation ratio for a good transcript τ is

$$\frac{\Pr[X_r = \tau]}{\Pr[X_i = \tau]} = \frac{2^{n \cdot \mathsf{ugu}} \times 2^{n \cdot \mathsf{uyr}}}{2^{n \cdot \mathsf{uyr}} \times 2^{n \cdot \mathsf{ugu}}} = 1,$$

hence $\epsilon_{\text{ratio}} = 0$.

Summing up (18), (19), ϵ_{bad} and ϵ_{ratio} one arrives at the bound from the theorem statement. □

D Supplementary Material Section 5

Attack Against Insecure Variant of `HEC`. The adversary $\mathcal{A}$, playing the RPRPd game, works as follows:

1. Query $(Y_L, Y_R) \leftarrow \textsc{En}(X_L, X_R)$, for some X_L, X_R and with X_R being one block long.
   Initial value for the CTR_K is $IV = X_L \oplus \mathsf{H}_h(T, X_R) \oplus Y_L$.
2. Query $(X'_L, X'_R) \leftarrow \textsc{De}(X_R \oplus Y_R, Y_R)$
   We first note that this query will be valid, meaning $X_R \oplus Y_R \neq Y_L$ with high probability. The deciphering $\mathsf{E}_K^{-1}(X_R \oplus Y_R)$ gives $IV \oplus 1$ as an output, thus it holds, $X'_L = IV \oplus 1 \oplus \mathsf{H}_h(T, X'_R)$.
3. Calculate the mask of $\mathsf{H}_h(T, X_R) \oplus \mathsf{H}_h(T, X'_R) = X_L \oplus Y_L \oplus IV \oplus X'_L \oplus IV \oplus 1 = X_L \oplus Y_L \oplus X'_L \oplus 1$.
   The adversary is now left with one equation with one unknown, namely the hash function key h, so it can then extract the hash key h.

Useful Lemmas. We present in the following two lemmas that we use for interpolation ratio calculation in the proof of Theorem 4.

Lemma 2. *For positive integers $a, b_1, \ldots, b_n$, such that $a \geq b_1 + \cdots + b_n$, it holds that*

$$(a)_{b_1} \cdots (a)_{b_n} \geq (a)_{b_1+\cdots+b_n}.$$

Proof. Let us prove the lemma by induction on n. Obviously, for $n = 1$, $(a)_{b_1} \geq (a)_{b_1}$. Next, suppose the inequality holds for n and let us show the inequality then holds true for $n+1$ as well. By assumption it follows

$$(a)_{b_1} \cdots (a)_{b_n} (a)_{b_{n+1}} \geq (a)_{b_1+\cdots+b_n} (a)_{b_{n+1}},$$

which is furthermore greater or equal than $(a)_{b_1+\cdots+b_{n+1}}$. □

The next lemma also appears in [14]. We reiterate it here for completeness.

Lemma 3. *For positive integers $n, k_1, \ldots, k_c, t_1, \ldots, t_c$, such that $\sum_{i=1}^{c} t_i k_i \leq 2^n$, it holds that*

$$\prod_{i=1}^{c} (2^{k_i n})_{t_i} \geq (2^n)_{t_1 k_1 + \cdots + t_c k_c}.$$