ZCZ – Achieving n-bit SPRP Security with a Minimal Number of Tweakable-block-cipher Calls

Ritam Bhaumik[1], Eik List[2], and Mridul Nandi[1⋆]

[1] Indian Statistical Institute, Kolkata, India
[2] Bauhaus-Universität Weimar, Weimar, Germany
bhaumik.ritam@gmail.com, eik.list@uni-weimar.de, mridul.nandi@gmail.com

Abstract. Strong Pseudo-random Permutations (SPRPs) are important for various applications. In general, it is desirable to base an SPRP on a single-keyed primitive for minimizing the implementation costs. For constructions built on classical block ciphers, Nandi showed at ASIACRYPT'15 that at least two calls to the primitive per processed message block are required for SPRP security, assuming that all further operations are linear. The ongoing trend of using tweakable block ciphers as primitive has already led to MACs or encryption modes with high security and efficiency properties. Thus, three interesting research questions are hovering in the domain of SPRPs: (1) if and to which extent the bound of two calls per block can be reduced with a tweakable block cipher, (2) how concrete constructions could be realized, and (3) whether full n-bit security is achievable from primitives with n-bit state size.

The present work addresses all three questions. Inspired by Iwata et al.'s ZHash proposal at CRYPTO'17, we propose the ZCZ (ZHash-Counter-ZHash) construction, a single-key variable-input-length SPRP based on a single tweakable block cipher whose tweak length is at least its state size. ZCZ possesses close to optimal properties with regards to both performance and security: not only does it require only asymptotically $3\ell/2$ calls to the primitive for ℓ-block messages, but we also show that this figure is close to the minimum by an PRP distinguishing attack on any construction with tweak size of $\tau = n$ bits and fewer than $(3\ell - 1)/2$ calls to the same primitive. Moreover, it provides optimal n-bit security for a primitive with n-bit state and tweak size.

Keywords: Symmetric-key cryptography · provable security · variable-input-length SPRP · tweakable block cipher · encryption

1 Introduction

SPRPs. Strong Pseudo-Random Permutations (SPRPs) (or wide-block ciphers), are important symmetric-key cryptographic schemes for protecting the privacy

⋆ Mridul Nandi has been supported in his research by the Wisekey project at the R. C. Bose Centre of Cryptology and Security, Indian Statistical Institute, Kolkata.

of variable-length messages. Their tweakable variants (STPRPs) are useful to build strong authenticated encryption [25, 53] or onion AE [49]. During the previous two decades, the symmetric-key community proposed a considerable corpus of SPRPs. From a high-level point of view, the existing constructions could be categorized into (1) Generalized Feistel networks, (2) Encrypt-Mix-Encrypt, (3) Hash-ECB-Hash, (4) Hash-Counter-Hash, and (5) miscellaneous designs. A brief review of existing works can be found in Appendix A.

Optimization Goals. The primary goals for optimizations in cryptographic schemes are, in general, low implementation costs, high provable security guarantees, and high performance. For the first criterion, it is desirable to construct higher-level schemes from a single well-analyzed primitive without large internal state and with a single key.

High security is essential in many domains that have to process large data stores without the ability of frequent re-keying. In most constructions, however, it only comes at the cost of decreased performance. Unsurprisingly, the challenges of combining high security guarantees with high performance have been identified as among the hot topics of symmetric-key cryptography at the ESC 2017 workshop [7].

Often, high security is associcated with security *beyond the birthday bound*. In the areas of authentication (e.g., [32, 55, 56]), encryption as well as authenticated encryption (e.g., [26, 27, 47]), beyond-birthday security has undergone a long line of research. In the area of SPRPs, however, the security of the vast majority of existing constructions is still limited to the birthday bound of $n/2$ bits, where n is the state size of the underlying primitive. So, the privacy guarantees are lost if $q \simeq 2^{n/2}$ message blocks have been encrypted under the same key. Assuming the AES as primitive, this would imply that significantly fewer than 2^{64} blocks could safely be encrypted under a single key.

Security of SPRPs: State of the Art. Among the earlier proposals, the LargeBlock1 and LargeBlock2 constructions by Minematsu and Iwata [38] as well as TCT_2 by Shrimpton and Terashima [53] are exceptional for their security guarantees. The LargeBlock designs can achieve optimal n-bit security, whereas TCT_2 is limited by $2n/3$ bits. Both share similarities to the Ψ_2 and Ψ_3 constructions from Coron et al. [16], which use two and three calls to a tweakable block cipher. Both LargeBlock2 and TCT_2 possess a sandwich structure, where an encryption layer is wrapped by two layers of hashing. In the former, the encryption layer is an application of Ψ_2 in ECB-mode; the hashing layers employs two calls to a polynomial hash of $2(\ell - 1)$multiplications each. TCT_2 can be seen as an unbalanced version of Ψ_3, where also $2(\ell - 1)$ of ℓ input blocks are hashed in each hashing layer. Both constructions are remarkable for their time. To be comparably efficient, however, they required two primitives, a block cipher and a universal hash function.

A different direction is followed by more recent encryption schemes: Mr. Monster Burrito [5] and HHFHFH [4] are both four-round unbalanced Feis-

tel networks built on large-state primitives, which were coined as heavyweight ciphers. Instead of providing beyond-birthday security, they possess large security margins due to a larger birthday bound of their internal primitives. However, it is exactly the large state size that limits their efficiency.

The only approach we are aware of that almost combines both security and performance desiderata is Simpira (v2) [19], a family of Feistel-like constructions built upon the AES round function. Its authors claim 128-bit security and high performance on current processors with support for AES native instructions. However, Simpira's security claim stems purely from heuristics, which will demand years of further intensive cryptanalysis to build trust into it.

Tweakable Block Ciphers. One established approach for achieving higher security without sacrificing performance significantly is to use a *tweakable block cipher* (TBC) [31] as underlying primitive. At the core, tweakable block ciphers employ an additional public input called tweak, which allows to efficiently separate the domains of different calls to the primitive. This fact can reduce the impact of internal collisions on the security of the scheme built around them. For message authentication codes (MACs), a series of recent works pushed the security bounds further [15, 28, 40], but a similar trend is also observable in the domain of encryption modes and authenticated encryption schemes [27, 30, 36, 47, 48]. This approach has also been used earlier for SPRPs [16, 35, 37, 38, 53] – those proposals, however, originate from at least half a decade ago where TBCs still used to be constructed in cumbersome fashion from classical block ciphers, which resulted in quite inefficient designs. Nowadays, we have the option of using efficient dedicated TBCs, such as Deoxys-BC, Joltik-BC [29], or Skinny [2].

The application of TBCs can also boost the efficiency of constructions, as has been demonstrated recently for MACs. At CRYPTO'17, Iwata et al. [28] introduced ZMAC, a TBC-based parallelizable, single-key single-primitive MAC whose internal hash function ZHash processed the message in both the tweak and plaintext simultaneously. The additional message bits per primitive call render ZMAC more efficient than previous MACs and invite adoption of the approach to other domains.

Open Research Questions. When abstracting away the details of the used primitive – as is usual when proving the security of a scheme – the number of calls to it per input block becomes the main efficiency metric. From Encrypt-Mix-Encrypt-based constructions, it is well-known that the bound is at most two calls per block (plus some minor overhead), assuming all further operations are linear. Thus, it is an interesting question if SPRPs can be built from fewer calls to a single-keyed primitive. Moreover, a strongly related question is that for the minimal number of calls necessary for SPRP security. From the theoretical perspective, Nandi [43] showed that 2ℓ calls for messages of ℓ blocks are necessary for SPRP security for constructions with a classical block cipher. Though, it seems as though this bound is reducible by using a TBC instead as the underlying primitive. For Hash-Counter-Hash-based constructions, the most efficient

Table 1: Asymptotic # of primitive calls for previous SPRP paradigms. We assume, their hash functions and encryption layers use a single-keyed (tweakable) block cipher with n-bit state and τ-bit tweak size to encrypt an ℓ-block message of σ bits in total. We assume the hashing layers use ZHash (as the most efficient blockcipher-based hash function we are aware of).

	#Block-cipher calls			
Paradigm	Top	Middle	Bottom	Total (asympt.)
LargeBlock2	$2\lceil(\ell-1)/2\rceil$	ℓ	$2\lceil(\ell-1)/2\rceil$	$4\lceil(\ell-1)/2\rceil+\ell$
TCT_2	$2\lceil(\ell-1)/2\rceil$	$2(\ell-1)$	$2\lceil(\ell-1)/2\rceil$	$4\lceil(\ell-1)/2\rceil+2\ell$
Encrypt-Mix-Encrypt	ℓ	$\lceil\ell/n\rceil$	ℓ	$2\ell+\lceil\ell/n\rceil$
Hash-ECB-Hash	ℓ	ℓ	ℓ	3ℓ
Hash-Counter-Hash	$\lceil\sigma/(n+\tau)\rceil$	ℓ	$\lceil\sigma/(n+\tau)\rceil$	$\ell+2\lceil\sigma/(n+\tau)\rceil$
ZCZ	$\ell/2$	$\ell/2+\lceil\ell/2n\rceil$	$\ell/2$	$3\ell/2+\lceil\ell/2n\rceil$

(T)BC-based hash function we are aware of is ZHash. For a TBC with n-bit state and τ-bit tweak length, it would yield a construction of about $\ell+2\lceil\sigma/(n+\tau)\rceil$ calls for messages of σ bits. For dedicated TBCs, such as Deoxys-BC-128-384 or Skinny-128-384, this figure still implies that approximately $5\ell/3$ calls are necessary. Regarding the other design principles, it is unclear if similar results are applicable to constructions based on the Encrypt-Mix-Encrypt or Hash-ECB-Hash paradigms. The latter demands an invertible hash function, for which we are unaware of how ZHash could be used. Therefore, we estimate that Hash-ECB-Hash constructions would need about ℓ primitive calls in each hashing layer, plus ℓ calls in the encryption layer.

If one would instantiate LargeBlock2 with ZHash in place of polynomial multiplications, one would have $2\lceil(\ell-1)/2\rceil$ calls in each hashing layer, plus ℓ calls in the middle. So, this would yield 3ℓ calls again. TCT_2 could use a ZHash layer each for both top and bottom hashing layer. While further modifications could make it more efficient, its proposal employed $2\ell-2$ calls in the middle. We compare the approaches in Table 1. Altogether, there remain three interesting research questions: (1) to which extent can the number of primitive calls be reduced when employing a tweakable block cipher, (2) how can a concrete construction be realized, and (3) can it be built with high provable security guarantees.

Contribution. This work tries to answer all three questions above: for the theoretical interest, (1) we show that a number of 1.5ℓ primitive calls per message block is close to minimal by a generic distinguisher on any construction that employs fewer than $(3\ell-1)/2$ calls to a single-keyed primitive per message block, where all further operations are linear. To fulfill the practitioner's interest, we propose (2) ZCZ (ZHash-Counter-ZHash), an almost fully parallelizable variable-input-length SPRP based on a single-keyed TBC with n-bit state size and n-bit tweak size. ZCZ matches approximately the optimal number of 1.5ℓ

calls to the primitive for an ℓ-block message, plus a small overhead. Finally, we show (3) that it achieves optimal n-bit security, i.e., the SPRP advantage of any adversary that asks at most q queries of σ blocks in total is in $O(\sigma^2/2^{2n})$.

For a fair comparison, we note that instantiations of Hash-Counter-Hash with ZHash and a TBC with very large tweak size of $\tau = 3n$, the number of calls to the primitive could become equal to that of ZCZ. However, such primitives would introduce a significant slowdown, be it due to the requirements of more rounds in a TWEAKEY-like cipher, or due to the need of calling an additional universal hash function for compressing the tweak. Concerning practical tweak sizes $\tau < 3n$, the number of calls is significantly lower for our construction.

Figure 1 provides a high-level overview on ZCZ. A given message M is split an input message into (M_L, M_R), where M_R consists of one $2n$-bit di-block, and M_L of the remaining di-blocks; the major part M_L is then processed by a variant of ZHash, that is denoted ZHash* here. It differs from ZHash in two aspects: ZHash* omits the XOR of the TBC output to the tweak input blocks. More prominently, ZHash* does not compress the input to two hash values, but is a permutation over $(n + \tau)^*$. So, the top layer returns the TBC outputs and the tweaks. $\widetilde{V}_1$ and $\widetilde{V}_2$ represent tweakable permutations. Internally, they can use the same primitive as also for ZHash*, and the tweakable variant of Counter mode, CTR*. H symbolizes an error-correcting code that sums up the inputs to $2n$ bits.

This high-level view allows to give a rationale for a dedicated analysis. A straight-forward use of a rate-1 counter mode would allow to apply a standard generic proof as for HCTR. At the same time, such an approach would yield 2ℓ calls to the primitive alone in the counter mode. In combination with ZHash*, this approach would need 4ℓ calls to the primitive for messages of 2ℓ blocks. ZCZ considers a special variant of counter mode that uses only ℓ blocks of entropy to mask 2ℓ blocks, similar as has been used in AEZ from version 2 [25]. However, this counter mode disallows to simply adopt the analysis from HCTR-like constructions when the goal is showing n-bit security. Therefore, a dedicated analysis is needed, which is a major contribution of the present work.

Yet Another Encryption Scheme? In spite of the motivations above, it may appear that our proposal is after all yet another encryption scheme, and with hundreds of encryption schemes already being present in the canon, it is difficult get excited about another one; notwithstanding small improvements in performance and security. We beg to differ on this point – primarily for two reasons: (1) very few of the existing encryption schemes provide security of n bits when using a primitive with an n-bit output—most in fact are only secure up to the birthday bound; as such, this is no small improvement in terms of security, but rather a leap; since there is a lot of current interest in the (still) small group of constructions that achieve this security bound, we believe our encryption scheme is an exciting addition to this group; (2) even more significant is the way we use the randomness generated by a tweakable blockcipher—most previous approaches were based on generic replacements of two or more blockcipher calls

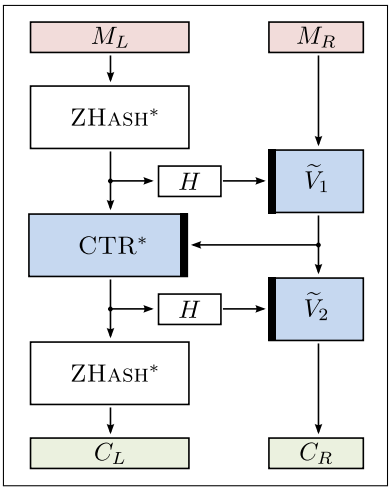


Fig. 1: High-level view on our proposal of ZCZ. Note that the ZHash* module used in this schematic representation is a slightly modified variation of the original ZHash—for instance, in the upper layer, it computes ZHash, passes the hash output to the right, and also passes some internal value downwards, to be masked in the counter mode. The explicit construction diagrams can be found in Figure 3 and Figure 4.

by a single call to a tweakable block cipher; the approach we use is new and not a corollary of any previous work; given its efficiency, we believe it can lead to exciting new directions in research on tweakable-blockcipher modes.

Outline. The remainder is structured as follows: first, Section 2 briefly summarizes the necessary preliminaries. Given a primitive with an effective tweak size[3] τ equals the state size, $\tau = n$ Section 3 will illustrate that every PRP with fewer than $3\ell - 1$ primitive calls for 2ℓ-block messages is insecure, which was the core motivation for our search for constructions with about 1.5ℓ calls. Subsequently, Section 4 defines our basic construction, which is first described for messages whose length is a positive multiple of $2n$ bits. Thereupon, Section 5 extends our definition to messages of more general lengths. Section 6 provides the core details of our security analysis.

While the space limitations prevented them from being included in the main matter of this work, we provide in Appendix D further insights on the starting point of our research. Therein, we also discuss attacks on insecure preliminary variants that motivated our studies towards the final design of ZCZ. Since we can imagine that both designers and cryptanalysts may benefit from those insights, we plan to publish them in a full version alongside this work.

[3] By effective tweak size, we mean the usable tweak domain without bits that are used for other purposes such as domain separation.

2 Preliminaries

General Notation. We use lowercase letters x for indices and integers, uppercase letters X, Y for binary strings and functions, and calligraphic uppercase letters $\mathcal{X}, \mathcal{Y}$ for sets. We denote the concatenation of binary strings X and Y by $X \,\|\, Y$ and the result of their bitwise XOR by $X \oplus Y$. For tuples of bit strings $(X_1, \ldots, X_x)$, $(Y_1, \ldots, Y_x)$ of equal domain, we denote by $(X_1, \ldots, X_x) \oplus (Y_1, \ldots, Y_x)$ the element-wise XOR, i.e., $(X_1 \oplus Y_1, \ldots, X_x \oplus Y_x)$. We mostly treat bit strings as representations of elements in the finite field $\mathbb{F}_{2^n}$ with a fixed primitive polynomial $p(\mathtt{x})$, where addition is equivalent to bitwise XOR $\oplus$. So, we write addition of bit strings to mean addition in the field. We indicate the length of X in bits by $|X|$, and write X_i for the i-th block. Moreover, we denote by $X \twoheadleftarrow \mathcal{X}$ that X is chosen independently uniformly at random from the set $\mathcal{X}$. We define three sets of particular interest: $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$ be the set of all functions $F : \mathcal{X} \to \mathcal{Y}$, $\mathsf{Perm}(\mathcal{X})$ the set of all permutations over $\mathcal{X}$, and $\widetilde{\mathsf{Perm}}(\mathcal{T}, \mathcal{X})$ for the set of tweaked permutations over $\mathcal{X}$ with associated tweak space $\mathcal{T}$.

$(X_1, \ldots, X_x) \xleftarrow{n} X$ denotes that X is split into the minimal number of n-bit blocks possible i.e., $X_1 \,\|\, \ldots \,\|\, X_x = X$, and $|X_i| = n$ for $1 \le i \le x-1$, and $|X_x| \le n$. So, when $|X| > 0$, then $|X_x| > 0$. If $|X| = 0$, $Y \xleftarrow{x} X$ sets Y to the empty string. We denote by $\langle X \rangle_n$ an encoding of an integer $X \in \mathbb{Z}_n$ as an n-bit string. For two sets $\mathcal{X}$ and $\mathcal{Y}$, a uniform random function $\rho : \mathcal{X} \to \mathcal{Y}$ maps inputs $X \in \mathcal{X}$ independently from other inputs and uniformly at random to outputs $Y \in \mathcal{Y}$. For an event E, we denote by $\Pr[E]$ the probability of E; ε is the empty string. For a given set $\mathcal{X}$ and integer x, we define $\mathcal{X}^{\le x} = \bigcup_{i=1}^{x} \mathcal{X}^i$ and $\mathcal{X}^+ = \bigcup_{j=1}^{\infty} \mathcal{X}^j$. For two integers n, k with $n \ge k \ge 1$, we denote the falling factorial as $(n)_k = \prod_{i=0}^{k-1} (n-i)$.

Adversaries. An adversary $\mathbf{A}$ is an efficient Turing machine that interacts with a given set of oracles that appear as black boxes to $\mathbf{A}$. We denote by $\mathbf{A}^{\mathcal{O}}$ the output of $\mathbf{A}$ after interacting with some oracle $\mathcal{O}$. We write $\Delta_{\mathbf{A}}(\mathcal{O}^1; \mathcal{O}^2) := |\Pr[\mathbf{A}^{\mathcal{O}^1} \Rightarrow 1] - \Pr[\mathbf{A}^{\mathcal{O}^2} \Rightarrow 1]|$ for the advantage of $\mathbf{A}$ to distinguish between oracles $\mathcal{O}^1$ and $\mathcal{O}^2$. All probabilities are defined over the random coins of the oracles and those of $\mathbf{A}$, if any. W.l.o.g., we assume that $\mathbf{A}$ never asks queries to which it already knows the answer.

A block cipher E with associated key space $\mathcal{K}$ and message space $\mathcal{M}$ is a mapping $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ such that for every key $K \in \mathcal{K}$, it holds that $E(K, \cdot)$ is a permutation over $\mathcal{M}$. A tweakable block cipher $\widetilde{E}$ with associated key space $\mathcal{K}$, tweak space $\mathcal{T}$, and message space $\mathcal{M}$ is a mapping $\widetilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ such that for every key $K \in \mathcal{K}$ and tweak $T \in \mathcal{T}$, it holds that $\widetilde{E}(K, T, \cdot)$ is a permutation over $\mathcal{M}$. We also write $\widetilde{E}_K^T(\cdot)$ as short form. In this work, we assume that strong pseudo-random permutations allow inputs of variable length, and to be length-preserving. This means, there is no single fixed length, but the length of the ciphertext always equals that of the plaintext and vice versa; moreover,

over all inputs of equal length, the construction is a permutation. The advantage is defined as follows.

Definition 1 (SPRP Advantage). Let $\mathcal{K}$ be a non-empty set and $\mathcal{M} \subset \{0,1\}^*$. Let $\mathit{\Pi} : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ be a length-preserving permutation. Let $\pi \twoheadleftarrow \mathsf{Perm}(\mathcal{M})$ be sampled from the set of all length-preserving permutations of $\mathcal{M}$, and $K \twoheadleftarrow \mathcal{K}$. Then, the SPRP advantage of $\mathbf{A}$ with respect to $\mathit{\Pi}$ is defined as $\mathbf{Adv}^{\mathit{\Pi}}_{\mathrm{SPRP}}(\mathbf{A}) \stackrel{\text{def}}{=} \Delta_{\mathbf{A}}(\mathit{\Pi}_K, \mathit{\Pi}_K^{-1}; \pi, \pi^{-1})$.

Definition 2 (STPRP Advantage). Let $\mathcal{K}$ and $\mathcal{T}$ be non-empty sets and let $\widetilde{E} : \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ denote a tweakable block cipher. Let $\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}(\mathcal{T}, \{0,1\}^n)$ and $K \twoheadleftarrow \mathcal{K}$. Then, the STPRP advantage of $\mathbf{A}$ w.r.t. $\widetilde{E}$ is defined as $\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{E}}(\mathbf{A}) \stackrel{\text{def}}{=} \Delta_{\mathbf{A}}(\widetilde{E}_K, \widetilde{E}_K^{-1}; \widetilde{\pi}, \widetilde{\pi}^{-1})$.

Definition 3 (Almost-XOR-Universal Hash Function). Let $\mathcal{K}$, $\mathcal{X}$, and $\mathcal{Y} \subseteq \{0,1\}^*$ be non-empty sets. Let $H : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a function, keyed by $K \in \mathcal{K}$. We call H ϵ-almost-XOR-universal (ϵ-AXU) if, for all distinct $X, X' \in \mathcal{X}$ and any $\mathit{\Delta} \in \mathcal{Y}$, it holds that $\Pr_{K \twoheadleftarrow \mathcal{K}}[H_K(X) - H_K(X') = \mathit{\Delta}] \leq \epsilon$, where addition and subtraction are in $\mathbb{F}_{2^n}$.

The H-Coefficient Technique. The H-coefficient technique is a proof approach due to Patarin [46]. It assumes that the results of the interaction of an adversary $\mathbf{A}$ with its oracles are collected in a transcript τ of the attack: $\tau = \langle (M_1, C_1, d_1), \ldots, (M_q, C_q, d_q) \rangle$, where (M_i, C_i) denotes the input and output of the i-th query of $\mathbf{A}$ and a Boolean variable d_i denotes the direction of the query; $d_i = 1$ indicates that C_i was result of an encryption query, and $d_i = 0$ that M_i was the result of a decryption query.

The task of $\mathbf{A}$ is to distinguish the real world $\mathcal{O}_{\mathrm{real}}$ from the ideal world $\mathcal{O}_{\mathrm{ideal}}$. A transcript τ is called *attainable* if the probability to obtain τ in the ideal world is non-zero. We denote by $\mathit{\Theta}_{\mathrm{real}}$ and $\mathit{\Theta}_{\mathrm{ideal}}$ the distribution of transcripts in the real and the ideal world, respectively. Then, the fundamental Lemma of the H-coefficients technique, whose proof is given in [13, 46], states:

Lemma 1 (Fundamental Lemma of the H-coefficient Technique [46]). Assume that the set of attainable transcripts is partitioned into two disjoint sets GoodT and BadT. Further assume that there exist $\epsilon_1, \epsilon_2 \geq 0$ such that for any transcript $\tau \in$ GoodT, it holds that

$$\frac{\Pr[\mathit{\Theta}_{\mathrm{real}} = \tau]}{\Pr[\mathit{\Theta}_{\mathrm{ideal}} = \tau]} \geq 1 - \epsilon_1, \quad \text{and} \quad \Pr[\mathit{\Theta}_{\mathrm{ideal}} \in \mathrm{BadT}] \leq \epsilon_2.$$

Then, for all adversaries $\mathbf{A}$, it holds that $\Delta_{\mathbf{A}}(\mathcal{O}_{\mathrm{real}}; \mathcal{O}_{\mathrm{ideal}}) \leq \epsilon_1 + \epsilon_2$.

3 On the Minimal Number of Required Primitive Calls

This section shows that any PRP with fewer than $3\ell - 1$ calls for messages of 2ℓ blocks to a primitive with n-bit tweak size and n-bit state size is insecure. We follow the approach by [43], who proved that an SPRP based on a single-keyed classical block cipher needs at least 2ℓ calls to the primitive for ℓ-block messages.

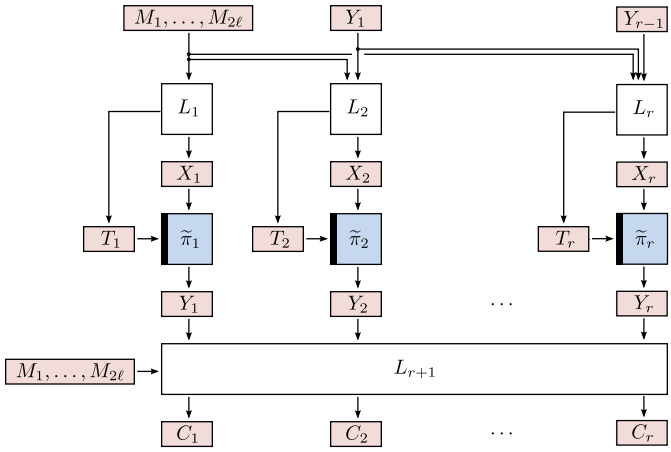


Fig. 2: Generic model of a PRP that consists of at most $r \leq 3\ell - 2$ calls to tweakable block ciphers $\widetilde{\pi}_i$ for messages of 2ℓ blocks.

3.1 Generic Construction

Define positive integers n, τ, and ℓ, and let $\mathcal{M} \subseteq \{0,1\}^*$ denote a space for which $(\{0,1\}^n)^{2\ell} \subseteq \mathcal{M}$. Let $r \leq 3\ell - 2$ and let

$$\widetilde{\pi}_i : \{0,1\}^\tau \times \{0,1\}^n \to \{0,1\}^n, \quad \text{for all } 1 \leq i \leq r,$$

denote tweakable permutations with tweak space $\{0,1\}^\tau$ and state size n. Let $\Pi[\widetilde{\pi}_1, \ldots, \widetilde{\pi}_r] : \mathcal{M} \to \mathcal{M}$ be a length-preserving cipher that employs as its only non-linear functions in total r calls to the permutations $\widetilde{\pi}_1, \ldots, \widetilde{\pi}_r$. For simplicity, we also write Π as short form, hereafter. All further components of Π are linear over $\mathbb{F}_{2^n}$. For any such construction, we can formulate this as follows. Let X_i denote the input to π_i, T_i the tweak to π_i, and let $Y_i \leftarrow \pi_i(X_i)$ denote its output. The linear operations in Π must be describable as non-zero linear functions $L_i : \mathcal{M} \times (\{0,1\}^n)^{i-1} \to \{0,1\}^n \times \{0,1\}^\tau$, for $1 \leq i \leq r$, and an additional non-zero linear function $L_{r+1} : \mathcal{M} \times (\{0,1\}^n)^r \to \mathcal{M}$ that, for all given inputs $(M, Y_1, \ldots, Y_r) \in \mathcal{M} \times (\{0,1\}^n)^r$, outputs C s.t. it holds that $|C| = |M|$. Then, we can describe the encryption with $\Pi(M)$ as

$$\begin{aligned} (X_i, T_i) &\leftarrow L_i\left(M, Y_1, \ldots, Y_{i-1}\right), && \text{for all } 1 \leq i \leq r, \\ Y_i &\leftarrow \widetilde{\pi}^{T_i}(X_i), && \text{for all } 1 \leq i \leq r, \text{ and} \\ C &\leftarrow L_{r+1}(M, Y_1, \ldots, Y_r). \end{aligned}$$

Π must be correct for all inputs, i.e., for all $M, C \in \mathcal{M}$, it must hold that $\Pi^{-1}(\Pi(M)) = M$ and $\Pi(\Pi^{-1}(C)) = C$. Figure 2 gives an illustration.

Remark 1. It may not be instantaneously clear why the generic construction above covers all considered schemes. Note that it computes the values X_i and T_i by a non-zero linear function of M, Y_1, Y_2, …, Y_{i-1}. So, the previous values Y_i

can also be used to generate X_i. Indeed, it is generic enough to include all such constructions where the only non-linear components are the permutation calls.

For simplicity, we consider independent permutations with tweak domain $\mathbb{F}_2^\tau$ in this section. For efficiency, our proposal later in this work will employ only a single tweakable primitive with a composite tweak domain $\mathcal{T}_D = \mathcal{D} \times \mathbb{F}_2^\tau$, where $\mathcal{D}$ is a non-empty set of domains. So, this approach achieves the same goal of having independent permutations. We consider that τ is the effectively usable size of the tweaks without domains.

3.2 A PRP Attack on Constructions with At Most $3\ell - 2$ Calls

Case $\tau = n$. Let $\mathbf{A}$ be an adversary with the goal to distinguish the outputs of a variable-input-length PRP Π under a secret key as above from an ideal PRP. First, $\mathbf{A}$ chooses two messages M and M' of 2ℓ blocks each, i.e., $M = (M_1, \ldots, M_{2\ell})$ and $M' = (M'_1, \ldots, M'_{2\ell})$. We define the differences $\Delta M = M - M'$, and analogously the differences ΔX_i, ΔY_i, and ΔC in the obvious manner. Choose M and M' such that it holds that $\Delta X_i = 0$ and $\Delta T_i = 0$, for $1 \leq i \leq \ell - 1$. Note that such a choice of M and M' must be possible since these variables correspond to $2\ell - 2$ equations ($\ell - 1$ equations for adjusting the values ΔX_i and $\ell - 1$ equations for adjusting the values ΔT_i) and there exist 2ℓ blocks ΔM_i. For instance, the adversary can efficiently derive an element N from the null space of $L_1, \ldots, L_{2(\ell-1)}$. It chooses M arbitrarily and derives $M' = M + N$.

From $\Delta X_i = 0^n$ and $\Delta T_i = 0^\tau$ for $1 \leq i \leq \ell - 1$, it follows that $\Delta Y_i = \widetilde{\pi}^{T_i}(X_i) \oplus \widetilde{\pi}^{T'_i}(X'_i) = 0^n$, for all $1 \leq i \leq \ell - 1$. The non-linear layer of calls to the tweakable block cipher maps $(\Delta X_1, \ldots, \Delta X_r)$ to $(\Delta Y_1, \ldots, \Delta Y_r)$. So, we obtain the equation

$$L_{r+1}(\Delta M, \underbrace{\Delta Y_1, \ldots, \Delta Y_{\ell-1}}_{=(0,\ldots,0)}, \Delta Y_\ell, \ldots, \Delta Y_r) = \Delta C.$$

Since $\mathbf{A}$ fixed ΔM and chose M and M' so that $\Delta X_1 = \ldots = \Delta X_{\ell-1} = 0^n$ and $\Delta T_1 = \ldots = \Delta T_{\ell-1} = 0^\tau$, we obtain $\Delta Y_1, \ldots, \Delta Y_{\ell-1} = 0^n$. So, there are at most $2\ell - 1$ free variables $\Delta Y_\ell, \ldots \Delta Y_r$, and 2ℓ equations for $\Delta C_1, \ldots, \Delta C_{2\ell}$, which implies that 2ℓ blocks of ΔC are a linear combination of $2\ell - 1$ values $\Delta Y_\ell, \ldots, \Delta Y_r$. So, in the real construction, L_{r+1} defines a map from $2\ell - 1$ to 2ℓ n-bit variables, and $\mathbf{A}$ can efficiently derive a solution $\Delta Y_\ell, \ldots, \Delta Y_r$ from the null space of the equation system. This becomes a distinguishing event since it happens with probability one in the real construction and with probability $1/2^n$ in the ideal world for this example. The distinguishing advantage is hence $1 - 1/2^n$. Hence, $\mathbf{A}$ can simply query it with two messages as above and output real if such a non-zero linear function L exists and random otherwise, as summarized in Algorithm 1.

For general values of τ. A similar attack is applicable for general values of τ. Though, we have to consider linearity over $\mathbb{F}_2$ then. Define

$$s = \left\lfloor \frac{2\ell n}{n + \tau} \right\rfloor - 1.$$

Algorithm 1 PRP attack on generic constructions Π with at most $3\ell - 2$ primitive calls, here for $\tau = n$.

function $\mathbf{A}^{\Pi}$
  Choose M_i for $1 \leq i \leq 2\ell$ arbitrarily
  Choose M'_i for $\ell \leq i \leq 2\ell$ s. t. it holds that
    $L_i(\Delta M_i) = (\Delta X_i, \Delta T_i) = (0^n, 0^\tau)$, for $1 \leq i \leq 2(\ell - 1)$
  Ask for the encryption of $C = \Pi(M)$ and $C' = \Pi(M')$
  Derive $\Delta C = C' - C$
  if there exists $(\Delta Y_\ell, \ldots, \Delta Y_r)$, s. t. $L_{r+1}(\Delta M, \Delta Y) = \Delta C$ **then**
    return "Real"
  return "Random"

The adversary chooses $M \in (\mathbb{F}_2^n)^{2\ell}$ arbitrarily, and $M' \in (\mathbb{F}_2^n)^{2\ell}$ with $M \neq M'$ s. t. $\Delta X_1 = \ldots \Delta X_s = 0^n$ and $\Delta T_1 = \ldots = \Delta T_s = 0^\tau$. Note that we consider the inputs $X_i \in \mathbb{F}_2^n$ and the tweaks $T_i \in \mathbb{F}_2^\tau$ as blocks. Again, such a choice of M' exists for the same reason as above and can be found efficiently from the null space of the linear functions L_1, L_2, ... that are involved in the computation of $\Delta X_1, \ldots, \Delta X_s$ and $\Delta T_1, \ldots, \Delta T_s$. Again, we obtain $\Delta Y_i = 0^n$, for $1 \leq i \leq s$ for the real construction. We obtain the equation

$$L_{r+1}(\Delta M, \underbrace{\Delta Y_1, \ldots, \Delta Y_s}_{=(0,\ldots,0)}, \Delta Y_{s+1}, \ldots, \Delta Y_r) = \Delta C.$$

The blocks $\Delta Y_{s+1}, \ldots, \Delta Y_r$ contain $(r-s)n$ bits, that are mapped through L_{r+1} to $\Delta C_{2\ell n}$ bits. For all schemes Π that use r calls to the primitive with

$$(r - s) \cdot n < 2\ell n, \qquad \text{which leads to} \qquad r < 2\ell \left(1 + \frac{n}{n + \tau}\right) - 1,$$

we obtain a compressing mapping. Then, there exist are more equations than variables, and the distinguisher as before applies. However, the advantage may be smaller and depends on the values of r, n, and τ.

4 Definition of The Basic ZCZ Construction

This section defines the basic ZCZ SPRP scheme. First, we consider messages that consist of at most $2n$ blocks, and will extend it thereupon to all messages whose length is a multiple of $2n$ bits. The subsequent section will then further define it for messages whose length is not necessarily a multiple of $2n$ bits.

Let $n, \tau, k, d \geq 1$ be integers with $d \ll n$; in the remainder, we restrict our interest to $n = \tau$, and define $N \stackrel{\text{def}}{=} 2^n$ as an alias. Let $\mathcal{B} = \{0,1\}^{2n}$ define a *di-block* (or dual block, double block), i.e., $2n$ bits. Moreover, we define non-empty sets of tweaks $\mathcal{T} = \{0,1\}^\tau$ and keys $\mathcal{K} = \{0,1\}^k$, as well as two sets for domains and indices $\mathcal{D} = \{\mathsf{t}, \mathsf{s}, \mathsf{c}, \mathsf{b}, \mathsf{t\$}, \mathsf{s\$}, \mathsf{c\$}, \mathsf{b\$}, \mathsf{xl}, \mathsf{xr}, \mathsf{yl}, \mathsf{yr}, \mathsf{p}, \mathsf{kd}\} \subseteq \{0,1\}^d$, and

Algorithm 2 Definition of the encryption algorithm of $\mathrm{ZCZ}[\widetilde{E}]$ given a tweakable block cipher $\widetilde{E}$. The code in the boxes is only part of $\mathrm{ZCZ}^*[\widetilde{E}]$ in Algorithm 3.

10: **function** $\mathrm{ZCZ}[\widetilde{E}_K](M)$
11: $r \leftarrow |M| \bmod 2n$
12: $\ell \leftarrow (|M| - r)/2n$
13: $z \leftarrow \lceil (\ell - 1)/n \rceil$
14: $L'_* \leftarrow \varepsilon$; $R'_* \leftarrow \varepsilon$
15: $\textsc{Parse}(M, \ell)$
16: $\textsc{TopEnc}[\widetilde{E}_K]()$
17: **if** $r > 0$ **then**
18: $\boxed{\textsc{PartialTopEnc}[\widetilde{E}_K]()}$
19: $\textsc{LastTopEnc}[\widetilde{E}_K](X_L, X_R)$
20: $\textsc{MidLayer}[\widetilde{E}_K](S, T)$
21: $\textsc{BotEnc}[\widetilde{E}_K]()$
22: $\textsc{LastBotEnc}[\widetilde{E}_K](Y_L, Y_R)$
23: **if** $r > 0$ **then**
24: $\boxed{\textsc{PartialBotEnc}[\widetilde{E}_K]()}$
25: $C \leftarrow (L'_1 \| R'_1 \| \cdots \| L'_\ell \| R'_\ell \| L'_* \| R'_*)$
26: **return** C

30: **procedure** $\textsc{TopEnc}[\widetilde{E}_K]$
31: $X^*_L \leftarrow X^*_R \leftarrow 0^n$
32: **for** $i \leftarrow 1 \ldots \ell - 1$ **do**
33: $X_i \leftarrow \widetilde{E}_K^{\mathsf{t},i,R_i}(L_i)$
34: $X^*_L \leftarrow X^*_L + \alpha^{\ell-1-i} X_i$
35: $X^*_R \leftarrow X^*_R + (\alpha^2)^{\ell-1-i}(X_i + R_i)$
36: $X_L \leftarrow \widetilde{E}_K^{\mathsf{xl},\ell,X^*_R}(X^*_L)$
37: $X_R \leftarrow \widetilde{E}_K^{\mathsf{xr},\ell,X^*_L}(X^*_R)$

40: **procedure** $\textsc{MidLayer}[\widetilde{E}_K](S, T)$
41: $S_0 \leftarrow S$
42: **for** $i \leftarrow 1 \ldots z$ **do**
43: $S_i \leftarrow \widetilde{E}_K^{\mathsf{s},0,i}(S_{i-1})$
44: **for** $i \leftarrow 1 \ldots z$ **do**
45: **for** $j \leftarrow 1 \ldots n$ **do**
46: $Z_{i,j} \leftarrow \widetilde{E}_K^{\mathsf{c},(i-1)n+j,T}(S_i)$

50: **procedure** $\textsc{LastTopEnc}[\widetilde{E}_K](X_L, X_R)$
51: $S \leftarrow \widetilde{E}_K^{\mathsf{t\$},\ell,R_\ell+X_R}(L_\ell + X_L)$
52: $T \leftarrow \widetilde{E}_K^{\mathsf{s\$},\ell,S}(R_\ell + X_R)$

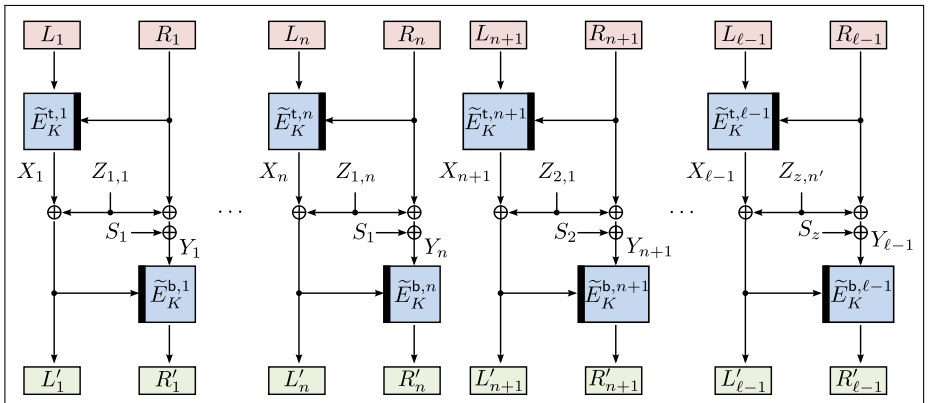
60: **procedure** $\textsc{BotEnc}[\widetilde{E}_K]$
61: $Y^*_L \leftarrow 0^n$
62: $Y^*_R \leftarrow 0^n$
63: **for** $i \leftarrow 1 \ldots z - 1$ **do**
64: **for** $j \leftarrow 1 \ldots n$ **do**
65: $k \leftarrow (i-1)n + j$
66: $L'_k \leftarrow X_k + Z_{i,j}$
67: $Y_k \leftarrow R_k + Z_{i,j} + S_i$
68: $R'_k \leftarrow \widetilde{E}_K^{\mathsf{b},k,L'_k}(Y_k)$
69: $Y^*_L \leftarrow Y^*_L + (\alpha^2)^{\ell-1-k}(Y_k + L'_k)$
70: $Y^*_R \leftarrow Y^*_R + (\alpha)^{\ell-1-k} Y_k$
71: **for** $j \leftarrow 1 \ldots \ell - 1 - (z-1)n$ **do**
72: $k \leftarrow (z-1)n + j$
73: $L'_k \leftarrow X_k + Z_{z,j}$
74: $Y_k \leftarrow R_k + Z_{z,j} + S_z$
75: $R'_k \leftarrow \widetilde{E}_K^{\mathsf{b},k,L'_k}(Y_k)$
76: $Y^*_L \leftarrow Y^*_L + (\alpha^2)^{\ell-1-k}(Y_k + L'_k)$
77: $Y^*_R \leftarrow Y^*_R + \alpha^{\ell-1-k} Y_k$
78: $Y_L \leftarrow \widetilde{E}_K^{\mathsf{yl},\ell,Y^*_R}(Y^*_L)$
79: $Y_R \leftarrow \widetilde{E}_K^{\mathsf{yr},\ell,Y^*_L}(Y^*_R)$

80: **procedure** $\textsc{LastBotEnc}[\widetilde{E}_K](Y_L, Y_R)$
81: $L'_\ell \leftarrow \widetilde{E}_K^{\mathsf{c\$},\ell,T}(S) + Y_L$
82: $R'_\ell \leftarrow \widetilde{E}_K^{\mathsf{b\$},\ell,T}(L'_\ell + Y_L) + Y_R$

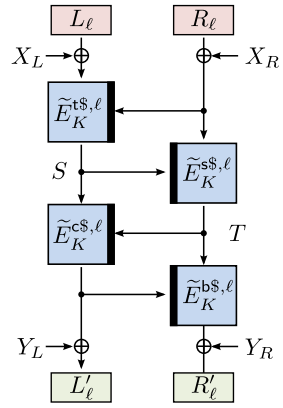
90: **procedure** $\textsc{Parse}(M, \ell)$
91: $i \leftarrow \ell \cdot 2n$
92: $(L_1, R_1, \cdots, L_\ell, R_\ell) \xleftarrow{n} M[0..i-1]$
93: **if** $r > 0$ **then**
94: $(L_*, R_*) \xleftarrow{n} M[i..|M|]$

$\mathcal{I} \subseteq \{1, \ldots, 2^n - 1\}$, s.t. the elements of $\mathcal{D}$ denote pairwise distinct integers. The purpose of domains and indices is to define an extended tweak set $\mathcal{T}_{D,I} = \mathcal{D} \times \mathcal{I} \times \mathcal{T}$ for a tweakable block cipher $\widetilde{E} : \mathcal{K} \times \mathcal{T}_{D,I} \times \{0,1\}^n \to \{0,1\}^n$.

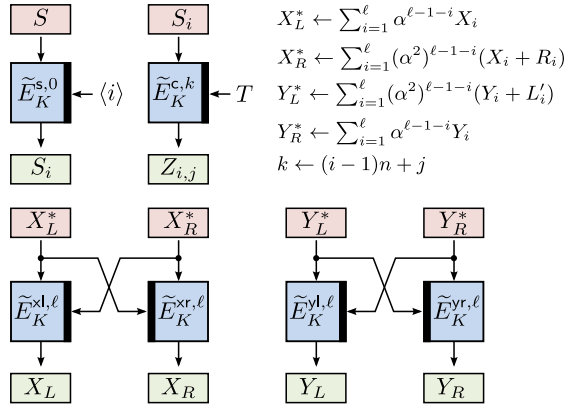
We interpret n-bit inputs also as elements in the finite field $\mathbb{F}_{2^n}$, which is the Galois Field $\mathbb{GF}(2^n)$ with a fixed irreducible polynomial $p(\mathtt{x})$, where we interpret a bit string $(x_{n-1} \ldots x_1 x_0)$ as polynomial $\sum_{i=0}^{n-1} a_i \cdot \mathtt{x}^i$ in $\mathbb{F}_{2^n}$, where bit x_i represents the coefficient $a_i \in \{0,1\}$, for $0 \le i \le n-1$, and the most significant bit is the leftmost, and the least significant bit is the rightmost bit. We consider all multiplications, and all additions of n-bit values to be in $\mathbb{F}_{2^n}$;



(a) Encryption of the first $\ell - 1$ di-blocks.



(b) Encryption of the final di-block.



(c) Deriving the chaining values S_i and $Z_{i,j}$; $n' \leftarrow (\ell - 1) \bmod n$.

Fig. 3: Encryption of a message with ℓ complete di-blocks with construction $\mathrm{ZCZ}[\widetilde{E}_K]$.

though, all additions in sub- and superscripts (that denote indices) represent normal integer additions. For $n = 128$, we suggest the irreducible polynomial to be that of OCB and GCM, i.e., $p(\mathtt{x}) = \mathtt{x}^{128} + \mathtt{x}^7 + \mathtt{x}^2 + \mathtt{x} + \mathtt{1}$. We define a generator element $\alpha \in \mathbb{F}_{2^n}$, where we suggest $\alpha = \mathtt{x}$ for the case $n = 128$ for the sake of concreteness. Moreover, note that we also will use the field $\mathbb{F}_{2^{n+\tau}}$, where we suggest for $n = \tau = 128$ the irreducible polynomial $p(\mathtt{x}) = \mathtt{x}^{256} + \mathtt{x}^{10} + \mathtt{x}^5 + \mathtt{x}^2 + \mathtt{1}$.

First, we consider the basic construction $\mathrm{ZCZ}[\widetilde{E}_K]$, which takes as input a secret key $K \in \mathcal{K}$ and a plaintext $M \in \mathcal{B}^{\leq n}$ consisting of $\ell \in [1..n]$ di-blocks. We stress that a message must consist of at least a single complete di-block. M is split into a sequence of di-blocks (L_i, R_i), for $1 \leq i \leq \ell$. The first $\ell - 1$

complete di-blocks are processed similarly as in the ZHASH construction by Iwata et al. [28]: for each di-block (L_i, R_i), the left n-bit branch is used as state input, and the right n-bit branch is used as tweak input to $\widetilde{E}_K^{\mathsf{t},i}$. The calls are domain-separated by the indices i. The TBC outputs X_i are multiplied by a power of the generating element $\alpha^{\ell-1-i}$ and accumulated to a value X_L^* using the Horner rule: $X_L^* \leftarrow \sum_{i=1}^{\ell-1} \alpha^{\ell-1-i} \cdot X_i$. Similarly, we derive a second value $X_R^* \leftarrow \sum_{i=1}^{\ell-1} (\alpha^2)^{\ell-1-i} \cdot (X_i + R_i)$ that takes also the tweak inputs R_i into account. Next, both values are encrypted using one of them as input and the other one as tweak under distinct tweak domains, similar to the finalization in [40]:

$$X_L \leftarrow \widetilde{E}_K^{\mathsf{xl},\ell,X_R^*}(X_L^*) \quad \text{and} \quad X_R \leftarrow \widetilde{E}_K^{\mathsf{xr},\ell,X_L^*}(X_R^*)$$

The outputs X_L and X_R are used to mask the branches of the final di-block, L_ℓ and R_ℓ. The different α and α^2 prevent that a collision in X_L would automatically lead to a collision also in X_R and vice versa. The encryption prevents that differences in the masks cancel with differences in the final di-block.

So, the final di-block depends on all other message bits. The final di-block is processed by a four-round Feistel-like network of TBC calls, where each TBC call employs a distinct domain independent from all others. The processing of the final di-block finally is a four-round Feistel-like network in the spirit of the $\widetilde{\Psi}_3$ construction by Coron et al. [16]. Here, it generates two intermediate values S and T after the first and second call to $\widetilde{E}$, respectively.

From S, we derive a chaining value $S_1 \leftarrow \widetilde{E}_K^{\mathsf{s},0,1}(S)$. Moreover, we derive from S and T $\ell - 1$ further chaining values $Z_{1,j} \leftarrow \widetilde{E}^{\mathsf{c},j,T}(S_1)$ by a TBC call each, again under distinct domains. The chaining values are employed in the middle layer of our construction and ensure that each di-block depends on all others. For the j-th di-block, the chaining value $Z_{1,j}$ is added to both branches of the j-th block. Moreover, S_1 is also added to the right branch of each di-block: $Y_j \leftarrow R_j + Z_{1,j} + S_j$. This step is required to prevent adversaries from observing the differences $\Delta Z_{1,j}$. We elaborate on attacks on preliminary versions of ZCZ in Appendix D.

After the middle layer, the values of the right branches of all di-blocks, Y_j, are also multiplied by $\alpha^{\ell-1-i}$, using the Horner rule, and accumulated to $Y_L^* \leftarrow \sum_{i=1}^{\ell-1} (\alpha^2)^{\ell-1-i}(Y_i + L_i')$ and $Y_R^* \leftarrow \sum_{i=1}^{\ell-1} \alpha^{\ell-1-i} Y_i$. Again, both values are encrypted to Y_L and Y_R using the same manner as for X_L and X_R, but distinct tweak domains. For all complete $\ell - 1$ di-blocks, both branches are again processed by another ZHASH layer at the bottom to compute the ciphertexts: $L_j' \leftarrow X_j$ and $R_j' \leftarrow \widetilde{E}_K^{\mathsf{b},i,L_j'}(Y_j)$. After the final, ℓ-th, complete di-block has undergone two further Feistel rounds, Y_L added to the left branch, and Y_R is added to the right branch of the ℓ-th di-block.

Extension to Longer Messages. Messages that consist of more than n di-blocks are partitioned into *chunks*. The i-th (complete) chunk denotes the series of the n consecutive di-blocks $(L_{(i-1)n+1}, R_{(i-1)n+1}, \ldots L_{i \cdot n}, R_{i \cdot n})$, and employs

the chaining values S_i and $Z_{i,j}$. We derive all chaining values under distinct domains as before; from S, we derive furthermore $\ell - 1$ chaining values $Z_{i,j}$ by a TBC call each as before. For the i-th chunk, S_i is computed as $S_i \leftarrow \widetilde{E}_K^{\mathsf{s},0,i}(S)$. Then, for $j \in [1..n]$, $Z_{i,j}$ for the j-th block of the i-th chunk is generated as $Z_{i,j} \leftarrow \widetilde{E}_K^{\mathsf{c},0,n(i-1)+j}(S_i)$. $Y_{n(i-1)+j}$ is then computed as

$$Y_{n(i-1)+j} \leftarrow R_{n(i-1)+j} + S_i + Z_{n(i-1)+j}.$$

The rest of the computations remain unchanged. Letting j take any value in $[1..\ell]$, we can rewrite this as

$$Y_j \leftarrow R_j + S_{\lceil j/n \rceil} + Z_j. \tag{2'}$$

The encryption of ZCZ$[\widetilde{E}_K]$ is defined in Algorithm 2, and illustrated in parts in Figure 3, already for more than n complete di-blocks. The figure employs bold bars in the blocks of $\widetilde{E}$ to indicate the parts of the tweaks that stem from $\mathcal{T}$. A precise description of the decryption is omitted for the sake of space limitation since it can be defined in the obvious way.

Instantiating The TBC and Extending The Tweak Space. A TBC $\widetilde{E}$ with native domain $\mathcal{T}_{D,I}$ would most probably be not very efficient or does not even exist out-of-the-box. In practice, however, one could employ instead a TBC $\underline{\widetilde{E}} : \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ with tweak domain $\mathcal{T} \subseteq \{0,1\}^n$ and augment its tweak domain.

Recall that $\mathcal{D} \subseteq \{0,1\}^d$ for $d \ll n$. Let $\mathsf{kd} \in \mathcal{D}$ be a distinct domain that is not used otherwise, and define $m = n + \tau$ and $m' = \lceil m/n \rceil$. Assume that $m' < n - d$. Then, we can generate an m-bit key from

$$\begin{aligned} K_i &\leftarrow \widetilde{E}_K^{\mathsf{kd} \,\|\, \langle i \rangle_{n-d}}(0), \quad \text{for } 1 \le i \le m' \\ K' &\leftarrow \mathsf{msb}_m(K_1 \,\|\, \ldots \,\|\, K_{m'}) \end{aligned}$$

We can define a masking function δ similar as in XEX [48], using the same polynomials, e.g., $(\mathtt{x})$ and $(\mathtt{x}+\mathtt{1})$ (i.e., the polynomials that are represented as integers 2 and 3, respectively). Note that the polynomials here are in $\mathbb{F}_{2^{n+\tau}}$. Given a tuple of domain $D \in \mathcal{D}$, counter $I \in \mathcal{I}$, and key K', the masking function computes

$$(U \,\|\, V) \leftarrow \delta(D, I, K') \stackrel{\text{def}}{=} (\mathtt{x})^I \cdot (\mathtt{x}+\mathtt{1})^D \cdot K'.$$

The resulting $(n+\tau)$-bit mask $(U \,\|\, V)$ is split into an n-bit part U and a τ-bit part V and added to input, output, and tweak part of the TBC inputs. So, we define the TBC $\widetilde{E}$ for an input-tweak tuple $(X, T) \in \{0,1\}^n \times \mathcal{T}$ as

$$\begin{aligned} \widetilde{E}_{K,K'}^{D,I,T}(X) &\stackrel{\text{def}}{=} \underline{\widetilde{E}}_K^{T+V}(X+U) + U \quad \text{with,} \\ (U \,\|\, V) &\leftarrow \delta(D, I, K') \end{aligned}$$

Algorithm 3 Functions of the encryption algorithm of $\mathrm{ZCZ}^*[\widetilde{E}]$ for messages whose length is not necessarily a multiple of $2n$ bit (but at least $2n$ bit). Recall that $r = |M| \bmod 2n$.

```
10: procedure PartialTopEnc[Ẽ_K]
11:     M_ℓ ← L_ℓ ‖ R_ℓ
12:     M_* ← pad_{2n}(L_* ‖ R_*)
13:     (L̄_*, R̄_*) ←^n M_*
14:     (U_ℓ, V_ℓ) ← H[Ẽ_K, 0](L̄_*, R̄_*)
15:     L_ℓ ← L_ℓ + U_ℓ
16:     R_ℓ ← R_ℓ + V_ℓ

20: function msb_x(X)
21:     return X[0..x − 1]

30: function pad_x(X)
31:     return X ‖ 1 ‖ 0^{x−|X|−1}

40: procedure PartialBotEnc[Ẽ_K]
41:     (P, Q) ← H[Ẽ_K, 2](L_ℓ + L'_ℓ, R_ℓ + R'_ℓ)
42:     W ← msb_r(P ‖ Q) ‖ 0^{2n−r}
43:     (P_*, Q_*) ←^n W
44:     L'_* ← L_* + P_*
45:     R'_* ← R_* + Q_*
46:     (L̄'_*, R̄'_*) ←^n pad_{2n}(L'_* ‖ R'_*)
47:     (U'_ℓ, V'_ℓ) ← H[Ẽ_K, 4](L̄'_*, R̄'_*)
48:     L'_ℓ ← L'_ℓ + U'_ℓ
49:     R'_ℓ ← R'_ℓ + V'_ℓ

50: function H[Ẽ_K, i](U, V)
51:     U' ← Ẽ_K^{p,i,V}(U)
52:     V' ← Ẽ_K^{p,i+1,V}(U)
53:     return (U', V')
```

$$U \leftarrow \mathsf{msb}_n(U \,\|\, V)$$
$$V \leftarrow \mathsf{lsb}_\tau(U \,\|\, V).$$

Since K' is uniquely derived from $\widetilde{E}_K$, we omit writing it hereafter. We stress that such an instantiation would imply a restriction of the number of employed indices and domains. For example, for $m = 128$, Rogaway [48] showed that the powers of those polynomials will not collide for indices up to $(I, D) \in [-2^{108} \ldots + 2^{108}] \times [-2^7 \ldots + 2^7]$. For selected larger values of m, e.g., $m \in \{256, 512\}$ one can find further bounds in [18].

5 The Construction ZCZ* for Messages with Partial Final Di-block

Next, we extend the definition of ZCZ for messages whose length is not a multiple of $2n$ bits, where we denote the last $r \leftarrow |M| \bmod 2n$ bits as *partial di-block*. Our approach is inspired by the generic designs by Minematsu [39] and Nandi [42]. First, we briefly review the latter before we show our adapted version of ZCZ^*.

The DE Domain Extender. In 2009 [42], Nandi proposed the domain extender $\mathrm{DE}[\Pi, F, H] : \{0,1\}^{\geq n} \to \{0,1\}^{\geq n}$ that takes a blockwise-operating length-preserving permutation $\Pi : (\{0,1\}^n)^+ \to (\{0,1\}^n)^+$, a PRF $F : \{0,1\}^n \to \{0,1\}^n$, and an XOR-universal hash function $H : \{0,1\}^n \times \{0,1\}^{2n} \to \{0,1\}^n$, and produces a length-preserving permutation over bit strings of any length $\geq n$ bits. A message $M \in \{0,1\}^{\geq n}$ is split into blocks $(M_1, \ldots, M_{\ell-1}, M_\ell) \xleftarrow{n} M$, and $\mathrm{DE}[\Pi, F, H]$ computes the corresponding ciphertext $C = (C_1, \ldots, C_\ell)$ as

$$M^*_{\ell-1} \leftarrow H(M_{\ell-1}, M_\ell)$$

$$
\begin{aligned}
(C_1, \ldots, C_{\ell-2}, C^*_{\ell-1}) &\leftarrow \Pi(M_1, \ldots, M_{\ell-2}, M^*_{\ell-1}) \\
C_\ell &\leftarrow F(M^*_{\ell-1} + C^*_{\ell-1}) +_{|M_\ell|} M_\ell \\
C_{\ell-1} &\leftarrow H(C^*_{\ell-1}, C_\ell),
\end{aligned}
$$

where $x +_n y \stackrel{\text{def}}{=} \mathsf{msb}_n(x) + y$ for any $x, y \in \{0,1\}^*$ and integer n. To obtain that DE is a permutation, the hash function H must satisfy $H(H(M_{\ell-1}, M_\ell), M_\ell) = M_{\ell-1}$ for any allowed input $M_{\ell-1}$, M_ℓ (see [42, Remark 2]).

Extending our Construction to Support Partial Di-Blocks. We require that the message has at least one full di-block. Let $M_* = (L_*, R_*)$ be the partial message di-block that follows after ℓ complete di- blocks. Further assume that the partial di-block consists of $\geq n$ bits that are split into $|L_*| = n$ and $|R_*| < n$. The right part is padded to n bits by a single 1 and as many zero bits as necessary to extend it to n bits: $\overline{R}_* \leftarrow \mathsf{pad}_n(R_*)$. The values are given as inputs to a hash function $\mathcal{H}[\widetilde{E}_K, i]$, with $i = 0$, that is illustrated on the right side of Figure 4. $\{H\}$ uses one of the two n-bit values as state and the other one as tweak input for two calls to $\widetilde{E}_K$ under distinct tweaks:

$$
U' \leftarrow \widetilde{E}^{\mathsf{p},i,V}_K(U) \quad \text{and} \quad V' \leftarrow \widetilde{E}^{\mathsf{p},i+1,V}_K(U).
$$

The $2n$-bit output (U', V') is added to the final complete di-block. The resulting final di-block (L_ℓ, R_ℓ) is then processed by $\mathrm{ZCZ}[\widetilde{E}_K]$. The sum of $(L_\ell, R_\ell) + (L'_\ell, R'_\ell)$ is then given again into $\mathcal{H}[\widetilde{E}_K, i]$, with $i = 2$ to produce a $2n$-bit value (P'_ℓ, Q'_ℓ). The most significant r bits of it are added to the final partial di-block to obtain the partial ciphertext di-block M'_*. M'_* is again padded to $2n$ bits and given as input to a third call to $\mathcal{H}[\widetilde{E}_K, i]$, with $i = 4$. The hash output is added to the final ciphertext di-block to produce M'_ℓ. If the partial di-block consists of less than n bits, it is also padded to $2n$ bits and processed analogously.

So, the hash function H from the original definition of $\mathrm{DE}[\Pi, F, H]$ is given by

$$
H(M_\ell, M_*) \stackrel{\text{def}}{=} M_\ell + \mathcal{H}[\widetilde{E}_K, i](\mathsf{pad}_{2n}(M_*))
$$

The requirement that

$$
H(H(M_\ell, M_*), M_*) = M_\ell + \mathcal{H}[\widetilde{E}_K, i](\mathsf{pad}_{2n}(M_*)) + \mathcal{H}[\widetilde{E}_K, i](\mathsf{pad}_{2n}(M_*)) = M_\ell
$$

holds for arbitrary M_ℓ and M_*.

Remark 2. Note that due to the structure of the used domain extension, the extended construction ZCZ^* still requires messages and ciphertexts to consist of at least one complete di-block, i.e., of at least $2n$ bit. A further minor improvement in future work could be the integration of smaller messages. For instance, the use of the very recent length-doubling construction LDT by Chen et al. [14] could reduce the minimal message length to $n+1$ bits. Though, this step would require an appropriate integration. Moreover, although we separated the definitions for clarity, ZCZ^* is a variable-input-length SPRP for lengths $\geq 2n$ bit.

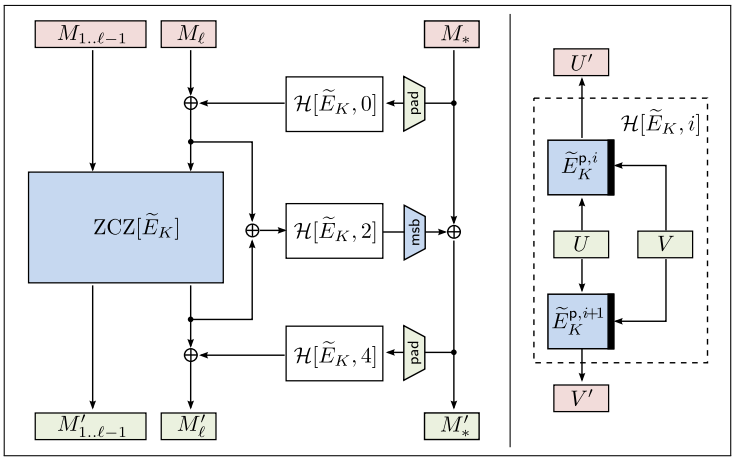


Fig. 4: Encryption of a partial message $M_1, \ldots, M_\ell, M^*$ whose length is not a multiple of $2n$ bit with $\mathrm{ZCZ}^*[\widetilde{E}_K]$. All preceeding di-blocks $M_1, \ldots, M_\ell$ are processed with $\mathrm{ZCZ}[\widetilde{E}_K]$ as before.

6 Security Analysis of ZCZ and ZCZ*

This section studies the SPRP security of ZCZ and ZCZ*. First, we consider the security of the basic construction. Thereupon, we consider the extensions for inputs whose length is not necessarily a multiple of $2n$ bits, but at least $2n$ bits. Finally, we take a look at the security of ZCZ*.

6.1 Security of The Basic Construction

Theorem 1. Let $\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}(\mathcal{T}_{D,I}, \{0,1\}^n)$. Let $\mathbf{A}$ be an SPRP adversary on $\mathrm{ZCZ}[\widetilde{\pi}]$, s.t. $\mathbf{A}$ asks at most q queries of domain $\mathcal{B}^{\leq n}$, that sum up to at most σ di-blocks in total. Then

$$\mathbf{Adv}^{\mathrm{SPRP}}_{\mathrm{ZCZ}[\widetilde{\pi}]}(\mathbf{A}) \leq \frac{3\sigma^2 + 9q^2}{2N^2}.$$

Proof. The queries 1 through q by $\mathbf{A}$ are collected in a transcript τ where we define two disjoint sets of indices E and D s.t. $[1..q] = E \sqcup D$, and it holds that E consists of exactly those indices i s.t. the i-th query of $\mathbf{A}$ is an encryption query; similarly, D consists of exactly those indices i s.t. the i-th query of $\mathbf{A}$ is an decryption query. We define ℓ^i to be the number of di-blocks in the i-query, where $\ell^i \leq n$.

In both worlds, the adversary's queries are answered immediately with the corresponding outputs; certain internal parts of the transcript will be revealed to the adversary after it made all its queries, but before it outputs its decision bit that represents its guess of which world it interacted with. The internal parts consist of $S^i, T^i, S^i_1, X^i_L, X^i_R, Y^i_L, Y^i_R$ for $i \in [1..q]$ and $Z^i_{1,j}$ for $i \in [1..q], j \in [1..\ell^i - 1]$; for ease of notation, we write Z^i_j to refer to $Z^i_{1,j}$.

EQUATIONS. First, we write the internal variables X_j^i, Y_j^i for $i \in [1..q], j \in [1..\ell^i]$ and $U_L^i, U_R^i, V_L^i, V_R^i$ for $i \in [1..q]$ in terms of $S^i, T^i, S_1^i, X_L^i, X_R^i, Y_L^i, Y_R^i, Z_j^i$:

$$X_j^i = L_j'^i + Z_j^i, \tag{1}$$
$$Y_j^i = R_j^i + Z_j^i + S_1^i, . \tag{2}$$

Moreover, we define four auxiliary variables to easier refer to them throughout the proof:

$$U_L^i = L_\ell^i + X_L^i, \tag{3}$$
$$U_R^i = R_\ell^i + X_R^i, \tag{4}$$
$$V_L^i = L_\ell'^i + Y_L^i, \tag{5}$$
$$V_R^i = R_\ell'^i + Y_R^i. \tag{6}$$

IDENTIFYING A BASIS. A basis is the set of variables (internal to the constructions) which can be sampled uniformly and independently in the ideal oracles after fixing the inputs and outputs that are known to adversary. By looking at the construction and eliminating the relationships between the internal variables, plaintexts, and ciphertexts, some internal variables can be chosen almost freely, and still the real construction will behave indistinguishable from the ideal world for the adversary even after observing the plain- and ciphertexts. We call those variables a basis. For $i \in [1..q], j \in [1..\ell^i]$, we define (i, j) to be *fresh* if either of the following is true:

- $i \in E$, and for any $i' \in [1..i-1]$: $(L_j^{i'}, R_j^{i'}) \neq (L_j^i, R_j^i)$;
- $i \in D$, and for any $i' \in [1..i-1]$: $(L_j'^{i'}, R_j'^{i'}) \neq (L_j'^i, R_j'^i)$.

For $i \in [2..q], i' \in [1..i-1]$, we say i is *akin* to i' if either of the following holds:

- $\ell^i = \ell^{i'}$, $i \in E$, and for any $j \in [1..\ell^i - 1]$: $(L_j^{i'}, R_j^{i'}) = (L_j^i, R_j^i)$;
- $\ell^i = \ell^{i'}$, $i \in D$, and for any $j \in [1..\ell^i - 1]$: $(L_j'^{i'}, R_j'^{i'}) = (L_j'^i, R_j'^i)$;

We say i is *new* if it is not akin to any $i' \in [1..i-1]$. Now we define the basis as follows: for $i \in [1..q]$,

- For $j \in [1..\ell^i - 1]$, Z_j^i is in the basis if (i, j) is fresh;
- X_L^i and X_R^i are in the basis if $i \in D$, or if $i \in E$ and i is new;
- Y_L^i and Y_R^i are in the basis if $i \in E$, or if $i \in D$ and i is new;
- S^i, T^i, and S_1^i are in the basis.

Let σ_F be the total number of fresh pairs in the set $\{(i, j) \mid i \in [q], j \in [\ell^i - 1]\}$, and let q_ν be the total number of new queries in $[1..q]$. Then the size of the basis is $\sigma_F + 2q_\nu + 5q$.

Extension from Basis. Now we show how all the internal variables X_j^i, Y_j^i for $i \in [1..q], j \in [1..\ell^i]$ and $U_L^i, U_R^i, V_L^i, V_R^i$ for $i \in [1..q]$ can be written in terms of basis variables. Since we have already seen how to write them in terms of $S^i, T^i, S_1^i, X_L^i, X_R^i, Y_L^i, Y_R^i$ for $i \in [1..q]$ and Z_j^i for $i \in [1..q], j \in [1..\ell^i - 1]$, and S^i, T^i, S_1^i for $i \in [1..q]$ are already in the basis, it suffices to show that Z_j^i for $i \in [1..q], j \in [1..\ell^i - 1]$ and $X_L^i, X_R^i, Y_L^i, Y_R^i$ for $i \in [1..q]$ can be written in terms of basis variables. An expression of an internal variable in terms of basis variables and the oracle inputs and outputs will be called the *extension expression* of the basis variable. Thus, whenever we sample all the basis elements, we can extend this through these equations to assign values to all the internal variables.

For $i \in E, j \in [1..\ell^i]$, let i' be such that (i', j) is fresh, and

$$(L_j^{i'}, R_j^{i'}) = (L_j^i, R_j^i).$$

Then i' is called the *j-predecessor* of i, denoted $i : j$. Similarly, for $i \in D, j \in [1..\ell^i]$, if for some i' we have (i', j) fresh and

$$(L_j'^{i'}, R_j'^{i'}) = (L_j'^i, R_j'^i),$$

we set $i : j = i'$. (Thus, when (i, j) is fresh, $i : j$ is i itself.) For $i \in E, j \in [1..\ell^i]$ we have from (1)

$$X_j^i = X_j^{i:j} = L_j'^{i:j} + Z_j^{i:j},$$

so

$$Z_j^i = L_j'^{i:j} + L_j'^i + Z_j^{i:j}; \tag{7}$$

and for $i \in D, j \in [1..\ell^i]$ we have from (2)

$$Y_j^i = Y_j^{i:j} = R_j^{i:j} + Z_j^{i:j} + S_1^{i:j},$$

so

$$Z_j^i = R_j^{i:j} + R_j^i + Z_j^{i:j} + S_1^{i:j} + S_1^i. \tag{8}$$

Now if i and $i : j$ are both in E or both in D, $Z_j^{i:j}$ is a basis element. (In particular, when $i : j = i$, Z_j^i is a basis element.) Otherwise, we can go back one step further to $(i : j) : j$, the j-predecessor of $i : j$, denoted $i : j^2$. We call (1) and (2) the *extension equations*. They will serve useful in the later proofs. Note that it does not hold in general that $(i : j) : j = i : j$. This holds only if $i : j$ and i are both in E or both in D, or when $i : j$ points to a fresh input block.

For $i \in [2..q]$, the smallest query index in $[1..i-1]$ which i is akin to is called the *origin* of i, denoted $\bar{i}$. We also define the origin of 1 to be 1 itself. Thus, for $i \in E$,

$$X_L^i = X_L^{\bar{i}}, \tag{9}$$

$$X_R^i = X_R^{\bar{i}}; \tag{10}$$

and for $i \in D$,

$$Y_L^i = Y_L^{\bar{i}}, \tag{11}$$

$$Y_R^i = Y_R^{\bar{i}}. \tag{12}$$

Since for $i \in E$, $X_L^{\bar{i}}$ and $X_R^{\bar{i}}$ are in the basis, and for $i \in D$, $Y_L^{\bar{i}}$ and $Y_R^{\bar{i}}$ are in the basis, this completes the extensions.

Oracles and bad Events. The real oracle employs $\mathrm{ZCZ}[\widetilde{\pi}]$ to answer the queries of $\mathbf{A}$. In the ideal world, the encryption oracle samples and returns $L_j'^i, R_j'^i$ for $i \in E, j \in [1..\ell^i]$ uniformly at random; the decryption oracle samples and returns L_j^i, R_j^i for $i \in D$, $j \in [1..\ell^i]$ uniformly at random. Once the interaction phase is over, the ideal world oracle samples and returns each basis element uniformly at random from $\{0,1\}^n$, with two exceptions:

– For $i \in E$, S^i is drawn uniformly from the set

$$\{0,1\}^n \setminus \left\{ S^{i'} \mid i \text{ is akin to } i', R^i = R^{i'} \right\};$$

– For $i \in D$, T^i is drawn uniformly from the set

$$\{0,1\}^n \setminus \left\{ T^{i'} \mid i \text{ is akin to } i', L'^i = L'^{i'} \right\}.$$

The real world releases the values of the basis variables to the adversary. (Thus, from the extension equations, $\mathbf{A}$ can calculate the values of the inputs, tweaks, and outputs of all internal TBC calls.)

The task of A is to distinguish the real world $\mathcal{O}_{\mathrm{real}}$ from the ideal world $\mathcal{O}_{\mathrm{ideal}}$, given a transcript τ of its interaction with the available oracles. We call a transcript τ bad iff $\tau \in \mathrm{BadT}$, and call it good otherwise. We say that the event bad has occurred if at least one of the following events occur:

– badA occurs when:
  - For some $i \in E, j \in [1..\ell^i]$, there exists $i' \in [1..i-1]$ with $\ell^{i'} \geq j$ s.t. $(L_j'^{i'}, R_j'^{i'}) = (L_j'^i, R_j'^i)$;
  - For some $i \in D, j \in [1..\ell^i]$, there exists $i' \in [1..i-1]$ with $\ell^{i'} \geq j$ s.t. $(L_j^{i'}, R_j^{i'}) = (L_j^i, R_j^i)$;
– badB occurs when:
  - For some $i \in [2..q]$ there exists $i' \in [1..i-1]$ with $\ell^i = \ell^{i'}$ s.t. one of the following holds:
    — $(U_L^i, U_R^i) = (U_L^{i'}, U_R^{i'})$;
    — $(S^i, U_R^i) = (S^{i'}, U_R^{i'})$;
    — $(S^i, T^i) = (S^{i'}, T^{i'})$;
    — $(V_L^i, T^i) = (V_L^{i'}, T^{i'})$;
    — $(V_L^i, V_R^i) = (V_L^{i'}, V_R^{i'})$;

– badC occurs when:
  - For some $i \in [1..q]$, there exists $i' \in [1..i-1]$ s.t. $(S_1^i, T^i) = (S_1^{i'}, T^{i'})$;
  - For some $i \in [1..q], j \in [1..\ell^i - 1]$, there exists $i' \in [1..i-1]$ with $\ell^{i'} \geq j+1$ s.t. $(Z_j^i, T^i) = (Z_j^{i'}, T^{i'})$;
– badD occurs when:
  - For some $i \in E, j \in [1..\ell^i - 1]$, there exists $i' \in [1..i-1]$ with $\ell^{i'} \geq j+1$ s.t. $(L'^i_j, Y_j^i) = (L'^{i'}_j, Y_j^{i'})$;
  - For some $i \in D, j \in [1..\ell^i - 1]$, there exists $i' \in [1..i-1]$ with $\ell^{i'} \geq j+1$ s.t. $(R_j^i, X_j^i) = (R_j^{i'}, X_j^{i'})$;
– badE occurs when:
  - For some $i \in [2..q]$, there exists $i' \in [1..i-1]$ s.t. i is not akin to i' and yet one of the following holds:
    — $(X_L^{*i}, X_R^{*i}) = (X_L^{*i'}, X_R^{*i'})$;
    — $(Y_L^{*i}, Y_R^{*i}) = (Y_L^{*i'}, Y_R^{*i'})$;

Thus, $\mathsf{bad} \stackrel{\text{def}}{=} \mathsf{badA} \vee \mathsf{badB} \vee \mathsf{badC} \vee \mathsf{badD} \vee \mathsf{badE}$. Clearly, it holds that

$$\Pr[\mathsf{bad}] \leq \Pr[\mathsf{badA}] + \Pr[\mathsf{badB}] + \Pr[\mathsf{badC}] + \Pr[\mathsf{badD}] + \Pr[\mathsf{badE}]. \quad (13)$$

Then, the proof follows from Lemmas 1, 2, and 3.

Lemma 2. $\Pr[\mathsf{bad}] \leq \dfrac{3\sigma^2 + 8q^2}{N^2}$.

Proof. Below, we show that each of the collision-pairs that would result in one of the bad events has a joint probability of $\leq 1/N^2$. Clearly, we need the assumption that all basis elements are uniformly sampled from $\{0,1\}^n$ for this purpose. Moreover, the values S^i and T^i are sampled without replacement under certain circumstances, their bound is at most $1/N(N-1)$, which can be upperbounded by $1/N(N-1) < 2/N^2$. Thus, for bounding the bad events, we simply need to bound the number of candidate collision-pairs.

For badA, there can be:

– at most $\sigma_E^2/2$ collision events of the form $(L'^{i'}_j, R'^{i'}_j) = (L'^i_j, R'^i_j)$;
– at most $\sigma_D^2/2$ collision events of the form $(L_j^{i'}, R_j^{i'}) = (L_j^i, R_j^i)$;

where σ_E is the total number of encryption query blocks and σ_D is the total number of decryption query blocks, so that $\sigma_E^2 + \sigma_D^2 \leq \sigma^2$. Thus

$$\Pr[\mathsf{badA}] \leq \frac{\sigma^2}{N^2}. \quad (14)$$

For badB, there can be:

– at most $q^2/2$ collision events of the form $(U_L^i, U_R^i) = (U_L^{i'}, U_R^{i'})$;
– at most $q^2/2$ collision events of the form $(S^i, U_R^i) = (S^{i'}, U_R^{i'})$;
– at most $q^2/2$ collision events of the form $(S^i, T^i) = (S^{i'}, T^{i'})$;

– at most $q^2/2$ collision events of the form $(V_L^i, T^i) = (V_L^{i'}, T^{i'})$;
– at most $q^2/2$ collision events of the form $(V_L^i, V_R^i) = (V_L^{i'}, V_R^{i'})$;

Thus

$$\Pr[\mathsf{badB}] \leq \frac{5q^2}{N^2}. \tag{15}$$

For badC, there can be:

– at most $q^2/2$ collision events of the form $(S_1^i, T^i) = (S_1^{i'}, T^{i'})$.
– at most $\sigma^2/2$ collision events of the form $(Z_j^i, T^i) = (Z_j^{i'}, T^{i'})$;

Thus

$$\Pr[\mathsf{badC}] \leq \frac{q^2 + \sigma^2}{N^2}. \tag{16}$$

For badD, there can be:

– at most $\sigma_E^2/2$ collision events of the form $(L_j'^i, Y_j^i) = (L_j'^{i'}, Y_j^{i'})$;
– at most $\sigma_D^2/2$ collision events of the form $(R_j^i, X_j^i) = (R_j^{i'}, X_j^{i'})$.

Thus

$$\Pr[\mathsf{badD}] \leq \frac{\sigma^2}{N^2}. \tag{17}$$

For badE, there can be:

– at most $q^2/2$ collision events of the form $(X_L^{*i}, X_R^{*i}) = (X_L^{*i'}, X_R^{*i'})$;
– at most $q^2/2$ collision events of the form $(Y_L^{*i}, Y_R^{*i}) = (Y_L^{*i'}, Y_R^{*i'})$.

Thus

$$\Pr[\mathsf{badE}] \leq \frac{2q^2}{N^2}. \tag{18}$$

The lemma follows from (13)–(18).

Now all that is left to do is to establish our claim that each of the collision-pairs that would result in one of the bad events has a joint probability $\leq 1/N^2$. This is to be done by examining each bad event separately. badA, badB and badC are fairly straightforward; for the sake of completeness we include short proofs in Appendix B. badD is more interesting; we provide below a complete analysis of it. The trickiest case is badE; in this section we examine one of its subcases in detail, and the complete case-by-case analysis is included in Appendix C.

Analysis of badD. We consider the two cases separately:

– $(L_j'^i, Y_j^i) = (L_j'^{i'}, Y_j^{i'})$, $i \in E$, $i' < i$: We will show that $Y_j^i = Y_j^{i'}$ always leads to an equation containing at least one basis variable that cannot get canceled out. The required bound follows since the basis variable and $L_j'^i$ are independently sampled. From (2) we have

$$R_j^i + Z_j^i + S_1^i = R_j^{i'} + Z_j^{i'} + S_1^{i'}. \tag{19}$$

Note that S_1^i cannot occur in the expansion of $Z_j^{i:j}$, since $i \in E$. Now we have two options of i':

- $i' \in E$: From (7) and (19) we have

$$R_j^i + L_j'^{i:j} + L_j'^i + Z_j^{i:j} + S_1^i = R_j^{i'} + L_j'^{i':j} + L_j'^{i'} + Z_j^{i':j} + S_1^{i'}.$$

  Here the basis element S_1^i cannot be canceled out, since $i' < i$.

- $i' \in D$: From (7), (8) and (19), we have

$$R_j^i + L_j'^{i:j} + L_j'^i + Z_j^{i:j} + S_1^i = R_j^{i'} + R_j^{i':j} + R_j^{i'} + Z_j^{i':j} + S_1^{i':j}.$$

  Again, the basis element S_1^i cannot be canceled out since $i' : j \leq i' < i$.

– $(R_j^i, X_j^i) = (R_j^{i'}, X_j^{i'})$, $i \in D$, $i' < i$: As above, we show that $X_j^i = X_j^{i'}$ always leads to an equation containing at least one basis variable that cannot get canceled out, and the required bound follows since the basis variable and R_j^i are independently sampled. From (1), we have

$$L_j'^i + Z_j^i = L_j'^{i'} + Z_j^{i'}. \tag{20}$$

  Now, we have two options of i':

- $i' \in E$: From (8), (7) and (20), we have

$$L_j'^i + R_j^{i:j} + R_j^i + Z_j^{i:j} + S_1^{i:j} + S_1^i = L_j'^{i':j} + Z_j^{i':j}.$$

  When $i : j < i$, the basis element S_1^i cannot be canceled out, and when $i = i : j$, we have $i' : j \leq i' < i = i : j$, so the basis element $Z_j^{i:j} = Z_j^i$ cannot be canceled out.

- $i' \in D$: From (8) and (19), we have

$$L_j'^i + R_j^{i:j} + R_j^i + Z_j^{i:j} + S_1^{i:j} + S_1^i = L_j'^{i'} + R_j^{i':j} + R_j^{i'} + Z_j^{i':j} + S^{i':j} + S^{i'},$$

  Here again, either S_1^i or the basis element Z_j^i cannot be canceled out, and the argument is identical to the above.

Analysis of badE. This is trickier than the other bad events, and requires some careful case analysis. We examine the two most difficult sub-cases here; the rest can be similarly handled. Let $i' < i$ and $\ell \stackrel{\text{def}}{=} \ell^{i'} = \ell^i$, and let $\alpha_j(\cdot)$ and $\alpha_j^2(\cdot)$ be linear functions defined as

$$\alpha_j(x) \stackrel{\text{def}}{=} \alpha^{\ell-1-j} \cdot x \quad \text{and} \quad \alpha_j^2(x) \stackrel{\text{def}}{=} (\alpha^2)^{\ell-1-j} \cdot x.$$

First consider the case $i \in E, i' \in E$. From (7), (26) and (27) we have

$$\sum_{j=0}^{\ell-1} \alpha_j(Z_j^{i:j} + Z_j^{i':j}) = \sum_{j=0}^{\ell-1} \alpha_j(L_j'^{i:j} + L_j'^{i':j}), \tag{21}$$

$$\sum_{j=0}^{\ell-1} \alpha_j^2(Z_j^{i:j} + Z_j^{i':j}) = \sum_{j=0}^{\ell-1} \alpha_j^2(L_j'^{i:j} + L_j'^{i':j} + R_j^{i:j} + R_j^{i':j}). \tag{22}$$

By choice of j_0, $i:j_0 \neq i':j_0$. Suppose $i:j_0 > i':j_0$. If $i:j_0 \in D$, using (8), we replace $Z_{j_0}^{i:j_0}$ by $R_{j_0}^{i:j_0^2} + R_{j_0}^{i:j_0} + Z_{j_0}^{i:j_0^2} + S_1^{i:j_0^2} + S_1^{i:j_0}$. The basis element $S_1^{i:j_0}$ does not get canceled out; moreover, $R_{j_0}^{i:j_0}$ remains only in the top equation, while it gets canceled out in the bottom equation. Since $i:j = i':j$ for all $j > j_0$, none of the adversary-queried blocks remaining in either equation came after $R_{j_0}^{i:j_0}$, so it is independent of the rest of the equation; along with the basis element $S_1^{i:j_0}$ (which appears in both equations), this makes the two collisions independent, thus occurring jointly with a probability $1/N^2$.

If $i:j_0 \in E$, $Z_{j_0}^{i:j_0}$ is in the basis, and does not cancel out. On the right hand side of both equations, $L_{j_0}'^{i:j_0}$ remains uncanceled as well, while all later adversary queries get canceled. Thus, the two equations can become dependent with probability at most $1/N$; then, the common collision can occur with probability at most $1/N$. Thus, in either case, the joint collision can occur with a probability of more than $1/N^2$. The analysis is similar when $i:j_0 < i:j_0$; then we focus on the latter instead.

Now consider the case $i \in E, i' \in D$. From (7), (8), (26) and (27) we have

$$\sum_{j=0}^{\ell-1} \alpha_j (Z_j^{i:j} + Z_j^{i':j} + S_1^{i'} + S_1^{i':j}) = \sum_{j=0}^{\ell-1} \alpha_j (L_j'^{i:j} + L_j'^{i':j} + R_j^{i'} + R_j^{i':j}), \tag{23}$$

$$\sum_{j=0}^{\ell-1} \alpha_j^2 (Z_j^{i:j} + Z_j^{i':j} + S_1^{i'} + S_1^{i':j}) = \sum_{j=0}^{\ell-1} \alpha_j^2 (L_j'^{i:j} + L_j'^{i':j} + R_j^{i:j} + R_j^{i':j}). \tag{24}$$

By choice of j_0 and j_1, $i:j_0 \neq i'$ and $i:j_1 \neq i'$. Suppose $i:j_0 < i'$. Then $S_1^{i'}$ and $R_{j_0}^{i'}$ remain uncanceled in (30), and no adversary query block queried after $R_{j_0}^{i'}$ remains uncanceled; in (31), $S_1^{i'}$ remains uncanceled again, but there is no $R_{j_0}^{i'}$ and no adversary query block queried after it. Thus these two can occur jointly with a probability at most $1/N^2$.

A symmetric argument can be used when $i:j_0 > i'$ and $i:j_0 \in D$: we replace $Z_{j_0}^{i:j_0}$ by $R_{j_0}^{i:j_0^2} + R_{j_0}^{i:j_0} + Z_{j_0}^{i:j_0^2} + S_1^{i:j_0^2} + S_1^{i:j_0}$ using (8), and observe that $S_1^{i:j_0}$ remains uncanceled in either equation, while $R_{j_0}^{i:j_0}$ remains uncanceled in (30), but gets canceled out in (31), and no adversary query block queried after it remains in either equation.

When $i:j_0 > i'$ and $i:j_0 \in E$, but $i:j_1$ satisfied one of the above two conditions, we can argue as above using $i:j_1$ instead. If we also have $i:j_1 > i'$ and $i:j_1 \in E$, we observe that $Z_{j_0}^{i:j_0}$ and $Z_{j_1}^{i:j_1}$ are basis elements that do not get canceled out in either equation. Their combined contribution to the left-hand side of (30) is $\alpha^{\ell-1-j_0} \cdot Z_{j_0}^{i:j_0} + \alpha^{\ell-1-j_1} \cdot Z_{j_1}^{i:j_1}$ and to the left-hand side of (31) is $(\alpha^2)^{\ell-1-j_0} \cdot Z_{j_0}^{i:j_0} + (\alpha^2)^{\ell-1-j_1} \cdot Z_{j_1}^{i:j_1}$. These two collisions are independent since $\alpha^{\ell-1-j_0} \cdot (\alpha^2)^{\ell-1-j_1} \neq \alpha^{\ell-1-j_1} \cdot (\alpha^2)^{\ell-1-j_0}$, and thus can occur with a probability at most $1/N^2$. (The entire subcase-tree analysis can be found in Appendix C.)

This completes the proof of Lemma 2.

Lemma 3. It holds that

$$\frac{\Pr\left[\Theta_{\text{real}} = \tau\right]}{\Pr\left[\Theta_{\text{ideal}} = \tau\right]} \geq 1 - \frac{q^2}{N^2}.$$

Proof. Let τ be a good transcript, i. e., none of the events badA, badB, badC, badD, or badE occurred. Then, in the ideal world, there are 2σ samplings for generating the query responses and $\sigma_F + 2q_\nu + 5q$ for generating the basis elements. In the ideal world, the basis elements are sampled uniformly at random and independently from each other. Hence, the probability for those is given by

$$\frac{1}{N^{\sigma_F + 2q_\nu + 5q}}.$$

The situation differs for the outputs of the scheme. The ideal world is an ideal SPRP; hence, the outputs are sampled without replacement. Since all queries are from the domain $\mathcal{B}^{\leq n} = \bigcup_{i=1}^{n} \mathcal{B}^i$, we can group the encryption and decryption queries into disjoint sets $\mathcal{L}^1, \ldots, \mathcal{L}^n$ such that their union contains all queries, and Set $\mathcal{L}^i$ contains exactly the queries of length i di-blocks. We define by $\textsc{Load}\left(\mathcal{L}^i\right)$ the number of queries in set $\mathcal{L}^i$, for all $1 \leq i \leq n$. The probability for ciphertext outputs from encryption queries and plaintext outputs from decryption queries is then

$$\prod_{i=1}^{n} \frac{1}{\left(N^{2i}\right)_{\textsc{Load}(\mathcal{L}^i)}}.$$

Since the size of all queries is at least $2n$ bits, we can lower bound the probability by

$$\prod_{i=1}^{n} \frac{1}{\left(N^{2i}\right)_{\textsc{Load}(\mathcal{L}^i)}} \leq \frac{1}{\left(N^2\right)_{2q}} \cdot \frac{1}{N^{2\sigma - 2q}}.$$

We obtain that

$$\Pr\left[\Theta_{\text{ideal}} = \tau\right] \leq \frac{1}{N^{\sigma_F + 2q_\nu + 3q + 2\sigma}} \cdot \frac{1}{\left(N^2\right)_q}. \tag{25}$$

In the real world, the real construction employs a permutation $\widetilde{\pi}^{\mathsf{T}}(\cdot)$ for each tweak $\mathsf{T} \in \mathcal{T}_{\mathcal{D}\times\mathcal{I}}$ that was used in the transcript, . We write the set of all occurred tweaks of all di-blocks of all queries in the transcript and write it as $\left\{\mathsf{T}^1, \ldots, \mathsf{T}^\theta\right\}$. We further define by $\textsc{Load}\left(\mathsf{T}\right)$ the load of a tweak T, i.e., the number of distinct inputs that were used for it over all queries and di-blocks of the transcript. It holds that

$$\sum_{i=1}^{\theta} \textsc{Load}\left(\mathsf{T}^i\right) = \sigma_F + 2\sigma + 2q_\nu + 5q$$

We adopt the notion of transcript-compatible permutations from [13]. We call $\widetilde{\pi}$ *compatible* with τ if for all queries, $\widetilde{\pi}$ produced all intermediate variables as

well as all outputs in τ. Let $\mathsf{Comp}(\tau)$ denote the set of tweakable permutations $\widetilde{\pi}$ that are compatible with τ. Thus

$$\Pr\left[\Theta_{\text{real}} = \tau\right] = \Pr\left[\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}\left(\mathcal{T}_{D,I}, \{0,1\}^n\right) : \widetilde{\pi} \in \mathsf{Comp}(\tau)\right].$$

For a fixed tweak T, the fraction of compatible permutations is given by

$$\prod_{i=0}^{\textsc{Load}(\mathsf{T})-1} \frac{1}{N-i} = \frac{1}{(N)_{\textsc{Load}(\mathsf{T})}}.$$

Over all tweaks T^i, for $1 \le i \le \theta$, the fraction of compatible permutations is given by

$$\prod_{i=1}^{\theta} \frac{1}{(N)_{\textsc{Load}(\mathsf{T}^i)}}$$

It is hard to work with this probability directly. Instead, since we are interested in a lower bound for the real-world probability of transcripts, we can lower bound the probability of all $\sigma_F + 2q_\nu + 5q$ basis variables by the naive probability that they are all computed from fresh tweaks:

$$\frac{1}{N^{\sigma_F+2q_\nu+5q}}.$$

For the ciphertext and plaintext outputs, we can employ similar sets $\mathcal{L}^i$, for $1 \le i \le n$, as we had for the ideal world, where Set $\mathcal{L}^i$ again consists of all queries of length i di-blocks. The probability of query outputs in the real world can then be lower bounded by the

$$\prod_{i=1}^{n} \frac{1}{(N^{2i})_{\textsc{Load}(\mathcal{L}^i)}}.$$

Now, we can upper bound the ratio of the probability of our transcripts by

$$\begin{aligned}
\frac{\Pr\left[\Theta_{\text{real}} = \tau\right]}{\Pr\left[\Theta_{\text{ideal}} = \tau\right]} &\ge \frac{\frac{1}{N^{\sigma_F+2q_\nu+5q}} \cdot \prod_{i=1}^{n} \frac{1}{(N^{2i})_{\textsc{Load}(\mathcal{L}^i)}}}{\frac{1}{N^{\sigma_F+2q_\nu+5q}} \cdot \frac{1}{N^{2\sigma-2q}} \cdot \frac{1}{(N^2)_q}} \\
&\ge \frac{\prod_{i=1}^{n} \frac{1}{(N^{2i})_{\textsc{Load}(\mathcal{L}^i)}}}{\frac{1}{(N^2)_q} \cdot \frac{1}{N^{2\sigma-2q}}} \\
&\ge \frac{\left(N^2\right)_q \cdot N^{2\sigma-2q}}{N^{2\sigma}} = \frac{\left(N^2\right)_q}{(N^2)^q} \\
&= \frac{(N^2)(N^2-1)\cdot\dots\cdot(N^2-q+1)}{(N^2)^q} \ge \left(\frac{N^2-q+1}{N^2}\right)^q \\
&\ge \left(\frac{N^2-q}{N^2}\right)^q = \left(1-\frac{q}{N^2}\right)^q
\end{aligned}$$

$$\overset{\text{Bernoulli}}{\geq} 1 - \frac{q^2}{N^2},$$

where the last inequality is Bernoulli's. So, we obtain our claim in Lemma 3.

6.2 Proof Sketch for Messages with Arbitrary Number of Complete Di-blocks

Theorem 2. Let $\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}(\mathcal{T}_{D,I}, \{0,1\}^n)$. Let $\mathbf{A}$ be an SPRP adversary on ZCZ$[\widetilde{\pi}]$ that asks at most q queries queries of domain $\mathcal{B}^+$, whose lengths sum up to at most σ di-blocks in total, and $\mathbf{A}$ runs in time at most TIME. Then

$$\mathbf{Adv}^{\text{SPRP}}_{\text{ZCZ}[\widetilde{\pi}]}(\mathbf{A}) \leq \frac{4\sigma^2 + 8q^2}{N^2}.$$

Proof Sketch. The proof follows a similar strategy as that of Theorem 1. So, we only consider the equations in the analysis of bad events that differ. We add each $S^i_k, i \in [1..q], k \in \left[1..\left\lceil \ell^i/n \right\rceil\right]$ to the basis. The ideal oracle samples the additional basis elements along with the original basis elements in the second step, and the definitions of the bad cases do not change. From the equations (1)–(6) that we began with, only (2) is now replaced by

$$Y^i_j = R^i_j + Z^i_j + S^i_{\lceil j/n \rceil}. \tag{2'}$$

In the extension equations, this changes only (8), which is replaced by

$$Z^i_j = R^{i:j}_j + R^i_j + Z^{i:j}_j + S^{i:j}_{\lceil j/n \rceil} + S^i_{\lceil j/n \rceil}. \tag{8'}$$

The definitions of the bad cases remain the same except badC, which now occurs when:

- For some $i \in [1..q], k \in \left[1..\left\lceil \ell^i/n \right\rceil\right]$, there exists $i' \in [1..i-1]$ with $\ell^{i'} \geq n(k-1)$ s.t. $(S^i_k, T^i) = (S^{i'}_k, T^{i'})$;
- For some $i \in [1..q], j \in [1..\ell^i - 1]$, there exists $i' \in [1..i-1]$ with $\ell^{i'} \geq j+1$ s.t. $(Z^i_{k,c}, T^i) = (Z^{i'}_{k,c}, T^{i'})$, where $k = \lceil j/n \rceil, c = j - n(k-1)$.

Of these, the counting does not change for the latter; for the former, there are now at most $c_{\max} q^2/2$ possible collision pairs now, where $c_{\max}$ is the maximum number of chunks in one query; we generously bound this by $\sigma^2/2$. This adds $(\sigma^2 - q^2)/2N$ to our earlier bound, to obtain the new bound for the extended version. To ensure that the counting argument for badE still goes through, we only note that for $k \in [1..\lceil \ell/n \rceil]$, S^i_k can only occur in any of the collision equations from badE with coefficients $\beta^{\ell-1-j}$ for $j \in [n(k-1)+1..nk]$, where β is either α or α^2, and for any choice of k, a non-empty subset of these coefficients cannot add to 0.

6.3 Proof Sketch for the Security of ZCZ*

Theorem 3. Let $\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}(\mathcal{T}_{D,I}, \{0,1\}^n)$. Let $\mathbf{A}$ be an SPRP adversary on $\mathrm{ZCZ}^*[\widetilde{\pi}]$ that asks at most q queries of domain $\{0,1\}^{\geq 2n}$, whose lengths sum up to at most σ di-blocks in total, q' of which contains an incomplete di-block at the end. Then

$$\mathbf{Adv}^{\mathrm{SPRP}}_{\mathrm{ZCZ}^*[\widetilde{\pi}]}(\mathbf{A}) \leq \frac{4\sigma^2 + 8q^2 + 9q'^2}{N^2}.$$

Proof Sketch. The ideal oracle's sampling mechanism for the tweakable blockcipher outputs for the partial di-block messages is slightly trickier. Let $\mathcal{I}$ denote the indices of the queries with incomplete di-blocks. Instead of simulating an ideal permutation, the ideal oracle simulates what [22] calls an $\pm\tilde{\mathbf{rnd}}$ oracle, which always returns random bits, as long as no pointless queries are asked. (It is easy to argue for our construction why not permitting pointless queries does not diminish the adversary's power, so we can confine our attention to the no-pointless-query scenario.)

We use the notation $(U,V), (U_m, V_m), (U', V')$ for the outputs of the blockcipher calls in the top, middle, and bottom layers respectively. M_j denotes (L_j, R_j), and $*$ denotes the index of the incomplete di-block.

- For the smallest $i \in \mathcal{I}$, $U^i_*, V^i_*, U'^i_*, V^i_*$ are sampled uniformly from $\{0,1\}^n$;
- For each i in $\mathcal{I}$ such that for no i' in $\mathcal{I}$ with $i' < i$ we have $(L^i_*, R^i_*) \neq (L^{i'}_*, R^{i'}_*)$:
  - U^i_* is sampled uniformly from $\{0,1\}^n \setminus \left\{U^{i'}_* \mid i' \in \mathcal{I}, i' < i\right\}$;
  - V^i_* is sampled uniformly from $\{0,1\}^n \setminus \left\{V^{i'}_* \mid i' \in \mathcal{I}, i' < i\right\}$;
- For each i in $\mathcal{I}$ such that for no i' in $\mathcal{I}$ with $i' < i$ we have $(L'^i_*, R'^i_*) \neq (L'^{i'}_*, R'^{i'}_*)$:
  - U'^i_* is sampled uniformly from $\{0,1\}^n \setminus \left\{U'^{i'}_* \mid i' \in \mathcal{I}, i' < i\right\}$;
  - V'^i_* is sampled uniformly from $\{0,1\}^n \setminus \left\{V'^{i'}_* \mid i' \in \mathcal{I}, i' < i\right\}$;
- For each $i \in \mathcal{I}$ the $(2n-s)$-bit suffix R^i of (U^i_{m*}, V^i_{m*}) is sampled uniformly from $\{0,1\}^{2n-s}$, and (U^i_{m*}, V^i_{m*}) is set to $(M^i_* + M'^i_*)||R^i$.

The new bad cases are:

- For some distinct i, i' in $\mathcal{I}$ with $\ell^i = \ell^{i'} = \ell$ we have
$$(M^i_{1..\ell-1}, M^i_\ell + (U^i_*, V^i_*)) = (M^{i'}_{1..\ell-1}, M^{i'}_\ell + (U^{i'}_*, V^{i'}_*));$$
- For some distinct i, i' in $\mathcal{I}$ with $\ell^i = \ell^{i'} = \ell$ we have
$$(M'^i_{1..\ell-1}, M'^i_\ell + (U^i_*, V^i_*)) = (M'^{i'}_{1..\ell-1}, M'^{i'}_\ell + (U^{i'}_*, V^{i'}_*));$$
- For some distinct i, i' in $\mathcal{I}$ with $\ell^i = \ell^{i'} = \ell$ we have
$$\begin{aligned}&(L^i_\ell + L'^i_\ell + U^i_* + U'^i_*, R^i_\ell + R'^i_\ell + V^i_* + V'^i_*)\\ &= (L^{i'}_\ell + L'^{i'}_\ell + U^{i'}_* + U'^{i'}_*, R^{i'}_\ell + R'^{i'}_\ell + V^{i'}_* + V'^{i'}_*);\end{aligned}$$

– For some distinct i, i' in $\mathcal{I}$ with $\ell^i = \ell^{i'} = \ell$ we have

$$(R^i_\ell + R'^i_\ell + V^i_* + V'^i_*, U^i_{m*}) = (R^{i'}_\ell + R'^{i'}_\ell + V^{i'}_* + V'^{i'}_*, U^{i'}_{m*});$$

– For some distinct i, i' in $\mathcal{I}$ with $\ell^i = \ell^{i'} = \ell$ we have

$$(R^i_\ell + R'^i_\ell + V^i_* + V'^i_*, V^i_{m*}) = (R^{i'}_\ell + R'^{i'}_\ell + V^{i'}_* + V'^{i'}_*, V^{i'}_{m*}).$$

The probabilities of these bad cases can be bounded by $q'^2/2N'^2$, $q'^2/2N'^2$, $q'^2/2N'^2$, $q'^2/2NN'$, $q'^2/2NN'$ in that order, where $N' = N - q'$. With the reasonable assumption that $q' \le N/2$, we can replace N' with $N/2$ in these bounds and have them sum to $8q'^2/N^2$, which is our bound for the combined probability of the new bad cases. The theorem follows from Theorem 2 and Lemma 6 of [22].

Our results in Theorems 1 and 3 had considered the instantiation with an ideal random tweaked permutation $\widetilde{\pi} \leftarrow \widetilde{\mathsf{Perm}}(\mathcal{T}_{I,D}, \{0,1\}^n)$. Corollaries 1 and 3 yield the resulting security bounds when ZCZ and ZCZ* are instantiated with a given tweakable block cipher.

Corollary 1. Let $\widetilde{E}_K : \mathcal{K} \times \mathcal{T}_{I,D} \times \{0,1\}^n \to \{0,1\}^n$ be a tweakable block cipher with $K \leftarrow \mathcal{K}$. Let $\mathbf{A}$ be an SPRP adversary on $\mathrm{ZCZ}[\widetilde{E}_K]$, s.t. $\mathbf{A}$ asks at most q queries of domain $\mathcal{B}^{\le n}$, that sum up to at most σ di-blocks in total, and $\mathbf{A}$ runs in time at most TIME. Then

$$\mathbf{Adv}^{\mathrm{SPRP}}_{\mathrm{ZCZ}[\widetilde{E}_K]}(\mathbf{A}) \le \frac{3\sigma^2 + 10q^2}{2N^2} + \mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{E}_K, \widetilde{E}_K^{-1}}(\mathbf{A}'),$$

where $\mathbf{A}'$ is an STPRP adversary against $\widetilde{E}_K$ that asks at most $a' = 3\sigma + \lceil \sigma/n \rceil + 6q$ queries and runs in time at most TIME $+ O(a')$.

Corollary 2. Let $\widetilde{E}_K : \mathcal{K} \times \mathcal{T}_{I,D} \times \{0,1\}^n \to \{0,1\}^n$ be a tweakable block cipher with $K \leftarrow \mathcal{K}$. Let $\mathbf{A}$ be an SPRP adversary on $\mathrm{ZCZ}^*[\widetilde{E}_K]$ that asks at most q queries of domain $\{0,1\}^{\ge 2n}$, whose lengths sum up to at most σ di-blocks in total, q' of which contains an incomplete di-block at the end, and $\mathbf{A}$ runs in time at most TIME. Then

$$\mathbf{Adv}^{\mathrm{SPRP}}_{\mathrm{ZCZ}^*[\widetilde{E}_K]}(\mathbf{A}) \le \frac{4\sigma^2 + 8q^2 + 9q'^2}{N^2} + \mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{E}_K, \widetilde{E}_K^{-1}}(\mathbf{A}'),$$

where $\mathbf{A}'$ is an STPRP adversary against $\widetilde{E}_K$ that asks at most $a' = 3\sigma + \lceil \sigma/n \rceil + 6q + 6q'$ queries and runs in time at most TIME $+ O(a')$.

References

1. 1619.2-2010, I.S.: IEEE Standard for Wide-Block Encryption for Shared Storage Media. IEEE Std 1619.2-2010 pp. 1–91 (March 2011). https://doi.org/10.1109/IEEESTD.2011.5729263

2. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS. In: Robshaw, M., Katz, J. (eds.) CRYPTO II. LNCS, vol. 9815, pp. 123–153. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_5, full version at https://eprint.iacr.org/2016/660.pdf
3. Bellare, M., Micciancio, D.: A New Paradigm for Collision-Free Hashing: Incrementality at Reduced Cost. In: Fumy, W. (ed.) EUROCRYPT. LNCS, vol. 1233, pp. 163–192. Springer (1997). https://doi.org/10.1007/3-540-69053-0_13, https://doi.org/10.1007/3-540-69053-0_13
4. Bernstein, D.J.: Some Challenges in Heavyweight Cipher Design. Tech. rep. (January 11 2016), https://cr.yp.to/talks/2016.01.15/slides-djb-20160115-a4.pdf
5. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V., Keer, R.V.: Using Keccak technology for AE: Ketje, Keyak and more (August 22 2014), sHA-3 2014 Workshop, UC Santa Barbara
6. Bhaumik, R., Nandi, M.: An Inverse-free Single Keyed Tweakable Enciphering Scheme. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT, Part II. LNCS, vol. 9453, pp. 159–180. Springer (2015). https://doi.org/10.1007/978-3-662-48800-3_7, https://doi.org/10.1007/978-3-662-48800-3_7
7. Biryukov, A., Daemen, J., Lucks, S., Vaudenay, S.: Topics and Research Directions for Symmetric Cryptography. In: Early Symmetric Crypto Workshop. vol. 2017 (2017), https://www.cryptolux.org/mediawiki-esc2017/images/9/9a/ASJS-Topics_SymCrypto-ESC17.pdf
8. Chakraborty, D., Ghosh, S., Lopez, C.M., Sarkar, P.: FAST: A New Family of Secure and Efficient Tweakable Enciphering Schemes. Cryptology ePrint Archive, Report 2017/849 (2017), http://eprint.iacr.org/2017/849
9. Chakraborty, D., Mancillas-López, C., Rodríguez-Henríquez, F., Sarkar, P.: Efficient Hardware Implementations of BRW Polynomials and Tweakable Enciphering Schemes. IEEE Trans. Computers **62**(2), 279–294 (2013). https://doi.org/10.1109/TC.2011.227, http://doi.ieeecomputersociety.org/10.1109/TC.2011.227
10. Chakraborty, D., Mancillas-López, C., Sarkar, P.: STES: A Stream Cipher Based Low Cost Scheme for Securing Stored Data. IACR Cryptology ePrint Archive **2013**, 347 (2013), http://eprint.iacr.org/2013/347
11. Chakraborty, D., Sarkar, P.: A New Mode of Encryption Providing a Tweakable Strong Pseudo-Random Permutation. In: Robshaw, M.J.B. (ed.) FSE. LNCS, vol. 4047, pp. 293–309. Springer (2006). https://doi.org/10.1007/11799313_19, http://dx.doi.org/10.1007/11799313_19
12. Chakraborty, D., Sarkar, P.: HCH: A New Tweakable Enciphering Scheme Using the Hash-Counter-Hash Approach. IEEE Transactions on Information Theory **54**(4), 1683–1699 (2008). https://doi.org/10.1109/TIT.2008.917623, http://dx.doi.org/10.1109/TIT.2008.917623
13. Chen, S., Steinberger, J.P.: Tight Security Bounds for Key-Alternating Ciphers. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT. LNCS, vol. 8441, pp. 327–350. Springer (2014). https://doi.org/10.1007/978-3-642-55220-5_19
14. Chen, Y.L., Luykx, A., Mennink, B., Preneel, B.: Efficient Length Doubling From Tweakable Block Ciphers. IACR Trans. Symmetric Cryptol. **2017**(3), 253–270 (2017). https://doi.org/10.13154/tosc.v2017.i3.253-270
15. Cogliati, B., Lee, J., Seurin, Y.: New Constructions of MACs from (Tweakable) Block Ciphers. In: IACR Transactions on Symmetric Cryptology. vol. 2017, pp. 27–58 (2017). https://doi.org/10.13154/tosc.v2017.i2.27-58

16. Coron, J., Dodis, Y., Mandal, A., Seurin, Y.: A Domain Extender for the Ideal Cipher. In: Micciancio, D. (ed.) TCC. LNCS, vol. 5978, pp. 273–289. Springer (2010). https://doi.org/10.1007/978-3-642-11799-2_36
17. García, N.I.G.T., Chakraborty, D.: Efficient Software Implementations of Disk Encryption Schemes Using AES-NI Support. Master's thesis, Unidad Zacatenco, Departamento de Computacion (Feb 2012)
18. Granger, R., Jovanovic, P., Mennink, B., Neves, S.: Improved Masking for Tweakable Blockciphers with Applications to Authenticated Encryption. In: Fischlin, M., Coron, J. (eds.) EUROCRYPT (1). LNCS, vol. 9665, pp. 263–293. Springer (2016), full version at https://eprint.iacr.org/2015/999.pdf
19. Gueron, S., Mouha, N.: Simpira v2: A Family of Efficient Permutations Using the AES Round Function. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT, Part I. LNCS, vol. 10031, pp. 95–125 (2016). https://doi.org/10.1007/978-3-662-53887-6_4
20. Halevi, S.: EME*: Extending EME to Handle Arbitrary-Length Messages with Associated Data. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT. LNCS, vol. 3348, pp. 315–327. Springer (2004). https://doi.org/10.1007/978-3-540-30556-9_25, https://doi.org/10.1007/978-3-540-30556-9_25
21. Halevi, S.: Invertible Universal Hashing and the TET Encryption Mode. In: Menezes, A. (ed.) CRYPTO. LNCS, vol. 4622, pp. 412–429. Springer (2007). https://doi.org/10.1007/978-3-540-74143-5_23, http://dx.doi.org/10.1007/978-3-540-74143-5_23
22. Halevi, S., Rogaway, P.: A Tweakable Enciphering Mode. In: Boneh, D. (ed.) CRYPTO. LNCS, vol. 2729, pp. 482–499. Springer (2003). https://doi.org/10.1007/978-3-540-45146-4_28
23. Halevi, S., Rogaway, P.: A Parallelizable Enciphering Mode. In: Okamoto, T. (ed.) CT-RSA. LNCS, vol. 2964, pp. 292–304. Springer (2004)
24. Hoang, V.T., Krovetz, T., Rogaway, P.: AEZ v5. http://competitions.cr.yp.to/caesar-submissions.html (2014)
25. Hoang, V.T., Krovetz, T., Rogaway, P.: Robust Authenticated-Encryption AEZ and the Problem That It Solves. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT (1). LNCS, vol. 9056, pp. 15–44. Springer (2015). https://doi.org/10.1007/978-3-662-46800-5_2
26. Iwata, T.: New Blockcipher Modes of Operation with Beyond the Birthday Bound Security. In: Robshaw, M.J.B. (ed.) FSE. LNCS, vol. 4047, pp. 310–327. Springer (2006). https://doi.org/10.1007/11799313_20
27. Iwata, T., Minematsu, K.: Stronger Security Variants of GCM-SIV. IACR Trans. Symmetric Cryptol. **2016**(1), 134–157 (2016). https://doi.org/10.13154/tosc.v2016.i1.134-157
28. Iwata, T., Minematsu, K., Peyrin, T., Seurin, Y.: ZMAC: A Fast Tweakable Block Cipher Mode for Highly Secure Message Authentication. In: Katz, J., Shacham, H. (eds.) CRYPTO, Part III. LNCS, vol. 10403, pp. 34–65. Springer (2017). https://doi.org/10.1007/978-3-319-63697-9_2, full version at https://eprint.iacr.org/2017/535
29. Jean, J., Nikolic, I., Peyrin, T.: Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT (2). LNCS, vol. 8874, pp. 274–288 (2014). https://doi.org/10.1007/978-3-662-45608-8_15
30. Jean, J., Nikolić, I., Peyrin, T.: Deoxys v1.41 (2016), third-round submission to the CAESAR competition. https://competitions.cr.yp.to/round3/deoxysv141.pdf

31. Liskov, M., Rivest, R.L., Wagner, D.: Tweakable Block Ciphers. In: Yung, M. (ed.) CRYPTO. LNCS, vol. 2442, pp. 31–46. Springer (2002). https://doi.org/10.1007/s00145-010-9073-y
32. Luykx, A., Preneel, B., Tischhauser, E., Yasuda, K.: A MAC Mode for Lightweight Block Ciphers. In: Peyrin, T. (ed.) FSE. LNCS, vol. 9783, pp. 43–59. Springer (2016). https://doi.org/10.1007/978-3-662-52993-5_3
33. McGrew, D.A., Fluhrer, S.R.: The Extended Codebook (XCB) Mode of Operation. IACR Cryptology ePrint Archive **2004**, 278 (2004), https://eprint.iacr.org/2004/278
34. McGrew, D.A., Fluhrer, S.R.: The Security of the Extended Codebook (XCB) Mode of Operation. In: Adams, C.M., Miri, A., Wiener, M.J. (eds.) SAC. LNCS, vol. 4876, pp. 311–327. Springer (2007). https://doi.org/10.1007/978-3-540-77360-3_20, https://doi.org/10.1007/978-3-540-77360-3_20
35. Minematsu, K.: Beyond-Birthday-Bound Security Based on Tweakable Block Cipher. In: Dunkelman, O. (ed.) FSE. LNCS, vol. 5665, pp. 308–326. Springer (2009), https://doi.org/10.1007/978-3-642-03317-9_19
36. Minematsu, K.: Parallelizable Rate-1 Authenticated Encryption from Pseudorandom Functions. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT. LNCS, vol. 8441, pp. 275–292. Springer (2014), https://doi.org/10.1007/978-3-642-55220-5_16
37. Minematsu, K.: Building blockcipher from small-block tweakable blockcipher. Designs, Code and Cryptography **74**(3), 645–663 (2015). https://doi.org/10.1007/s10623-013-9882-8
38. Minematsu, K., Iwata, T.: Building Blockcipher from Tweakable Blockcipher: Extending FSE 2009 Proposal. In: Chen, L. (ed.) IMACC. pp. 391–412 (2011). https://doi.org/10.1007/978-3-642-25516-8_24
39. Minematsu, K., Matsushima, T.: Tweakable Enciphering Schemes from Hash-Sum-Expansion. In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) INDOCRYPT. LNCS, vol. 4859, pp. 252–267. Springer (2007). https://doi.org/10.1007/978-3-540-77026-8_19, https://doi.org/10.1007/978-3-540-77026-8_19
40. Naito, Y.: Full PRF-Secure Message Authentication Code Based on Tweakable Block Cipher. In: Au, M.H., Miyaji, A. (eds.) ProvSec. LNCS, vol. 9451, pp. 167–182. Springer (2015), https://link.springer.com/chapter/10.1007/978-3-319-26059-4_9
41. Nandi, M.: Improving upon HCTR and Matching Attacks for Hash-Counter-Hash Approach. IACR Cryptology ePrint Archive **2008**, 90 (2008), http://eprint.iacr.org/2008/090
42. Nandi, M.: A Generic Method to Extend Message Space of a Strong Pseudorandom Permutation. Computación y Sistemas **12**(3) (2009), http://cys.cic.ipn.mx/ojs/index.php/CyS/article/view/1204, full version at https://eprint.iacr.org/2008/091.pdf
43. Nandi, M.: On the Optimality of Non-Linear Computations of Length-Preserving Encryption Schemes. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT (II). LNCS, vol. 9453, pp. 113–133. Springer (2015). https://doi.org/10.1007/978-3-662-48800-3_5
44. Naor, M., Reingold, O.: A Pseudo-Random Encryption Mode (1997), manuscript available from www.wisdom.weizmann.ac.il/~naor
45. Naor, M., Reingold, O.: On the Construction of Pseudorandom Permutations: Luby-Rackoff Revisited. J. Cryptology **12**(1), 29–66 (1999). https://doi.org/10.1007/PL00003817, http://dx.doi.org/10.1007/PL00003817

46. Patarin, J.: The "Coefficients H" Technique. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC. LNCS, vol. 5381, pp. 328–345. Springer (2008). https://doi.org/10.1007/978-3-642-04159-4_21
47. Peyrin, T., Seurin, Y.: Counter-in-Tweak: Authenticated Encryption Modes for Tweakable Block Ciphers. In: Robshaw, M., Katz, J. (eds.) CRYPTO I. LNCS, vol. 9814, pp. 33–63. Springer (2016). https://doi.org/10.1007/978-3-662-53018-4_2
48. Rogaway, P.: Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In: ASIACRYPT. LNCS, vol. 3329, pp. 16–31. Springer (2004). https://doi.org/10.1007/978-3-540-30539-2_2
49. Rogaway, P., Zhang, Y.: Onion-AE: Foundations of Nested Encryption. PoPETs **2018**(2), 85–104 (2018). https://doi.org/10.1515/popets-2018-0014
50. Sarkar, P.: Improving Upon the TET Mode of Operation. In: Nam, K., Rhee, G. (eds.) ICISC. LNCS, vol. 4817, pp. 180–192. Springer (2007)
51. Sarkar, P.: Efficient Tweakable Enciphering Schemes from (Block-Wise) Universal Hash Functions. IEEE Trans. on Inf. Theory **55**(10), 4749–4760 (2009). https://doi.org/10.1109/TIT.2009.2027487, http://dx.doi.org/10.1109/TIT.2009.2027487
52. Sarkar, P.: Tweakable Enciphering Schemes Using Only the Encryption Function of a Block Cipher. Inf. Process. Lett. **111**(19), 945–955 (2011). https://doi.org/10.1016/j.ipl.2011.06.014, http://dx.doi.org/10.1016/j.ipl.2011.06.014
53. Shrimpton, T., Terashima, R.S.: A Modular Framework for Building Variable-Input-Length Tweakable Ciphers. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT (1). LNCS, vol. 8269, pp. 405–423. Springer (2013), https://doi.org/10.1007/978-3-642-42033-7_21
54. Wang, P., Feng, D., Wu, W.: HCTR: A Variable-Input-Length Enciphering Mode. In: Feng, D., Lin, D., Yung, M. (eds.) CISC. LNCS, vol. 3822, pp. 175–188. Springer (2005)
55. Yasuda, K.: A New Variant of PMAC: Beyond the Birthday Bound. In: Rogaway, P. (ed.) CRYPTO. LNCS, vol. 6841, pp. 596–609. Springer (2011). https://doi.org/10.1007/978-3-642-22792-9_34
56. Zhang, L., Wu, W., Sui, H., Wang, P.: 3kf9: Enhancing 3GPP-MAC beyond the Birthday Bound. In: Wang, X., Sako, K. (eds.) ASIACRYPT. LNCS, vol. 7658, pp. 296–312. Springer (2012). https://doi.org/10.1007/978-3-642-34961-4_19

A Related Work

This section briefly revisits existing SPRPs that we categorized into (1) Generalized Feistel networks, (2) Encrypt-Mix-Encrypt, (3) Hash-ECB-Hash, (4) Hash-Counter-Hash, and (5) miscellaneous designs.

The original Naor-Reingold construction [45] is a four-round Feistel network with two layers of an ϵ-AXU family of hash functions that wrap two layers of encryption, which could be called the starting point of wide-block ciphers. Schemes based on the Encrypt-Mix-Encrypt paradigm (also Encrypt-Mask-Encrypt) combine two wrapping layers of encryption with an intermediate layer of linear mixing. There are at least six such constructions: CMC [22], EME and EME$^+$ [23], EME* [20], AEZ-Core [24, 25], and FMix [6]. Among them, EME* has become

part of the P1619.2 standard for Wide-Block Encryption for Shared Storage Media [1]. The length doubler LDT by Chen et al. [14] could be also classified into this category, and could potentially provide even security beyond the birthday bound; however, its domain and range are limited to n up to $2n - 1$ bits.

Hash-ECB-Hash constructions consist of an encryption layer sandwiched by two hashing layers at top and bottom, a design principle that originated by Naor and Reingold [44]. Further examples include PEP [11], TET [21], or HEH [50].

Hash-Counter-Hash constructions resemble an unbalanced three-round Feistel network of two universal hash functions that wrap an encryption layer of counter mode. The first such construction was XCBv1 by McGrew and Viega that was later adapted, proven, and, as XCBv3, has become patented and part of the IEEE P1619.2 standard [1, 33, 34]. The name of the approach stems from HCTR by Wang et al. [54]. More constructions in this category include, e.g., HCH [12], HMC [41] (Hash Modified Counter), the LargeBlock1/2 constructions [38] and HSE [39]. In an ongoing series of works, Sarkar et al. [9, 17, 51] have been proposing various further versions, such as an improved HEH, and similar schemes with OFB (HOH), and counter mode (HMCH), respectively. In TES and its extensions STES and FAST [8, 10, 52], the authors later eliminated the need for the inverse operation of the block cipher.

A recent design that does not fully fit in the former categories is e.g., the two instantiations of Protected IV [53]. Protected IV is a modular framework that resembles the Ψ_3 construction by Coron et al. [16], i.e., a three-round unbalanced Feistel-like network based on tweakable ciphers. Its authors propsosed two instances of PIV, coined TCT_1 and TCT_2; the former with birthday-bound and the latter with $2n/3$-bit security, which stems from the use of a two-round Even-Mansour primitive.

The recent proposals Mr. Monster Burrito [5], HHFHFH [4], and Simpira (v2) [19] can be classified again as Feistel networks. Mr. Monster Burrito and HHFHFH are four-round unbalanced Feistel networks that were coined them heavyweight ciphers with high security guarantees due to the use of large-block permutations as primitives. Simpira is a family of variable-length ciphers based on the round function of the AES, where, family means that the Feistel design differs for small input sizes, and becomes general for inputs of eight blocks and above. In contrast to most designs, Simpira is based on the wide-trail heuristic where the authors bounded the number of rounds necessary to guarantee at least 25 active S-boxes and used three times this number of rounds.

B Analyses of badA, badB, badC

Analysis of badA. For $i \in E, j \in [1..\ell^i]$, L'^i_j and R'^i_j are sampled uniformly and independently from $\{0,1\}^n$. Thus, any collision over them occurs with a probability of $1/N$, and any disjoint pair of such collisions, being independent, jointly occurs with a probability $1/N^2$. The same reasoning holds for L^i_j and R^i_j, $i \in D, j \in [1..\ell^i]$.

Analysis of badB. Using (3)–(6), we can rewrite these collisions as:

- $(L_\ell^i + X_L^i, R_\ell^i + X_R^i) = (L_\ell^{i'} + X_L^{i'}, R_\ell^{i'} + X_R^{i'})$;
- $(S^i, R_\ell^i + X_R^i) = (S^{i'}, R_\ell^{i'} + X_R^{i'})$;
- $(S^i, T^i) = (S^{i'}, T^{i'})$;
- $(L_\ell'^i + Y_L^i, T^i) = (L_\ell'^{i'} + Y_L^{i'}, T^{i'})$;
- $(L_\ell'^i + Y_L^i, R_\ell'^i + Y_R^i) = (L_\ell'^{i'} + Y_L^{i'}, R_\ell'^{i'} + Y_R^{i'})$.

The main observation here is that when when i is akin to i', $(L_\ell^i, R_\ell^i) \neq (L_\ell^{i'}, R_\ell^{i'})$, and $(L_\ell'^i, R_\ell'^i) \neq (L_\ell'^{i'}, R_\ell^{i'})$; further, for $i \in E$, $(S^i, R_\ell^i) \neq (S^{i'}, R_\ell^{i'})$ and for $i \in D$, $(T^i, L_\ell'^i) \neq (T^{i'}, L_\ell'^{i'})$ (by construction of ideal oracle). This ensures that each of the collision-pairs is either impossible or consists of four uniformly sampled random variables; thus the reasoning for badA above holds for badB as well. In fact, the same reasoning carries over to badC.

C Full Analysis of badD

Recall that $i' < i$ and $\ell \stackrel{\text{def}}{=} \ell^{i'} = \ell^i$, and $\alpha_j(\cdot)$ and $\alpha_j^2(\cdot)$ are linear functions defined as

$$\alpha_j(x) \stackrel{\text{def}}{=} \alpha^{\ell-1-j} \cdot x \quad \text{and} \quad \alpha_j^2(x) \stackrel{\text{def}}{=} (\alpha^2)^{\ell-1-j} \cdot x.$$

We look at the two main cases separately, and branch into sub-cases for each:

- $(X_L^{*i}, X_R^{*i}) = (X_L^{*i'}, X_R^{*i'})$: We can write this collision as

$$\sum_{j=0}^{\ell-1} \alpha_j(X_j^i + X_j^{i'}) = 0 \quad \text{and} \quad \sum_{j=0}^{\ell-1} \alpha_j^2(X_j^i + X_j^{i'}) = \sum_{j=0}^{\ell-1} \alpha_j^2(R_j^i + R_j^{i'}).$$

  Using (1) we can rewrite these as

$$\sum_{j=0}^{\ell-1} \alpha_j(Z_j^i + Z_j^{i'}) = \sum_{j=0}^{\ell-1} \alpha_j(L_j'^i + L_j'^{i'}), \tag{26}$$

$$\sum_{j=0}^{\ell-1} \alpha_j^2(Z_j^i + Z_j^{i'}) = \sum_{j=0}^{\ell-1} \alpha_j^2(L_j'^i + L_j'^{i'} + R_j^i + R_j^{i'}). \tag{27}$$

  We first observe that since i is not akin to i', $X_j^i + X_j^{i'}$ cannot trivially disappear for all $j \in [1, .., \ell-1]$. Also, since $\alpha_j(X_j^i + X_j^{i'})$ sum to 0, there must be at least two indices in $[1, .., \ell-1]$ where $X_j^i + X_j^{i'}$ does not trivially disappear; let j_0 and j_1 be the two largest such indices, with $j_0 > j_1$. Now, we split into various sub-cases:

- $i \in E, i' \in E$: From (7), (26) and (27) we have

$$\sum_{j=0}^{\ell-1} \alpha_j (Z_j^{i:j} + Z_j^{i':j}) = \sum_{j=0}^{\ell-1} \alpha_j (L_j'^{i:j} + L_j'^{i':j}), \tag{28}$$

$$\sum_{j=0}^{\ell-1} \alpha_j^2 (Z_j^{i:j} + Z_j^{i':j}) = \sum_{j=0}^{\ell-1} \alpha_j^2 (L_j'^{i:j} + L_j'^{i':j} + R_j^{i:j} + R_j^{i':j}). \tag{29}$$

  By choice of j_0, $i : j_0 \neq i' : j_0$. Suppose $i : j_0 > i' : j_0$. If $i : j_0 \in D$, using (8), we replace $Z_{j_0}^{i:j_0}$ by $R_{j_0}^{i:j_0^2} + R_{j_0}^{i:j_0} + Z_{j_0}^{i:j_0^2} + S_1^{i:j_0^2} + S_1^{i:j_0}$. The basis element $S_1^{i:j_0}$ does not get canceled out; moreover, $R_{j_0}^{i:j_0}$ remains only in the top equation, while it gets canceled out in the bottom equation. Since $i : j = i' : j$ for all $j > j_0$, none of the adversary-queried blocks remaining in either equation came after $R_{j_0}^{i:j_0}$, so it is independent of the rest of the equation; along with the basis element $S_1^{i:j_0}$ (which appears in both equations), this makes the two collisions independent, thus occurring jointly with a probability $1/N^2$.
  If $i : j_0 \in E$, $Z_{j_0}^{i:j_0}$ is in the basis, and does not cancel out. On the right hand side of both equations, $L_{j_0}'^{i:j_0}$ remains uncanceled as well, while all later adversary queries get canceled. Thus, the two equations can become dependent with probability at most $1/N$; then, the common collision can occur with probability at most $1/N$. Thus, in either case, the joint collision can occur with a probability of more than $1/N^2$. The analysis is similar when $i : j_0 < i : j_0$; then we focus on the latter instead.
- $i \in E, i' \in D$: From (7), (8), (26) and (27) we have

$$\sum_{j=0}^{\ell-1} \alpha_j (Z_j^{i:j} + Z_j^{i':j} + S_1^{i'} + S_1^{i':j}) = \sum_{j=0}^{\ell-1} \alpha_j (L_j'^{i:j} + L_j'^{i':j} + R_j^{i'} + R_j^{i':j}), \tag{30}$$

$$\sum_{j=0}^{\ell-1} \alpha_j^2 (Z_j^{i:j} + Z_j^{i':j} + S_1^{i'} + S_1^{i':j}) = \sum_{j=0}^{\ell-1} \alpha_j^2 (L_j'^{i:j} + L_j'^{i':j} + R_j^{i:j} + R_j^{i':j}). \tag{31}$$

  By choice of j_0 and j_1, $i : j_0 \neq i'$ and $i : j_1 \neq i'$. Suppose $i : j_0 < i'$. Then $S_1^{i'}$ and $R_{j_0}^{i'}$ remain uncanceled in (30), and no adversary query block queried after $R_{j_0}^{i'}$ remains uncanceled; in (31), $S_1^{i'}$ remains uncanceled again, but there is no $R_{j_0}^{i'}$ and no adversary query block queried after it. Thus these two can occur jointly with a probability at most $1/N^2$.

  A symmetric argument can be used when $i : j_0 > i'$ and $i : j_0 \in D$: we replace $Z_{j_0}^{i:j_0}$ by $R_{j_0}^{i:j_0^2} + R_{j_0}^{i:j_0} + Z_{j_0}^{i:j_0^2} + S_1^{i:j_0^2} + S_1^{i:j_0}$ using (8), and observe that $S_1^{i:j_0}$ remains uncanceled in either equation, while $R_{j_0}^{i:j_0}$ remains

uncanceled in (30), but gets canceled out in (31), and no adversary query block queried after it remains in either equation.

When $i : j_0 > i'$ and $i : j_0 \in E$, but $i : j_1$ satisfied one of the above two conditions, we can argue as above using $i : j_1$ instead. If we also have $i : j_1 > i'$ and $i : j_1 \in E$, we observe that $Z_{j_0}^{i:j_0}$ and $Z_{j_1}^{i:j_1}$ are basis elements that do not get canceled out in either equation. Their combined contribution to the left-hand side of (30) is $\alpha^{\ell-1-j_0} \cdot Z_{j_0}^{i:j_0} + \alpha^{\ell-1-j_1} \cdot Z_{j_1}^{i:j_1}$ and to the left-hand side of (31) is $(\alpha^2)^{\ell-1-j_0} \cdot Z_{j_0}^{i:j_0} + (\alpha^2)^{\ell-1-j_1} \cdot Z_{j_1}^{i:j_1}$. These two collisions are independent since $\alpha^{\ell-1-j_0} \cdot (\alpha^2)^{\ell-1-j_1} \neq \alpha^{\ell-1-j_1} \cdot (\alpha^2)^{\ell-1-j_0}$, and thus can occur with a probability at most $1/N^2$.

- $i \in D, i' \in E$: From (7), (8), (26) and (27) we have

$$\sum_{j=0}^{\ell-1} \alpha_j (Z_j^{i:j} + Z_j^{i':j} + S_1^i + S_1^{i:j}) = \sum_{j=0}^{\ell-1} \alpha_j (L_j'^{i:j} + L_j'^{i':j} + R_j^i + R_j^{i:j}), \tag{32}$$

$$\sum_{j=0}^{\ell-1} \alpha_j^2 (Z_j^{i:j} + Z_j^{i':j} + S_1^i + S_1^{i:j}) = \sum_{j=0}^{\ell-1} \alpha_j^2 (L_j'^{i:j} + L_j'^{i':j} + R_j^{i:j} + R_j^{i':j}). \tag{33}$$

  This case is straightforward: S_1^i does not get canceled out in either equation, and R_j^i remains only in (32), with no terms queried after it. Thus the joint probability does not exceed $1/N^2$.
- $i \in D, i' \in D$: From (8), (26) and (27) we have

$$\begin{aligned} &\sum_{j=0}^{\ell-1} \alpha_j (Z_j^{i:j} + Z_j^{i':j} + S_1^i + S_1^{i'} + S_1^{i:j} + S_1^{i':j}) \\ &= \sum_{j=0}^{\ell-1} \alpha_j (L_j'^{i:j} + L_j'^{i':j} + R_j^i + R_j^{i'} + R_j^{i:j} + R_j^{i':j}), \qquad (34) \\ &\sum_{j=0}^{\ell-1} \alpha_j^2 (Z_j^{i:j} + Z_j^{i':j} + S_1^i + S_1^{i'} + S_1^{i:j} + S_1^{i':j}) \\ &= \sum_{j=0}^{\ell-1} \alpha_j^2 (L_j'^{i:j} + L_j'^{i':j} + R_j^{i:j} + R_j^{i':j}). \qquad (35) \end{aligned}$$

  The argument here is identical to that of the case above.

– $(Y_L^{*i}, Y_R^{*i}) = (Y_L^{*i'}, Y_R^{*i'})$: We can write this collision as

$$\sum_{j=0}^{\ell-1} \alpha_j^2 (Y_j^i + Y_j^{i'}) = \sum_{j=0}^{\ell-1} \alpha_j^2 (L_j'^i + L_j'^{i'}) \quad \text{and} \quad \sum_{j=0}^{\ell-1} \alpha_j (Y_j^i + Y_j^{i'}) = 0.$$

Using (2) we can rewrite these as

$$\sum_{j=0}^{\ell-1} \alpha_j^2 (Z_j^i + Z_j^{i'} + S_1^i + S_1^{i'}) = \sum_{j=0}^{\ell-1} \alpha_j^2 (L_j'^i + L_j'^{i'} + R_j^i + R_j^{i'}), \tag{36}$$

$$\sum_{j=0}^{\ell-1} \alpha_j (Z_j^i + Z_j^{i'} + S_1^i + S_1^{i'}) = \sum_{j=0}^{\ell-1} \alpha_j (R_j^i + R_j^{i'}). \tag{37}$$

As before, we let j_0 and j_1 be the two largest indices where $Y_j^i + Y_j^{i'}$ does not trivially vanish, with $j_0 > j_1$. Again, we split into various sub-cases:

- $i \in E, i' \in E$: From (7), (36) and (37) we have

$$\sum_{j=0}^{\ell-1} \alpha_j^2 (Z_j^{i:j} + Z_j^{i':j} + S_1^i + S_1^{i'}) = \sum_{j=0}^{\ell-1} \alpha_j^2 (L_j'^{i:j} + L_j'^{i':j} + R_j^{i:j} + R_j^{i':j}), \tag{38}$$

$$\begin{aligned} &\sum_{j=0}^{\ell-1} \alpha_j (Z_j^{i:j} + Z_j^{i':j} + S_1^i + S_1^{i'}) \\ &= \sum_{j=0}^{\ell-1} \alpha_j (L_j'^i + L_j'^{i'} + L_j'^{i:j} + L_j'^{i':j} + R_j^{i:j} + R_j^{i':j}). \end{aligned} \tag{39}$$

  Another straightforward case: S_1^i does not get canceled out in either equation, and $L_j'^i$ remains only in (39), with no terms queried after it. Thus the joint probability does not exceed $1/N^2$.

- $i \in E, i' \in D$: From (7), (8), (36) and (37) we have

$$\sum_{j=0}^{\ell-1} \alpha_j^2 (Z_j^{i:j} + Z_j^{i':j} + S_1^i + S_1^{i':j}) = \sum_{j=0}^{\ell-1} \alpha_j^2 (L_j'^{i:j} + L_j'^{i':j} + R_j^{i:j} + R_j^{i':j}), \tag{40}$$

$$\sum_{j=0}^{\ell-1} \alpha_j (Z_j^{i:j} + Z_j^{i':j} + S_1^i + S_1^{i':j}) = \sum_{j=0}^{\ell-1} \alpha_j (L_j'^i + L_j'^{i:j} + R_j^{i:j} + R_j^{i':j}). \tag{41}$$

  The argument here is identical to that of the case above.

- $i \in D, i' \in E$: From (7), (8), (36) and (37) we have

$$\sum_{j=0}^{\ell-1} \alpha_j^2 (Z_j^{i:j} + Z_j^{i':j} + S_1^{i:j} + S_1^{i'}) = \sum_{j=0}^{\ell-1} \alpha_j^2 (L_j'^{i:j} + L_j'^{i':j} + R_j^{i:j} + R_j^{i':j}), \tag{42}$$

$$\sum_{j=0}^{\ell-1} \alpha_j (Z_j^{i:j} + Z_j^{i':j} + S_1^{i:j} + S_1^{i'}) = \sum_{j=0}^{\ell-1} \alpha_j (L_j'^{i'} + L_j'^{i':j} + R_j^{i:j} + R_j^{i':j}). \tag{43}$$

By choice of j_0 and j_1, $i:j_0 \neq i'$ and $i:j_1 \neq i'$. Suppose $i:j_0 < i'$. Then $S_1^{i'}$ and $R_{j_0}^{i'}$ remain uncanceled in (43), and no adversary query block queried after $L'^{i'}_{j_0}$ remains uncanceled; in (42), $S_1^{i'}$ remains uncanceled again, but there is no $L'^{i'}_{j_0}$ and no adversary query block queried after it. Thus these two can occur jointly with a probability at most $1/N^2$.

A symmetric argument can be used when $i:j_0 > i'$ and $i:j_0 \in E$: we observe that $S_1^{i:j_0}$ remains uncanceled in either equation, while $L'^{i:j_0}_{j_0}$ remains uncanceled in (42), but gets canceled out in (43), and no adversary query block queried after it remains in either equation.

When $i:j_0 > i'$ and $i:j_0 \in D$, but $i:j_1$ satisfied one of the above two conditions, we can argue as above using $i:j_1$ instead. If we also have $i:j_1 > i'$ and $i:j_1 \in D$, we observe that $Z_{j_0}^{i:j_0}$ and $Z_{j_1}^{i:j_1}$ are basis elements that do not get canceled out in either equation. Their combined contribution to the left-hand side of (30) is $\alpha^{\ell-1-j_0} \cdot Z_{j_0}^{i:j_0} + \alpha^{\ell-1-j_1} \cdot Z_{j_1}^{i:j_1}$ and to the left-hand side of (31) is $(\alpha^2)^{\ell-1-j_0} \cdot Z_{j_0}^{i:j_0} + (\alpha^2)^{\ell-1-j_1} \cdot Z_{j_1}^{i:j_1}$. These two collisions are independent since $\alpha^{\ell-1-j_0} \cdot (\alpha^2)^{\ell-1-j_1} \neq \alpha^{\ell-1-j_1} \cdot (\alpha^2)^{\ell-1-j_0}$, and thus can occur with probability at most $1/N^2$.

- $i \in D, i' \in D$: From (8), (36) and (37) we have

$$\sum_{j=0}^{\ell-1} \alpha_j^2 (Z_j^{i:j} + Z_j^{i':j} + S_1^{i:j} + S_1^{i':j}) = \sum_{j=0}^{\ell-1} \alpha_j^2 (L'^{i:j}_j + L'^{i':j}_j + R_j^{i:j} + R_j^{i':j}), \tag{44}$$

$$\sum_{j=0}^{\ell-1} \alpha_j (Z_j^{i:j} + Z_j^{i':j} + S_1^{i:j} + S_1^{i':j}) = \sum_{j=0}^{\ell-1} \alpha_j (R_j^{i:j} + R_j^{i':j}). \tag{45}$$

By choice of j_0, $i:j_0 \neq i':j_0$. Suppose $i:j_0 > i':j_0$. The basis element $S_1^{i:j_0}$ does not get canceled out in either equation; moreover, $L'^{i:j_0}_{j_0}$ remains only in (44), while it gets canceled out in (45); also, none of the adversary-queried blocks queried after it remains in either equation. Thus the two collisions can occur jointly with a probability not exceeding $1/N^2$. The argument is symmetric when $i:j_0 < i':j_0$.

D Insecure Preliminary Variants

It took some time to arrive at the present definition of ZCZ. This section examines four variants of this design which allow attacks, and which provide an implicit rationale of our current definition.

D.1 Basic Design

Our first attempts started with an EME- and HCTR-like structure with two wrapping layers of ZHash. This already implies to cluster the message into $2n$-bit di-blocks (or dual blocks). A chaining was necessary to achieve PRP security,

i.e., every ciphertext bit must depend on every plaintext bit. For SPRP security, the opposite is also necessary, i.e., a chaining must also make every plaintext bit depend on every ciphertext bit. EME and AEZ provide an elegant solution where an X accumulator sums up the results X_i from the top encryption layer of the first $\ell - 1$ di-blocks (i.e., all but the final one):

$$X_i \leftarrow \widetilde{E}_K^{\mathsf{t},i,R_i}(L_i).$$

The accumulated value is then randomized and added it into the computation of the final di-block.

$$X \leftarrow \sum_{i=1}^{\ell-1-i} X_i, \qquad X_L \leftarrow \widetilde{E}_K^{\mathsf{x},0}(X), \qquad X_R \leftarrow \widetilde{E}_K^{\mathsf{x},1}(X).$$

The final di-block is processed then through the first layer of encryption and mixing in the middle. The outputs are then used to derive two n-bit values S and T:

$$S \leftarrow \widetilde{E}_K^{t\$,\ell,R_\ell+X_R}(L_\ell + X_L), \qquad T \leftarrow \widetilde{E}_K^{s\$,\ell,S}(R_\ell + X_R).$$

From S and T, a counter-mode layer derives chaining values Z_i, that are added to all previous di-blocks $1, \ldots, \ell - 1$:

$$Z_i \leftarrow \widetilde{E}_K \mathsf{c}, i, T(S) \qquad L'_i \leftarrow X_i + Z_i \qquad Y_i \leftarrow R_i + Z_i.$$

The result yields then the inputs Y_i, for $1 \le i \le \ell - 1$, to the bottom layer of encryption for the first $\ell - 1$ di-blocks:

$$R'_i \leftarrow \widetilde{E}_K^{\mathsf{b},i,L'_i}(Y_i).$$

The final di-block continues encryption of the bottom layer:

$$P_L \leftarrow \widetilde{E}_K^{\mathsf{c}\$,\ell,T}(S) \qquad P_R \leftarrow \widetilde{E}_K^{\mathsf{b}\$,\ell,P_L}(T).$$

The values Y_i are again accumulated to a chaining value Y

$$Y \leftarrow \sum_{i=1}^{\ell-1-i} Y_i, \qquad Y_L \leftarrow \widetilde{E}_K^{\mathsf{y},0}(Y), \qquad Y_R \leftarrow \widetilde{E}_K^{\mathsf{y},1}(Y),$$

that is used to mask the output of the final di-block, mirroring the top:

$$L'_\ell \leftarrow Y_L + P_L, \qquad R'_\ell \leftarrow Y_R + P_R.$$

This design yielded a basis for the later structure. However, the naive usage of ZHash in this approach opens several possible attack vectors:

– We had to compute additional values Z_i to prevent that an adversary could observe differences ΔY_i in the left branches of the ciphertext di-blocks $\Delta L'_i$.

– We introduced multiplications with powers of a primitive element in the computations of X_L and X_R to prevent that a collision in either would also lead to a collision in the other value (and similarly for Y_L and Y_R).
– Thereupon, we noticed that pure sums for X_L and X_R would be insufficient as the differences in ΔX_L and ΔX_R could be canceled by smart choice of ΔL_ℓ and ΔR_ℓ, and a randomization of X_L and X_R was required.
– Moreover, while multiplications with powers of α are quite fast, it is well-known since the days of EME [23, 25] that one can cancel differences in certain di-blocks that yield a zero sum if the multiplications of the same chaining value cover more than n di-blocks. Therefore, we had to introduce new chaining values similar to EME* [20].

To illustrate the relevance of each design aspect on its own, we describe in the following attacks on preliminary versions with respect to the final ZCZ construction, where only the particular element that we added as countermeasure would be missing.

D.2 Variant I: Omitting the Additions of S^i to The Right Branches

In this construction, the values Y_j^i are computed as $Y_j^i = R_j^i + Z_j^i$. Suppose that we make q queries such that query i has at least i di-blocks, and for each $i \in [1..q-1]$,

$$\left(L_i^i, R_i^i\right) = \left(L_i^{i+1}, R_i^{i+1}\right).$$

This ensures that $X_i^i = X_i^{i+1}$, so we can calculate

$$\Delta Y_i \stackrel{\text{def}}{=} Y_i^i + Y_i^{i+1} = L_i'^i + L_i'^{i+1}.$$

When q is chosen in the order of n, with high probability, we can find a zero-sum subset of $\left\{\alpha^{q-1-i} \cdot \Delta Y_i \mid i \in [1..q-1]\right\}$. Assume for some $\mathcal{I} \subseteq [1..q-1]$, we have

$$\sum_{i \in \mathcal{I}} \alpha^{q-1-i} \cdot \Delta Y_i = 0.$$

For details on how to efficiently identify such a subset, see e.g., Bellare and Micciancio [3]. Moreover, from $Y_i^i + Y_i^{i+1} = L_i'^i + L_i'^{i+1}$, it follows that $Y_i^i = L_i'^i = Y_i^{i+1} + L_i'^{i+1}$ for all i. Therefore, it holds that

$$\sum_{i \in \mathcal{I}} \alpha^{2(q-1-i)} \cdot \left(\Delta Y_i + \Delta L_i'^i\right) = 0.$$

For a distinguisher, we ask two decryption queries as follows: the first query consists of q blocks, with

$$\left(L_1'^1, R_1'^1\right), \left(L_2'^2, R_2'^2\right), \ldots, \left(L_{q-1}'^{q-1}, R_{q-1}'^{q-1}\right)$$

as the first $q-1$ di-blocks; in the second query, for each $i \in \mathcal{I}$, the i-th di-block is replaced by $(L_i'^{i+1}, R_i'^{i+1})$, and everywhere else, it is identical to the first query. These two queries lead to a collision in both $(Y^{*1}_L, Y^{*1}_R) = (Y^{*2}_L, Y^{*2}_R)$, and hence will result in a collision in the right half of the final plaintext di-block $(L^1_\ell, R^1_\ell) = (L^2_\ell, R^2_\ell)$.

In contrast to this variant, the adversary cannot control differences in the values Y_i over different queries with a common prefix in ZCZ. There, the values S_i cannot efficiently be replicated over different queries. The unknown difference in the values S_i masks the difference in the values Y_i, rendering the attack ineffective.

D.3 Variant II: Omitting the Multiplications by Powers of α

As a consequence of the attack above, we also derived the values S^i and added them to the right branches of all di-blocks but the final one. Still, a number of further attack vectors remained. A clear point in the basic design was that a collision in X_L would automatically lead to a collision also in X_R. Subsequently, we added multiplications by powers of a primitive element α to one of them. Their relevance is outlined in the following attack.

In this variant, the values which means X^*_L and X^*_R are computed as

$$X^*_L = \sum_{i=1}^{\ell} X_i, \quad \text{and} \quad X^*_R = \sum_{i=1}^{\ell} X_i + R_i;$$

the values Y^*_L and Y^*_R are computed analogously. Clearly, one can query $q = 2^{n/2}$ queries which share equal di-blocks $R^i_j = R^k_j$ for all $1 \le i < k \le q$ and all j. This variant allows then a birthday distinguisher. Given $2^{n/2}$ queries, the probability is significant that there exist two messages for which $X^{*i}_L = X^{*k}_j$ holds. This event implies $X^{*i}_R = X^{*k}_R$ naturally.

D.4 Variant III: Omitting the Encryptions from (X^*_L, X^*_R) and (Y^*_L, Y^*_R) to (X_L, X_R) and (Y_L, Y_R)

Having introduced the second element Z_i for adding and the multiplications with powers of α, we already had a version that would be almost secure up to the birthday bound. However, we observed soon that an adversary could still choose the values $(\Delta L_\ell, \Delta R_\ell)$ well such that their differences would cancel those in ΔX_L and ΔX_R after about $2^{n/2}$ queries. In the following, we sketch an attack if we would omit the encryptions from from (X^*_L, X^*_R) and (Y^*_L, Y^*_R) to (X_L, X_R) and (Y_L, Y_R).

In this construction, it holds that

$$X_L = \sum_{j=1}^{\ell-1} \alpha^{\ell-1-j} X_j \quad \text{and} \quad X_R = \sum_{j=1}^{\ell-1} (\alpha^2)^{\ell-1-j} (X_j + R_j),$$

which means $X_L = X_L^*$, $X_R = X_R^*$, $Y_L = Y_L^*$, and $Y_R = Y_R^*$. This construction would allow a birthday distinguisher. Assume, we choose $q = 2^{n/2}$ messages that possess equal length ℓ and that consist of at least two di-blocks plus the final di-block each. The messages differ from each other in exactly three blocks: the final di-block (L_ℓ^i, R_ℓ^i) and some block L_j^i for some fixed index j. The final di-blocks are chosen so that it holds for every message:

$$R_\ell^i = \alpha^{2j-j} L_\ell^i = \alpha^j L_\ell^i.$$

If there exist two messages M^i and M^k for which $\Delta X_j = X_j^i + X_j^k = \Delta L_\ell = L_\ell^i + L_\ell^k$, then it holds that

$$\begin{aligned} \Delta X_L &= \Delta L_\ell \text{ and} \\ \Delta X_R &= \alpha^{\ell-1-j} \Delta X_j = \Delta R_\ell. \end{aligned}$$

For this pair of messages M^i and M^k, the values $S^i = S^k$ and $T^i = T^k$ will be identical, and we obtain collisions in the ciphertexts of all di-blocks whose plaintext di-blocks were equal between both messages. When choosing $2^{n/2}$ messages, the probability to obtain such a pair becomes significant. However, encrypting the sums X_L^*, X_R^*, Y_L^*, and Y_R^* effectively prevents such distinguishers.

D.5 Variant IV: Using S_1 for More Than n Di-Blocks

Finally, there is another attack if one doubles a single chaining value S for more than n times, as has already been observed by Halevi and Rogaway [23] on EME. Here, we discuss its impact on the preliminary version of ZCZ that would have only S_1.

For messages that consist of more than one chunk (i.e., a sequence of n non-final di-blocks), we have to derive a new chaining value S_i for every chunk. Otherwise, a similar attack as for Variant I would be possible also here. For this variant, the values Y_j^i would be computed as

$$Y_j^i = R_j^i + Z_j^i + S_1^i,$$

for all $1 \le j \le \ell - 1$, and all queries i. For our distinguisher, we can apply the same strategy as in Variant I, and ask q queries of same length ℓ, with $\ell > 2n+1$ di- blocks, such that the first $\ell - 1$ di-blocks are always equal over all queries:

$$(L_j^i, R_j^i) = (L_j^{i'}, R_j^{i'}), \quad \text{for all } 1 \le i \ne i' \le q, \quad 1 \le j \le \ell - 1.$$

The queries are pairwise distinct in their final di-block (L_ℓ, R_ℓ). As in the attack on Variant I, the adversary can observe the differences

$$\Delta Z_j^{i,i'} = \Delta Z_j^i + \Delta Z_j^{i'}$$

from the differences in the left branches of the di-blocks

$$\Delta L'^{i,i'}_j = \Delta L'^{i}_j + \Delta L'^{i'}_j.$$

Given $\ell > n+1$, there exist subsets of indices $\mathcal{J} \subseteq \{1, \ldots, n+1\}$ s.t. it holds:

$$\sum_{j \in \mathcal{J}} \alpha^{\ell-1-j} = 0.$$

Such sets must exist for this variant, e.g., for the case $n = 128$ and $\ell = 130$, and the primitive polynomial of GCM, one option would be $\mathcal{J} = \{0, 1, 2, 7, 128\}$. The concrete set of indices $\mathcal{J}$ depends on the chosen primitive polynomial. When j is in the order of $2n$, there are about 2^n choices (since we can multiply any polynomial of degree at most n to the primitive polynomial) of $\mathcal{J}$. For a second necessary condition, define $\Delta W_j^{i,i'} = \alpha^{\ell-1-j} \cdot \Delta Z_j^{i,i'}$. Then, the second condition is

$$\sum_{j \in \mathcal{J}} \Delta W_j^{i,i'} = 0.$$

Among the 2^n choices for $\mathcal{J}$, there exists one subset of indices that will also have this second condition fulfilled. To efficiently find a solution for both conditions, the adversary can define $2n$-bit values

$$\left(\Delta W_j^{i,i'} \parallel \alpha^{\ell-1-j}\right)$$

and find a subset $\{J\}'$ for which the sum yields 0^{2n} by solving linear equations, analogously to the approach in [3]. Once the adversary has found it, it holds that

$$\sum_{j \in \mathcal{J}'} \left(\alpha^{\ell-1-j} \cdot \Delta Y_j^{i,i'}\right) = \underbrace{\sum_{j \in \mathcal{J}'} \left(\alpha^{\ell-1-j} \cdot \Delta Z_j^{i,i'}\right)}_{=0} + \underbrace{\sum_{j \in \mathcal{J}'} \left(\alpha^{\ell-1-j} \cdot \Delta S_1^{i,i'}\right)}_{=0} = 0.$$

Note that the adversary can hold the values R_j^i equal over all queries $1 \leq i \leq q$. Since

$$\sum_{j \in \mathcal{J}} \alpha^{\ell-1-j} = 0 \quad \text{implies that} \quad \sum_{j \in \mathcal{J}} (\alpha^2)^{\ell-1-j} = 0,$$

the collision would affect always both X_L^* and X_R^* (or the corresponding values Y_L^* and Y_R^* if the attack uses decryption queries). So, the same attack as for Variant I would apply.

Naturally, the strategy of deriving a new chaining value for every chunk of n di-blocks helped protecting against it, as has already been used in EME* [20] and AEZ [25].