IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2017, No. 4, pp. 306–325. DOI:10.13154/tosc.v2017.i4.306-325

ZMAC$^+$ – An Efficient Variable-output-length Variant of ZMAC

Eik List[1] and Mridul Nandi[2]

[1] Bauhaus-Universität Weimar, Weimar, Germany
eik.list@uni-weimar.de
[2] Indian Statistical Institute, Kolkata, India,
mridul@isical.ac.in

Abstract. There is an ongoing trend in the symmetric-key cryptographic community to construct highly secure modes and message authentication codes based on tweakable block ciphers (TBCs). Recent constructions, such as Cogliati et al.'s HaT or Iwata et al.'s ZMAC, employ both the n-bit plaintext and the t-bit tweak simultaneously for higher performance.
This work revisits ZMAC, and proposes a simpler alternative finalization based on HaT. As a result, we propose HtTBC, and call its instantiation with ZHash as a hash function ZMAC$^+$. Compared to HaT, ZMAC$^+$ (1) requires only a single key and a single primitive. Compared to ZMAC, our construction (2) allows variable, per-query parametrizable output lengths. Moreover, ZMAC$^+$ (3) avoids the complex finalization of ZMAC and (4) improves the security bound from $O(\sigma^2/2^{n+min(n,t)})$ to $O(q/2^n + q(q+\sigma)/2^{n+min(n,t)})$ while retaining a practical tweak space.

Keywords: Message authentication code · tweakable block cipher · provable security.

1 Introduction

Tweakable Block Ciphers. A *tweakable block cipher* (or TBC for short) $\widetilde{\pi} : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \to \mathcal{X}$ is a family of cryptographic permutations over inputs $X \in \mathcal{X}$, where both the secret key $K \in \mathcal{K}$ and the public tweak $T \in \mathcal{T}$ define the permutation. The additional tweak distinguishes them from classic block ciphers. While introduced in the Hasty Pudding Cipher [SO98], the concept has been formalized first by Liskov, Rivest, and Wagner in [LRW02].

Block-cipher-based Message Authentication Codes. Message Authentication Codes (MACs) are secret-key cryptographic schemes. Given a message, their goal is to produce a key-dependent tag that can be verified by all parties who share the secret key, but that is infeasible to be forged otherwise. While MACs can be stateful, randomized, nonce-based, or stateless deterministic, we focus on the latter in the remainder. Over the decades, many block-cipher-based constructions have been proposed, e. g., CBC-MAC [BKR94], OMAC [IK03b, IK03a], or PMAC [BR02]. The security of earlier constructions has often have been limited by birthday bound, which motivated numerous researches to propose MACs with increased security guarantees, e.g., PMAC$^+$ [Yas11], or LightMAC [LPTY16].

Secure Constructions from Tweakable Block Ciphers. For the domain of MACs, a number of TBC-based highly secure MACs were published. Inspired by the design of Yasuda's PMAC$^+$ [Yas11], Naito [Nai15] proposed two fully-parallelizable deterministic BBB-secure MACs, PMAC_TBC1k and PMAC_TBC3k, that employed one and three

Received: 2017-09-01, Accepted: 2017-11-01, Published: 2017-12-15

keys, respectively. Later in [LN17], Naito's single-key construction was revisited and combined with the CTRT mode for deterministic authenticated encryption.

HaT, ZHash, and ZMAC. There is a recent trend of MACs to process the message input in both the tweak and the state input of a TBC simultaneously for higher efficiency. Cogliati et al. [CLS17] proposed (among others) two such fully secure MACs, called HaT and NaT for Hash-as-Tweak and Nonce-as-Tweak, from the combination of a universal hash function with a call to a tweakable block cipher for finalization, among which HaT caught our particular interest, for it is stateless and deterministic. At CRYPTO'17, Iwata et al. [IMPS17] introduced ZMAC, and further employed it for authenticated encryption. Like previous MACs, ZMAC combines a TBC-based hash function whose output is given into a PRF to derive the authentication tag. Its innovation, however, stemmed from its hash function ZHash: in contrast to previous constructions [KR11, LN17, Nai15], ZHash splits the message into $(n+t)$-bit blocks, where each block uses both the n-bit plaintext input and t-bit tweak input of the TBC, which rendered ZHash highly efficnet. To derive the authentication tag from the output of ZHash, Iwata et al. employed a finalization based on two sums of two independent permutations each. Analyzing the security of the sum of independent permutations has seen intensive research for decades (e. g., see [BI99, BKR98, Luc00, MP15, Pat08a, Pat13]), and has always been highly sophisticated topic. The choice by Iwata et al. was motivated by a recent result of $O((q/2^n)^{3/2})$ by Dai et al. [DHT17], whose lengthy proof again reflects the complexity of the subject.

Variable-output-length PRFs. PRFs with variable output lengths (VOLPRFs) produce a variable-length output similar to e. g., stream ciphers. Though, the notion of extendable-output functions (XOF) received attention lately with the standardization of the SHA-3-based XOFs SHAKE128 and SHAKE256 [NIS15], or recent proposals as e. g., Naito's sandwich keyed sponge [Nai16] or Bertoni et al.'s Farfalle [BDP$^+$16]. The flexibility of VOLPRFs is advantageous for they can be used not only as a MAC, but also as a mode of operation, a key-derivation function, or a wide-block cipher, e. g., in the HHFHFH mode [Ber16]. Moreover, Hoang et al. [HKR15] showed that secure wide-block ciphers can be further extended in straight-forward manner to robust AE schemes [BR00].

Contribution. This work revisits ZMAC and considers an alternative, yet simpler finalization ZFin$^+$. We propose HtTBC, a VOLPRF that combines a hash function producing a $(n+t)$-bit output with a finalization that uses both the plaintext and the tweak inputs of the TBC, similar to HaT. We call our instantiation of HtTBC with ZHash as hash function ZMAC$^+$. As result, this work provides a fourfold contribution: compared to HaT, ZMAC$^+$ (1) requires only a single key and a single primitive. Compared to both HaT and ZMAC, our construction (2) is a VOLPRF with per-query parametrizable output length. Moreover, ZMAC$^+$ (3) avoids the complex sum-of-PRPs finalization, and (4) improves the security bound of $O(\sigma^2/2^{n+\min\{n,t\}})$ from ZMAC to $O(q/2^n + q(q+\sigma)/2^{n+\min\{n,t\}})$ without the need of enlarging the tweak space by many indices, as in ZMAC[$\mathbb{ZHASH}$] [IMPS17]. Table 1 summarizes and compares the properties of ZMAC$^+$ with previous TBC-based MACs.

Outline. The remainder of this work is structured as follows. After the preliminaries, Section 3 proposes ZHash, HtTBC, and ZMAC$^+$. Section 4 studies the security of HtTBC and derives the bounds for PRF and VOLPRF security under two requirements of the hash function. The subsequent Sections 5 and 6 analyze those requirements for ZHash. Section 7 discusses briefly potential instantiations; Section 8 concludes this work.

Table 1: Comparison of stateless deterministic BBB-secure TBC-based PRFs with our proposal. Output length is in bit. #Bit/TBC call = #message bits processed per TBC call in the hash function. HaT depends on two calls to ϵ-AU hash functions (HFs). $\tau = \min\{n, t\}$. *: PMAC2x and PMACx were revised in the full versoin of [LN17] after a padding flaw pointed out by [MI17].

Construction	#Primitives	Hashing	Finalization	#Bit/TBC call	#Keys	Security	Output length	Reference
PMAC_TBC1k	1	TBC	TBC	n	1	$O(n^2q^2/2^{2n})$	$2n$	[Nai15]
PMAC_TBC3k	1	TBC	TBC	n	3	$O(q^2/2^{2n})$	$2n$	[Nai15]
PMACx*	1	TBC	TBC	n	1	$O(q^2/2^{2n})$	n	[LN17]
PMAC2x*	1	TBC	TBC	n	1	$O(q^2/2^{2n})$	$2n$	[LN17]
HaT	2	HF	TBC	–	3	$O(q^2\epsilon^2 + q/2^n)$	n	[CLS17]
ZMAC	1	TBC	TBC	$n+t$	1	$O(\sigma^2/2^{n+\tau})$	$2n$	[IMPS17]
ZMAC$^+$	1	TBC	TBC	$n+t$	1	$O(q/2^n + q(q+\sigma)/2^{n+\tau})$	dn	**This work**

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. For a given set $\mathcal{X}$, we write $\mathcal{X}^+$ to denote $\bigcup_{i=1}^{\infty} \mathcal{X}^i$, and $\mathcal{X}^*$ to mean $\bigcup_{i=0}^{\infty} \mathcal{X}^i$. We write $\{0,1\}^x$ for the set of bit strings of length x. We denote the concatenation of binary strings X and Y by $X \,\|\, Y$ and the result of their bitwise XOR by $X \oplus Y$. We write the length of X in bits as $|X|$, and X_i for the i-th block of X. For any $X \in \{0,1\}^n$ and $i \leq n$, we denote by $\textsc{msb}_i(X)$ the i most significant and by $\textsc{lsb}_i(X)$ the i least significant bits of X. Furthermore, we denote by $X \twoheadleftarrow \mathcal{X}$ that X is chosen uniformly at random from the set $\mathcal{X}$. We define two sets of particular interest: $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$ is the set of all functions $F : \mathcal{X} \to \mathcal{Y}$ 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 n-bit blocks i. e., $X_1 \,\|\, \ldots \,\|\, X_x = X$, and $|X_i| = n$ for $1 \leq i \leq x - 1$, and $|X_x| \leq n$. For any $X \in \{0,1\}^{n+t}$, we denote by $(X_1, X_2) \xleftarrow{n,t} \mathcal{X}$ the splitting of X into $X_1 = \textsc{msb}_n(X)$ and $X_2 = \textsc{lsb}_t(X)$. Moreover, we define $\langle x \rangle_n$ to denote the encoding of a non-negative integer x into its n-bit representation. For bit strings $X \in \{0,1\}^n$ and $Y \in \{0,1\}^t$ of different lengths $n \neq t$, we define

$$X \oplus_t Y \overset{\text{def}}{=} \begin{cases} \textsc{msb}_t(X) \oplus Y & \text{if } t \leq n, \\ (X \,\|\, 0^{t-n}) \oplus Y & \text{if } t > n. \end{cases}$$

For two sets $\mathcal{X}$ and $\mathcal{Y}$, a uniform random function $\rho : \mathcal{X} \to \mathcal{Y}$ is a mapping of 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. For two integers n, k with $n \geq k \geq 1$, we denote the falling factorial as $(n)_k \overset{\text{def}}{=} \prod_{i=0}^{k-1}(n - i)$.

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 $\Delta_{\mathbf{A}}(\mathcal{O}^1; \mathcal{O}^2) \overset{\text{def}}{=} |\Pr[\mathbf{A}^{\mathcal{O}^1} \Rightarrow 1] - \Pr[\mathbf{A}^{\mathcal{O}^2} \Rightarrow 1]|$ 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 the adversary, if any. W.l.o.g., we assume that $\mathbf{A}$ never asks queries to which it already knows the answer.

Block Ciphers and Tweakable Block Ciphers. A block cipher is a mapping $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ with associated key space $\mathcal{K}$ and message space $\mathcal{M}$ such that for every key $K \in \mathcal{K}$, $E(K, \cdot)$ is a permutation over $\mathcal{M}$. A tweakable block cipher $\widetilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ with associated key space $\mathcal{K}$, tweak space $\mathcal{T}$, and message space $\mathcal{M}$ is a permutation over $\mathcal{M}$ for every key $K \in \mathcal{K}$ and tweak $T \in \mathcal{T}$. We also write $\widetilde{E}_K^T(\cdot)$ as short form of $\widetilde{E}(K, T, \cdot)$ in the remainder.

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

Definition 2 (PRF Advantage)**.** Let $\mathcal{K}$, $\mathcal{X}$, and $\mathcal{Y}$ be non-empty sets and let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a keyed function. Let $\rho \twoheadleftarrow \mathsf{Func}(\mathcal{X}, \mathcal{Y})$ and $K \twoheadleftarrow \mathcal{K}$. Then, the PRF advantage of $\mathbf{A}$ w. r. t. F is defined as $\mathbf{Adv}_F^{\mathrm{PRF}}(\mathbf{A}) \overset{\text{def}}{=} \Delta_{\mathbf{A}}(F_K; \rho)$.

For the remaining definitions in this section, let $\mathcal{K}$, $\mathcal{X}$, and $\mathcal{Y}$ be non-empty sets, where we restrict our considerations to $\mathcal{Y} \subset \{0,1\}^*$, and let $H : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ and $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be keyed functions.

Definition 3 (VOLPRF Advantage)**.** Let $F : \mathcal{K} \times \mathcal{X} \times \mathbb{N} \to (\mathcal{Y})^+$ be a keyed function whose output length is determined by the second parameter, i. e., which, for arbitrary input $(X, d) \in \mathcal{X} \times \mathbb{N}$, always outputs some $Y \in (\mathcal{Y})^d$. Let $K \twoheadleftarrow \mathcal{K}$, and define the ideal oracle $\rho \twoheadleftarrow \mathsf{Func}(\mathcal{X} \times \mathbb{N}, \mathcal{Y}^+)$ s. t. for arbitrary input $(X, d) \in \mathcal{X} \times \mathbb{N}$, it always outputs some $Y \twoheadleftarrow (\mathcal{Y})^d$. Then, the PRF advantage of $\mathbf{A}$ w. r. t. F is defined as $\mathbf{Adv}_F^{\mathrm{VOLPRF}}(\mathbf{A}) \overset{\text{def}}{=} \Delta_{\mathbf{A}}(F_K; \rho)$.

Definition 4 (Differential Probability)**.** For two distinct messages $X, X' \in \mathcal{X}$ and all $\Delta \in \mathcal{Y}$, we define the differential probability of H as

$$\mathsf{DP}_H\left(X, X', \Delta\right) \overset{\text{def}}{=} \Pr_{K \twoheadleftarrow \mathcal{K}}\left[H_K(X) \oplus H_K(X') = \Delta\right].$$

Definition 5 (Almost-Universal Hash Function [CW79])**.** We say that H is ϵ-almost-universal (ϵ-AU) if, for all distinct $X, X' \in \mathcal{X}$, it holds that $\mathsf{DP}_H\left(X, X', 0\right) \leq \epsilon$.

Almost-XOR-universal hash functions were introduced in [Kra94], the term was coined in [Rog95].

Definition 6 (Almost-XOR-Universal Hash Function [Kra94])**.** We say that H is ϵ-almost-XOR-universal (ϵ-AXU) if, for all distinct $X, X' \in \mathcal{X}$ and any $\Delta \in \mathcal{Y}$, it holds that $\mathsf{DP}_H(X, X', \Delta) \leq \epsilon$.

Minematsu and Iwata [MI15] defined a variant called partial-almost-XOR-universality.

Definition 7 (Partial-AXU Hash Function [MI15])**.** Let $\mathcal{Y} = \{0,1\}^n \times \{0,1\}^t$. We say that H is (n, t, ϵ)-partial-AXU (pAXU) if, for all distinct $X, X' \in \mathcal{X}$ and all $\Delta \in \{0,1\}^n$, it holds that $\mathsf{DP}_H\left(X, X', (\Delta, 0^t)\right) \leq \epsilon$.

Later in this work, we will have to derive upper bounds on the probability of differences between parts of hash function outputs. For this purpose, we derive the notion of truncated almost-XOR-universal hash functions.

Definition 8 (Truncated-AXU Hash Function)**.** Let $\mathcal{Y} = \{0,1\}^n \times \{0,1\}^t$. We say that H is (n, t, ϵ)-truncated-AXU (tAXU) if, for all distinct $X, X' \in \mathcal{X}$ and all $\Delta_2 \in \{0,1\}^t$, it holds that

$$\sum_{\Delta_1 \in \{0,1\}^n} \mathsf{DP}_H\left(X, X', (\Delta_1, \Delta_2)\right) \leq \epsilon.$$

Intuitively, truncated-AXU states that the function H' that always returns the output of $H(X)$ truncated to the t least significant bits, for all inputs (X), is ϵ-AXU.

Algorithm 1 Definition of ZMAC$^+$.

```
function HtTBC[$\widetilde{E}_K, H$]($\underline{M}, d$)
    $M \leftarrow$ Encode$_{n,t}(\underline{M}, d)$
    $(Y, X) \leftarrow H[\widetilde{E}_K](M)$
    return ZFin$^+[\widetilde{E}_K](Y, X, d)$

function Encode$_{n,t}(\underline{M}, d)$
    $p \leftarrow (n+t) - ((|\underline{M}| + n + 1) \bmod (n+t))$
    if $p = (n + t)$ then
        $p \leftarrow 0$
    return $\underline{M} \parallel 1 \parallel 0^p \parallel \langle d \rangle_n$

function ZFin$^+[\widetilde{E}_K](Y, X, d)$
    for $i = 1$ to $d$ do
        $T_i \leftarrow X \oplus \langle i - 1 \rangle_t$
        $U_i \leftarrow \widetilde{E}_K^{1,T_i}(Y)$
    return $(U_1, \ldots, U_d)$

function ZMAC$^+[\widetilde{E}_K](\underline{M}, d)$
    return HtTBC[$\widetilde{E}_K$, ZHash]($\underline{M}, d$)

function ZHash$[\widetilde{E}_K](M)$
    $X \leftarrow 0^t$; $Y \leftarrow 0^n$
    $L \leftarrow \widetilde{E}_K^{2,\langle 0 \rangle_t}(\langle 1 \rangle_n)$; $R \leftarrow \widetilde{E}_K^{2,\langle 1 \rangle_t}(\langle 1 \rangle_n)$
    Parse $(M[1], \ldots, M[m]) \xleftarrow{n+t} M$
    for $i \leftarrow 1$ to $m$ do
        $(M_L[i], M_R[i]) \xleftarrow{n,t} M[i]$
        $S_i \leftarrow M_L[i] \oplus 2^{i-1} L$
        $T_i \leftarrow M_R[i] \oplus_t 2^{i-1} R$
        $Y_i \leftarrow \widetilde{E}_K^{0,T_i}(S_i)$
        $X_i \leftarrow M_R[i] \oplus_t Y_i$
        $X \leftarrow X \oplus X_i$
        $Y \leftarrow 2 \cdot (Y \oplus Y_i)$
    return $(Y, X)$
```

The H-Coefficient Technique. The H-coefficients technique is a proof method due to Patarin [CS14, Pat08b]. The results of the interaction of an adversary $\mathbf{A}$ with its oracles are collected in a transcript τ: $\tau = \langle (X_1, Y_1), \ldots, (X_q, Y_q) \rangle$, where (X_i, Y_i) denotes input and output of the i-th query of $\mathbf{A}$. The task of $\mathbf{A}$ is to distinguish the real world $\mathcal{O}_{\text{real}}$ from the ideal world $\mathcal{O}_{\text{ideal}}$. A transcript τ is called *attainable* if the probability to obtain τ in the ideal world is non-zero. One assumes that $\mathbf{A}$ does not ask duplicate queries or queries prohibited by the game or to which it already knows the answer. Denote by Θ_{real} and Θ_{ideal} the distribution of transcripts in the real and the ideal world, respectively. Then, the fundamental Lemma of the H-coefficients technique states:

Lemma 1 (Fundamental Lemma of the H-coefficient Technique [Pat08b])**.** Assume, the set of attainable transcripts is partitioned into two disjoint sets GoodT and BadT. Further assume, there exist $\epsilon_1, \epsilon_2 \geq 0$ such that for any transcript $\tau \in$ GoodT, it holds that

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

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

The proof is given in [CS14, Pat08b].

3 ZHash, HtTBC, and ZMAC$^+$

This section recalls the definitions of ZHash, ZFin, and ZMAC by Iwata et al. [IMPS17] and refines a new finalization ZFin$^+$ as well as the constructions HtTBC and ZMAC$^+$. Throughout the remainder, we define $k, n, t \geq 1$ as fixed positive integers, and non-empty sets $\mathcal{K} = \{0,1\}^k$, $\mathcal{L}$, $\mathcal{T} = \{0,1\}^t$, and $\mathcal{D} = \{1, \ldots, 2^{\min\{n,t\}}\}$. Since ZHash is combined with a finalization in ZMAC, it is practical to consider a tweakable block cipher whose tweak space adds an additional domain. For this purpose, we define a set of domains $\mathcal{I} = \{0, 1, 2\}$, an augmented tweak space $\mathcal{T}' = \mathcal{I} \times \mathcal{T}$, and consider a TBC $\widetilde{E} : \mathcal{K} \times \mathcal{T}' \times \{0,1\}^n \to \{0,1\}^n$.

ZHash. ZHash is a fully parallelizable TBC-based hash function that processes each $(n+t)$-bit block of the message by what Iwata et al. called the XT construction, a reduced version of the XTX tweak-domain extender by Minematsu and Iwata [MI15]. Each call

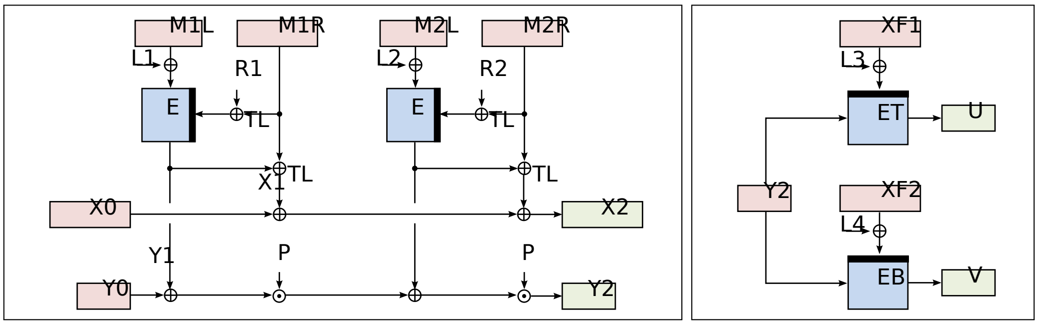

Figure 1: Components of ZMAC$^+[\widetilde{E}_K]$. **Left:** ZHASH$[\widetilde{E}_K]$. **Right:** ZFIN$^+[\widetilde{E}_K]$ for $d = 2$.

to XT takes a message block $M[i] = (M_L[i] \parallel M_R[i])$; its most significant n bit, $M_L[i]$, are XORed with a masking key $2^{i-1}L$ to produce an input S_i; its least significant t bit, $M_R[i]$, are XORed with an independent masking key $2^{i-1}R$ to produce a tweak T_i. To address the case when $t \neq n$, T_i is computed by $T_i \leftarrow 2^{i-1}R \oplus_t M_R[i]$. (S_i, T_i) serves then as state and tweak input to the TBC. The output of every call to $\widetilde{E}$ in ZHASH, $Y_i \leftarrow \widetilde{E}_K^{0,T_i}(S_i)$ is XORed (partially if $t \neq n$) again to $M_R[i]$ to produce an n-bit value Y_i, and a t-bit value X_i. The values X and Y accumulate the XOR sum of the terms X_i and $2^{m+1-i}Y_i$, for $i \in \{1, \dots, m\}$, in PMAC$^+$-like fashion to obtain a hash function with beyond-birthday-bound security. A definition of ZHASH is given in Algorithm 1.
In contrast to ZHASH, our variant takes an additional parameter d that defines the number of output blocks to render HTTBC a VOLPRF. For this purpose, we defined the additional set $\mathcal{D}$ for the domain of d, and employ an injective encoding that, given $(\underline{M}, d) \in \mathcal{M} \times \mathcal{D}$, always appends a single 1 bit to $\underline{M}$, followed by p zero bits and by d encoded as n-bit string, where p denotes the minimal non-negative number so that the length of the encoded output has a multiple of $n + t$ bit.

ZFin$^+$. ZFIN$^+$ is our finalization. For arbitrary inputs $(Y, X, d) \in \{0,1\}^n \times \mathcal{T} \times \mathcal{D}$, it computes and outputs $(U_1, \dots, U_d) \in (\{0,1\}^n)^d$, where

$$U_i \overset{\text{def}}{=} \widetilde{E}_K^{1, X \oplus \langle i-1 \rangle_t}(Y), \qquad \text{for all } 1 \leq i \leq d.$$

HtTBC. HASH-THEN-TBC (HTTBC) resembles the well-known Hash-then-PRF paradigm, and is close to Cogliati et al.'s stateless deterministic variant HAT. We first define HTTBC in Algorithm 1 and below.

Definition 9 (HTTBC)**.** Let $\widetilde{E} : \mathcal{K} \times \mathcal{T}' \times \{0,1\}^n \to \{0,1\}^n$ and let $H : \mathcal{L} \times (\{0,1\}^{n+t})^* \to \{0,1\}^n \times \mathcal{T}$ be a keyed function. For all non-negative integers n and t, let ENCODE$_{n,t}$: $\mathcal{M} \times \mathcal{D} \to (\{0,1\}^{n+t})^*$ be defined as in Algorithm 1. Then, for all $\underline{M} \in \mathcal{M}$, $d \in \mathcal{D}$, $K \in \mathcal{K}$, and $L \in \mathcal{L}$, we define HTTBC$[\widetilde{E}, H] : \mathcal{K} \times \mathcal{L} \times \mathcal{M} \times \mathcal{D} \to (\{0,1\}^n)^+$ as

$$\begin{aligned} \text{HTTBC}[\widetilde{E}, H]_{K,L}\,(\underline{M}, d) &\overset{\text{def}}{=} (U_1, \dots, U_d)\,, \text{ where} \\ M &\leftarrow \text{ENCODE}_{n+t}(\underline{M}, d), \\ (Y, X) &\leftarrow H[\widetilde{E}_K]_L(M), \text{ and} \\ (U_1, \dots, U_d) &\leftarrow \text{ZFIN}^+[\widetilde{E}_K](Y, X, d). \end{aligned}$$

While HTTBC is structurally close to HAT [CLS17], the latter requires two hash-function instances with independent keys, plus a key for the TBC, which sums up to three key in total. In contrast, HTTBC employs at most two keys: one for a hash function plus that for the TBC, which renders it is slightly more general and practical. Moreover, using a TBC-based hash function with tweak-domain separation, we effectively employ a single key for the TBC only. Most prominently, HtTBC is a VOLPRF, whereas HaT is fixed to always output a single n-bit tag.

ZMAC$^+$. Finally, ZMAC$^+$ is defined as an instantiation of HtTBC with the tweakable block cipher $\widetilde{E}_K$ and ZHash for H, and having the same key $K \in \mathcal{K}$ in all calls to $\widetilde{E}$:

$$\mathrm{ZMAC}^+[\widetilde{E}_K]\,(\underline{M}, d) \overset{\mathrm{def}}{=} \textsc{HtTBC}[\widetilde{E}_K, \mathrm{Z}\textsc{Hash}]\,(\underline{M}, d)$$

for all $\underline{M} \in \mathcal{M}$, $d \in \mathcal{D}$, and $K \in \mathcal{K}$.
Note that we employ in total three domains from $\mathcal{I}$ for domain separation: $\mathcal{I} = \{0, 1, 2\}$. We fix the domain 0 in all calls to XT in ZHash, to 1 for all calls to $\widetilde{E}$ in ZFin$^+$, and to 2 for deriving the masking keys L and R. Note that L and R have to be computed only once for every update of the TBC key K. So, all calls to $\widetilde{E}$ across the purposes of hashing, finalization, and deriving the masking keys represent independent permutations.
As a result, ZMAC$^+$ slightly reduces the number of domain integers from $\mathcal{I} = \{0, 1, \ldots, 9\}$ of the original ZMAC[ZHash], and significantly reduces it compared to the set of domains $\mathcal{J} = \{1, \ldots, 2^n - 1\}$ that was used in ZMAC[$\mathbb{ZHASH}$] (which employed the block position $i \in \mathcal{J}$ in every call to $\widetilde{E}$ to obtain independent permutations [IMPS17]).

4 Analysis of HtTBC

For the remainder, let $\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}(\mathcal{T}', \{0,1\}^n)$. Over all queries $(\underline{M}^i, d^i) \in \mathcal{M} \times \mathcal{D}$ of an adversary, we define the maximum among all second inputs as $d_{\max} \overset{\mathrm{def}}{=} \max_i d^i$. Furthermore, we define auxiliary variables $\delta \overset{\mathrm{def}}{=} \lceil \log_2(d_{\max}) \rceil \leq t$, and $\sigma' \overset{\mathrm{def}}{=} \sum_{i=1} d^i$. Let $H : \mathcal{L} \times (\{0,1\}^{n+t})^* \to \{0,1\}^n \times \mathcal{T}$.

4.1 Analysis of HtTBC for Arbitrary-Length Outputs

Theorem 1. Let $\widetilde{\pi}$ and H be defined as above. Let H be (n, t, ϵ)-tAXU and $L \twoheadleftarrow \mathcal{L}$. Then, for any VOLPRF adversary $\mathbf{A}$ on HtTBC$[\widetilde{\pi}, H]$ that makes at most q queries whose output lengths d^i sum up to at most σ' blocks in total, it holds that

$$\mathbf{Adv}^{\mathrm{VOLPRF}}_{\textsc{HtTBC}[\widetilde{\pi},H]}(\mathbf{A}) \leq \frac{(\sigma')^2 \epsilon}{2^n} + \max_{\underline{M}^1, \ldots, \underline{M}^q} \sum_{i<j}^{q} \sum_{k=0}^{d^i + d^j - 2} \mathsf{DP}_H \left[\left(\underline{M}^i, d^i\right), \left(\underline{M}^j, d^j\right), \left(0^n, \langle k \rangle_t\right) \right].$$

Remark 1. Note that if the hash function H is ϵ_2-AXU, the second summand can be simplified to $q\sigma'\epsilon_2$. However, even if ϵ_2 is not generally small for all differences, we can still obtain a good security bound, as we will show for ZHash.

Because of the structural similarities, the proof of Theorem 1 can follow a similar argumentation as Theorem 2 in [CLS17]; however, we can disregard the inequalities list used there since we consider a PRF instead of a MAC adversary, and the former has no verification oracle available. Since PRF security implies MAC security [BGM04, BKR00], we omit a MAC proof in the remainder.

Proof of Theorem 1. The queries by $\mathbf{A}$ are collected in a transcript $\tau = (L, \{(\underline{M}^i, d^i, X^i, Y^i, U^i)\}_{1 \leq i \leq q})$. In the real world, $\underline{M}^i$ denotes the i-th message, d^i the i-th input for the desired output length, and (Y^i, X^i) and $U^i = (U^i_1, \ldots, U^i_{d^i}) \leftarrow$ HtTBC$[\widetilde{\pi}, H]$ the i-th outputs of H and HtTBC, respectively. The ideal world maps inputs $(\underline{M}^i, d^i) \in \mathcal{M} \times \mathcal{D}$ to random outputs $U^i \twoheadleftarrow (\{0,1\}^n)^{d^i})$, where the outputs U^i are chosen uniformly at random of the expected length. Moreover, in the ideal world, $L \twoheadleftarrow \mathcal{L}$ is sampled independently at random from the set of all possible outputs, and $(Y^i, X^i) \leftarrow H_L(\underline{M}^i, d^i)$.

In both worlds, the queries $(\underline{M}^i, d^i)$ are answered immediately with the corresponding outputs U^i; the remaining parts of the transcript will be revealed to the adversary after it made all its queries, but before it output its decision bit that represents its guess of which world it interacted with. The task of $\mathbf{A}$ is to distinguish the real world $\mathcal{O}_{\text{real}}$ from the ideal world $\mathcal{O}_{\text{ideal}}$. A transcript τ is called *attainable* if the probability to obtain τ in the ideal world is non-zero. The set of all attainable transcripts can be partitioned into two disjoint sets GoodT and BadT. We call a transcript τ bad iff $\tau \in$ BadT, and denote it as good otherwise. A transcript is called bad if at least one of the following statements holds:

- bad$_1$: $\exists i, j \in \{1, \ldots, q\}$, $k \in \{0, \ldots, d^i - 1\}$ and $k' \in \{0, \ldots, d^j - 1\}$ s. t. $i \neq j$ and $(Y^i, X^i) \oplus (Y^j, X^j) = (0^n, \langle k \rangle_t \oplus \langle k' \rangle_t)$. Note that this can be rewritten to $(Y^i, X^i) \oplus (Y^j, X^j) = (0^n, \langle k \rangle_t)$ for some $k \in \{0, \ldots, d^i + d^j - 2\}$.
- bad$_2$: $\exists i, j \in \{1, \ldots, q\}$, $k \in \{0, \ldots, d^i - 1\}$, and $k' \in \{0, \ldots, d^j - 1\}$ s. t. $i \neq j$, $(X^i \oplus \langle k \rangle_t) = (X^j \oplus \langle k' \rangle_t)$, $U^i_{k+1} = U^j_{k'+1}$, and $Y^i \neq Y^j$.

The proof of Theorem follows then from Lemmas 2 and 3 below. The bad events represent possible input or output collisions in the finalization: bad$_1$ models the event that two pairs of state and tweak inputs $(Y^i, X^i \oplus \langle k \rangle_t) = (Y^j, X^j \oplus \langle k' \rangle_t)$ collide. Then, the corresponding outputs of the TBC would have to collide in the real world, but would collide only with negligible probability in the ideal world. The second event bad$_2$ represents the case that two pairs of tweaks and outputs of the TBC $(X^i \oplus \langle k \rangle_t, U^i_{k+1}) = (X^j \oplus \langle k' \rangle_t, U^j_{k'+1})$ collide. Then, the inputs Y^i and Y^j would have to be identical in the real world, but are so only with negligible probability in the ideal world. So, in both bad events, the adversary could easily distinguish the worlds. However, their probability to occur is sufficiently small as is studied in Lemma 2. □

Lemma 2. It holds that

$$\Pr\left[\Theta_{\text{ideal}} \in \text{BadT}\right] \leq \frac{(\sigma')^2 \epsilon}{2^n} + \max_{\underline{M}^1, \ldots, \underline{M}^q} \sum_{i<j}^{q} \sum_{k=0}^{d^i + d^j - 2} \mathsf{DP}_H \left[\left(\underline{M}^i, d^i\right), \left(\underline{M}^j, d^j\right), \left(0^n, \langle k \rangle_t\right) \right].$$

Proof. It holds that $\Pr[\Theta_{\text{ideal}} \in \text{BadT}] \leq \Pr[\mathsf{bad}_1] + \Pr[\mathsf{bad}_2]$. We upper bound the probability of those bad events in the following.

bad$_1$. In this case, it holds that $\underline{M}^i$ and $\underline{M}^j$ yielded $(Y^i, X^i) \oplus (Y^j, X^j) = (0^n, \langle k \rangle_t)$ for some $k \in \{0, \ldots, d^i + d^j - 2\}$. So, the outputs would have to be equal, and with high probability, $\mathbf{A}$ could distinguish the worlds. Over q queries, it holds that

$$\Pr\left[\mathsf{bad}_1\right] \leq \max_{\underline{M}^1, \ldots, \underline{M}^q} \sum_{i<j}^{q} \sum_{k=0}^{d^i + d^j - 2} \mathsf{DP}_H \left[\left(\underline{M}^i, d^i\right), \left(\underline{M}^j, d^j\right), \left(0^n, \langle k \rangle_t\right) \right].$$

bad$_2$. In this case, it holds that $\underline{M}^i$ and $\underline{M}^j$ yielded the same tweak input $(X^i \oplus \langle k \rangle_t) = (X^i \oplus \langle k' \rangle_t)$ for $k \in \{0, \ldots, d^i - 1\}$ and $k' \in \{0, \ldots, d^j - 1\}$, and the same $U^i_{k+1} = U^j_{k'+1}$ occurred for different inputs $Y^i \neq Y^j$. Clearly, this event can occur only in the ideal world and allows the adversary to distinguish. The probability for such a difference between X^i and X^j is upper bounded by ϵ since H is ϵ-tAXU. The probability that $U^i_{k+1} = U^j_{k'+1}$ is $\min\{d^i, d^j\}/2^n$ since for each block $X^i \oplus \langle k \rangle_t$, there is a unique mapping to exactly one block with $X^j \oplus \langle k' \rangle_t$. So, for a fixed pair of i and j, it follows

$$\Pr\left[(X^i \oplus \langle k \rangle_t) = (X^j \oplus \langle k' \rangle_t), Y^i \neq Y^j, U^i_{k+1} = U^j_{k'+1} \right] \leq \frac{(d^i \cdot d^j) \cdot \epsilon}{2^n}.$$

From the union bound over q queries, it follows that

$$\Pr\left[\mathsf{bad}_2\right] \le \frac{\epsilon}{2^n} \cdot \sum_{i<j}^{q} \left(d^i \cdot d^j\right) \le \frac{\epsilon}{2^n} \cdot \frac{1}{2} \cdot \left((\sigma')^2 - \sum_{i=1}^{q} (d^i)^2\right) \le \frac{(\sigma')^2 \cdot \epsilon}{2^n},$$

which gives our claim in Lemma 2. □

Before proceeding with the proof of good transcripts, we formulate a short fact that will be useful in turn. Since its proof follows from simple arithmetic, we leave its verification to the interested reader.

Fact 1. Let $u_1, \ldots, u_r$ and $v_1, \ldots, v_s$ be positive integers such that it holds

$$\sum_{i=1}^{r} u_i = \sum_{j=1}^{s} v_j, \tag{1}$$

$$r \le s, \quad \text{and} \tag{2}$$

$$v_i \le u_i, \quad \text{for all } 1 \le i \le r. \tag{3}$$

Then, it holds for any positive integer $N \ge \sum_{i=1}^{r} u_i$ that

$$\prod_{i=1}^{r} (N)_{u_i} \le \prod_{i=1}^{s} (N)_{v_i} \qquad \text{and thus} \qquad \prod_{i=1}^{r} \frac{1}{(N)_{u_i}} \ge \prod_{i=1}^{s} \frac{1}{(N)_{v_i}}.$$

Lemma 3. It holds that

$$\frac{\Pr\left[\Theta_{\mathrm{real}} = \tau\right]}{\Pr\left[\Theta_{\mathrm{ideal}} = \tau\right]} \ge 1.$$

Proof. Let $\tau \in \textsc{GoodT}$. We consider the set of all tweaks

$$\left\{X^i \oplus \langle k \rangle_t\right\}_{\substack{1 \le i \le q \\ 0 \le k < d^i}}$$

that occurred over all output blocks of all queries of the transcript. We rewrite this set as $\{\mathsf{X}^1, \ldots, \mathsf{X}^r\}$ with $r \le \sigma'$ s. t. all reordered tweaks X^i are pairwise distinct. Further, we define by q_i, for $1 \le i \le r$, the number of queries for which $H_L(\underline{M}^i)$ produced X^i; naturally, it holds that $\sum_i^r q_i = \sigma'$. Since hash key and outputs are sampled uniformly at random and independently from each other in the ideal world, it holds that

$$\Pr\left[\Theta_{\mathrm{ideal}} = \tau\right] = \frac{1}{|\mathcal{L}|} \cdot \prod_{i=1}^{q} \frac{1}{(2^n)_{d^i}}.$$

So, we can focus on lower bounding the probability of obtaining τ in the real world. Therein, the key L is also sampled uniformly at random from $\mathcal{L}$, and we can concentrate on the probability to obtain the query results. For this purpose, we adopt the notion of transcript-compatible permutations from [CS14]. We call $\widetilde{\pi}$ *compatible* with τ if for all $1 \le i \le q$ and for all $1 \le k \le d^i$, it holds that

$$\widetilde{\pi}(X^i \oplus \langle k-1 \rangle_t, Y^i) = U^i_k.$$

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

$$\Pr\left[\Theta_{\mathrm{real}} = \tau\right] = \frac{1}{|\mathcal{L}|} \cdot \Pr\left[\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}\left(\mathcal{T}', \{0,1\}^n\right) : \widetilde{\pi} \in \mathsf{Comp}(\tau)\right].$$

Over all tweaks X^i, for $1 \leq i \leq r$, the fraction of compatible permutations is given by

$$\prod_{i=1}^{r} \prod_{j=1}^{q_i} \frac{1}{2^n - (j-1)} = \prod_{i=1}^{r} \frac{1}{(2^n)_{q_i}}. \tag{4}$$

We can construct a sum of blocks in the ideal world as

$$\sum_{i=1}^{q} \sum_{j=1}^{d^i} 1 = \sum_{i=1}^{\sigma'} 1,$$

where the 1's represent the summands v_j in Fact 1. Using the combined knowledge of

$$\begin{aligned} \sum_{i=1}^{r} q_i = \sum_{i=1}^{\sigma'} 1 &= \sigma', \\ r \leq \sigma', &\quad \text{and} \\ v_j = 1 \leq q_j, &\quad \text{for all } 1 \leq j \leq r, \end{aligned}$$

allows us to directly apply Fact 1 and obtain

$$\prod_{i=1}^{r} \frac{1}{(2^n)_{q_i}} \geq \prod_{j=1}^{\sigma'} \frac{1}{(2^n)_{v_j}} = \prod_{j=1}^{\sigma'} \frac{1}{2^n} = \prod_{j=1}^{q} \frac{1}{(2^n)^{d^j}}.$$

Multiplying both sides with $1/|\mathcal{L}|$ yields the probabilities to obtain τ in the real world on the left-hand side, and in the ideal world on the right-hand side:

$$\Pr\left[\Theta_{\text{real}} = \tau\right] \geq \Pr\left[\Theta_{\text{ideal}} = \tau\right].$$

Our claim in Lemma 3 follows. □

4.2 Analysis of HtTBC for Single-Block Outputs

Before we study the terms of ϵ and DP in Theorem 1 for ZMAC$^+$, we derive a small corollary when the output length is limited to only a single block for each query.

Corollary 1. Let H be ϵ_1-tAXU and ϵ_2-AU. Let $L \twoheadleftarrow \mathcal{L}$, $\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}(\mathcal{T}', \{0,1\}^n)$, and let $d_{\max} = 1$. Then, it holds for any PRF adversary on $\textsc{HtTBC}[\widetilde{\pi}, H]$ that makes at most q queries that

$$\mathbf{Adv}^{\text{PRF}}_{\textsc{HtTBC}[\widetilde{\pi},H]}(\mathbf{A}) \leq \binom{q}{2} \cdot \left(\frac{2\epsilon_1}{2^n} + \epsilon_2\right).$$

The proof of Corollary 1 can be conducted similarly as that of Theorem 1 and is therefore omitted. The core observation is that all queries are restricted to a single output block, and therefore bad_2 represents the event of a collision of $(Y^i, X^i) = (Y^j, X^j)$ for $i \neq j$; hence, its probability is at most $q^2/2 \cdot \epsilon_2$.

5 DP-Analysis of ZHash

This section bounds the probability of output differences of the form $(0, \langle k \rangle_t)$ for integers k for ZHASH.

Theorem 2. Let $\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}(\mathcal{T}', \{0,1\}^n)$. Let $(\underline{M}, d), (\underline{M}', d') \in \mathcal{M} \times \mathcal{D}$ be distinct and $M \leftarrow \textsc{Encode}_{n,t}(\underline{M}, d)$ and $M' \leftarrow \textsc{Encode}_{n,t}(\underline{M}, d)$ s. t. $M, M' \in \{0,1\}^{n+t}$ have at most m and m' $(n+t)$-bit blocks each, where $1 \leq m \leq m' < 2^{\min\{n,t\}-2}$. Then

$$\sum_{k=0}^{d+d'-2} \mathsf{DP}_{\mathrm{Z}\textsc{Hash}[\widetilde{\pi}]} \left[M, M' \left(0^n, \langle k \rangle_t\right)\right] \leq \begin{cases} \frac{2(d+d')}{2^n} & \text{if } C1, \\ \frac{2(m+m'+1)}{2^{n+\min\{n,t\}}} + \frac{4(d+d')}{2^{n+\min\{n,t\}}} & \text{otherwise.} \end{cases}$$

The Boolean variable $C1$ is true iff $m = m'$ and there exists $s \in \{1, \ldots, m\}$ s. t. $M[s] \neq M'[s]$ and $M[i] = M'[i]$ for all $i \neq s$.

Proof. The proof follows a similar strategy as the collision bound proof of ZHash by Iwata et al. [IMPS17]. Let $M = (M[1], \ldots, M[m])$ and $M' = (M'[1], \ldots, M'[m'])$, be two distinct messages after the encoding, which ensures that m and m' cannot be 0. Further, let $(Y, X) = \mathrm{Z}\textsc{Hash}[\widetilde{\pi}](M)$ and $(Y', X') = \mathrm{Z}\textsc{Hash}[\widetilde{\pi}](M')$ denote their respective outputs. W.l.o.g., we assume $m \leq m'$. We denote the blockwise components $M_L[i]$, $M_R[i]$, $M'_L[i]$, $M'_R[i]$, $S_i = 2^{i-1}L \oplus M_L[i]$, $T_i = 2^{i-1}R \oplus M_R[i]$, $Y_i = \widetilde{\pi}^{T_i}(S_i)$, and $X_i = M_R[i] \oplus Y_i$. The corresponding variables S'_i, T'_i, Y'_i, and X'_i are defined analogously. Moreover, we write their differences as $\Delta X = X \oplus X'$, $\Delta Y = Y \oplus Y'$, etc; furthermore, we define the factors of the bottom lane in ZHash by $\lambda_i \overset{\text{def}}{=} 2^{m+1-i}$ and $\lambda'_i \overset{\text{def}}{=} 2^{m'+1-i}$ when m and m' are clear from the context. In the following, we distinguish between two mutually exclusive scenarios: $t \leq n$ and $t > n$ and consider four cases in each scenarios. All together, they represent all possible options.

Scenario $t \leq n$. We start with the scenario that $t \leq n$.

Case 1: $m = m'$ and there exists $s \in \{1, \ldots, m\}$ s.t. $M[s] \neq M'[s]$ and $M[i] = M'[i]$ for all $i \neq s$. In this case, it holds that

$$\Delta X = \bigoplus_{i=1}^{m} \Delta X_i = \Delta X_s \quad \text{and} \quad \Delta Y = \bigoplus_{i=1}^{m} \lambda_i \cdot \Delta Y_i = \lambda_s \cdot \Delta Y_s.$$

Since $\Delta X_s = \mathrm{MSB}_t(\Delta Y_s)$, we can rewrite our condition as

$$\begin{cases} \Delta X_s &= \langle k \rangle_t \\ \Delta Y_s &= 0^n \end{cases} \iff \begin{cases} \mathrm{MSB}_t\left(\Delta Y_s\right) \oplus \Delta M_R[s] &= \langle k \rangle_t \\ \Delta Y_s &= 0^n. \end{cases}$$

So, this case holds iff $\Delta Y_s = 0^n$ and $\Delta M_R[s] = \langle k \rangle_t$. Any other option, e. g., $\Delta M_R[s] \neq \langle k \rangle_t$ results in zero probability in this case. The fact that $\Delta Y_s = 0^n$ implies that the inputs $M_R[s]$ and $M'_R[s]$ must differ. So, since we sample Y_s and Y'_s from independent permutations, the probability to obtain $\Delta Y_s = 0^n$ is $1/2^n$, independent from the tweak length t. For fixed k, it follows for this case that

$$\mathsf{DP}_{\mathrm{Z}\textsc{Hash}[\widetilde{\pi}]} \left[M, M', \left(0^n, \langle k \rangle_t\right)\right] \leq \frac{1}{2^n}.$$

Note that, since all cases are mutually exclusive, the remaining cases assume that Case 1 does not occur. In the remaining cases 2 through 4, we first regard arbitrary differences $(\nabla Y, \nabla X) \in \{0,1\}^n \times \{0,1\}^t$, and will concentrate on differences $(0, \langle k \rangle_t)$ later.

Case 2: $m = m'$ and there exist $r, s \in \{1, \ldots, m\}$ s.t. $r \neq s$, $M[r] \neq M'[r]$, and $M[s] \neq M'[s]$. Since there may exist more than two blocks where M and M' differ, we fix r and s to denote the two smallest distinct indices where $M[r] \neq M'[r]$, and

$M[s] \neq M'[s]$. For both M and M', we fix all other values Y_i and Y'_i, for all $1 \leq i \leq m$, $i \neq r$, and $i \neq s$. In this case, it holds that

$$\begin{aligned} \Delta X &= \Delta X_r \oplus \Delta X_s \oplus \Delta_1, & \Delta_1 &\overset{\text{def}}{=} \nabla X \oplus \textstyle\bigoplus_{1 \leq i \leq m, i \notin \{r,s\}} \Delta X_i, \\ \Delta Y &= \lambda_r \cdot \Delta Y_r \oplus \lambda_s \cdot \Delta Y_s \oplus \Delta_2, & \Delta_2 &\overset{\text{def}}{=} \nabla Y \oplus \textstyle\bigoplus_{1 \leq i \leq m, i \notin \{r,s\}} \lambda_i \cdot \Delta Y_i. \end{aligned}$$

As in the AU proof by Iwata et al., the terms Δ_1 and Δ_2 are XOR-sums of variables ΔX_i and ΔY_i which are determined by sums of random variables Y_i and Y'_i, and the sums of message blocks $M_R[i]$ and $M'_R[i]$, respectively. We can rewrite the above to

$$\begin{cases} \Delta X = \nabla X \\ \Delta Y = \nabla Y \end{cases} \Longleftrightarrow \begin{cases} \mathrm{MSB}_t\left(\Delta Y_r \oplus \Delta Y_s\right) = \Delta_3 \\ \lambda_r \cdot \Delta Y_r \oplus \lambda_s \cdot \Delta Y_s = \Delta_2, \end{cases}$$

for $\Delta_3 = \Delta_1 \oplus \Delta M_R[r] \oplus \Delta M_R[s]$. We can further transform it to

$$\Pr\begin{bmatrix} \Delta X = \nabla X, \\ \Delta Y = \nabla Y \end{bmatrix} \leq \max_{\substack{\Delta_3 \in \{0,1\}^t \\ \Delta_2 \in \{0,1\}^n}} \sum_{\substack{\Delta_4 \in \{0,1\}^n \\ \mathrm{MSB}_t(\Delta_4) = \Delta_3}} \Pr\begin{bmatrix} \Delta Y_r \oplus \Delta Y_s = \Delta_4, \\ \lambda_r \cdot \Delta Y_r \oplus \lambda_s \cdot \Delta Y_s = \Delta_2 \end{bmatrix}.$$

As stated by Iwata et al., for any Δ_2 and Δ_4, the equational system above has a unique solution since $r \neq s$ and $\lambda_r \neq \lambda_s$:

$$\begin{aligned} \Delta Y_r &= (\lambda_s \Delta_4 \oplus \Delta_2) \cdot (\lambda_r \oplus \lambda_s)^{-1} \\ \Delta Y_s &= \Delta_4 \oplus \Delta Y_r. \end{aligned}$$

We cannot assume that the variables ΔY_r and ΔY_s are independent or result from random tweaked permutations. However, we can distinguish between two subcases, namely whether the tuple (S_r, T_r) is unique or not. We define a Boolean variable $\mathsf{STColl}(r)$ that, for an index r, is true iff there exists some $i \neq r$ s. t. $(S_r, T_r) = (S_i, T_i)$ or $(S_r, T_r) = (S'_i, T'_i)$. If $\mathsf{STColl}(r)$ is false, then we call the tuple (S_r, T_r) fresh and non-fresh otherwise. Assuming that $t \geq n$, it holds that

$$\Pr\left[\mathsf{STColl}(r)\right] \leq \frac{(m+1) + (m'+1) - 1}{2^{n+t}} = \frac{m + m' + 1}{2^{n+t}},$$

where the additional block in $m+1$ and $m'+1$, respectively, stems from the encoding of the output length d. A similar argument can be formulated for an event $\mathsf{STColl}(s)$ that is true if (S_s, T_s) is not fresh. We define composite random variables $\mathsf{STColl}(i, j) = \mathsf{STColl}(i) \vee \mathsf{STColl}(j)$, which are true iff any or both of two blocks of interest is not fresh, for given $i, j \in \{1, \ldots, m'\}$. So, the probability that any of these two events occurs, i. e., that $\mathsf{STColl}(r, s)$ is true, can be upper bounded by $\frac{2(m+m'+1)}{2^{n+t}}$, and we can focus on the case that (S_r, T_r) and (S_s, T_s) are fresh. Then, the point probabilities of Y_r and that for Y_s are $1/(2^n - (m + m' + 1))$ each. It follows that

$$\begin{aligned} \Pr\begin{bmatrix} \Delta X = \nabla X, \\ \Delta Y = \nabla Y \end{bmatrix} &\leq \frac{2(m+m'+1)}{2^{n+t}} + \left(\max_{\substack{\Delta_3 \in \{0,1\}^t \\ \Delta_2 \in \{0,1\}^n}} \sum_{\substack{\Delta_4 \in \{0,1\}^n \\ \mathrm{MSB}_t(\Delta_4) = \Delta_3}} \frac{1}{(2^n - (m+m'+1))^2} \right) \\ &\leq \frac{2(m+m'+1)}{2^{n+t}} + \frac{2^{n-t}}{(2^n - (m+m'+1))^2} \\ &\leq \frac{2(m+m'+1)}{2^{n+t}} + \frac{4}{2^{n+t}}, \end{aligned}$$

using the assumption that $m + m' < 2^{n-2}$.

Case 3: $m' = m + 1$. Here, we can isolate the blocks $M[m]$, $M'[m]$, and $M'[m+1]$:

$$\begin{aligned} \Delta X &= X'_{m+1} \oplus X'_m \oplus X_m \oplus \Delta_1, & \Delta_1 &\overset{\text{def}}{=} \nabla X \oplus \textstyle\bigoplus_{i=1}^{m-1} \Delta X_i, \\ \Delta Y &= \lambda'_{m+1} \cdot Y'_{m+1} \oplus (\lambda'_m \oplus \lambda_m) \cdot \Delta Y_m \oplus \Delta_2, & \Delta_2 &\overset{\text{def}}{=} \nabla Y \oplus \textstyle\bigoplus_{i=1}^{m-1} (\lambda'_i \oplus \lambda_i) \cdot \Delta Y_i. \end{aligned}$$

It holds that $\lambda'_{m+1} = \lambda_m = 2$ and $\lambda'_m = 2^2$. The terms Δ_1 and Δ_2 are again XOR sums of variables ΔX_i, ΔY_i, that themselves are determined by sums of random variables Y_i, Y'_i, and the sums of message blocks $M_R[i]$ and $M'_R[i]$, respectively. So, we can derive that

$$\begin{cases} \Delta X &= \nabla X \\ \Delta Y &= \nabla Y \end{cases} \Longleftrightarrow \begin{cases} \text{MSB}_t\left(Y'_{m+1} \oplus Y'_m \oplus Y_m\right) &= \Delta_3 \\ 2\left(Y'_{m+1} \oplus 2Y'_m \oplus Y_m\right) &= \Delta_2, \end{cases}$$

for $\Delta_3 = \Delta_1 \oplus M'_R[m+1] \oplus \Delta M_R[m]$. It follows that

$$\Pr\begin{bmatrix} \Delta X = \nabla X \\ \Delta Y = \nabla Y \end{bmatrix} \leq \max_{\substack{\Delta_1 \in \{0,1\}^t \\ \Delta_2 \in \{0,1\}^n}} \sum_{\substack{\Delta_3 \in \{0,1\}^n \\ \text{MSB}_t(\Delta_3) = \Delta_1}} \Pr\begin{bmatrix} Y'_{m+1} \oplus Y'_m \oplus Y_m = \Delta_3 \\ 2\left(Y'_{m+1} \oplus 2Y'_m \oplus Y_m\right) = \Delta_2 \end{bmatrix}$$

By substituting $A = Y'_{m+1} \oplus Y_m$ and $B = Y'_m$ and $\Delta_4 = 2^{-1}\Delta_2$, we obtain

$$\begin{cases} A \oplus B &= \Delta_3 \\ A \oplus 2B &= \Delta_4 \end{cases}$$

Again, we can distinguish whether (S'_{m+1}, T'_{m+1}) is fresh or not. For a positive integer r, we define Boolean variables $\mathsf{STColl}'(r)$ similarly as $\mathsf{STColl}(r)$. $\mathsf{STColl}'(r)$ is true iff there exists no i s. t. $(S'_r, T'_r) = (S'_i, T'_i)$ or $(S'_r, T'_r) = (S_i, T_i)$. We can rewrite the equational system above to

$$\Pr\begin{bmatrix} A \oplus B &= \Delta_3 \\ A \oplus 2B &= \Delta_4 \end{bmatrix} \leq \Pr\left[\begin{matrix} A \oplus B &= \Delta_3 \\ A \oplus 2B &= \Delta_4 \end{matrix} \middle| \neg\mathsf{STColl}'(m+1)\right] + \Pr\left[\mathsf{STColl}'(m+1)\right].$$

In the latter case, it holds that

$$\Pr\left[\mathsf{STColl}'(m+1)\right] \leq \frac{m + m' + 1}{2^{n+t}}.$$

So, we can concentrate on the former case in the remainder of this case. If (S'_{m+1}, T'_{m+1}) is fresh, the point probability of Y'_{m+1} is at most $1/(2^n - (m + m' + 1))$. The equational system has a unique solution (A, B) over $\mathbb{F}_{2^n}$. The probability that B is the correct value is at most $1/(2^n - (m+m'+1))$. Since A contains Y'_{m+1} and B does not, A is independent from B under the assumption of $\neg\mathsf{STColl}'(m+1)$. The probability that Y has the correct value to fulfill both equations is at most $1/(2^n - (m+m'+1))$. Hence, using $m+m' < 2^{n-2}$ and summing over at most 2^{n-t} values Δ_3 s. t. $\text{MSB}_t(\Delta_3) = \Delta_1$, the probability becomes

$$\Pr\begin{bmatrix} \Delta X = \nabla X \\ \Delta Y = \nabla Y \end{bmatrix} \leq \frac{m + m' + 1}{2^{n+t}} + \frac{2^{n-t}}{(2^n - (m + m' + 1))^2} \leq \frac{m + m' + 1}{2^{n+t}} + \frac{4}{2^{n+t}}.$$

Case 4: $m' \geq m + 2$. We can isolate the blocks $M'_{m'}$ and $M'_{m'-1}$ and obtain

$$\begin{aligned} \Delta X &= X'_{m'} \oplus X'_{m'-1} \oplus \Delta_1, & \Delta_1 &\overset{\text{def}}{=} \nabla X \oplus \textstyle\bigoplus_{i=1}^{m} \Delta X_i \\ \Delta Y &= \lambda'_{m'} \cdot Y'_{m'} \oplus \lambda'_{m'-1} \cdot Y'_{m'-1} \oplus \Delta_2, & \Delta_2 &\overset{\text{def}}{=} \nabla Y \oplus \textstyle\bigoplus_{i=1}^{m} \lambda_i \cdot Y_i \oplus \lambda'_i \cdot Y'_i. \end{aligned}$$

Again, we can upper bound the probability that $(S'_{m'}, T'_{m'})$ or $(S'_{m'-1}, T'_{m'-1})$ are non-fresh by $\frac{2(m+m'+1)}{2^{n+t}}$ and focus on the opposite case in the remainder. We obtain an equational system similar to the one in Case 3. Following a similar argumentation that the point probability of $Y'_{m'}$ and $Y'_{m'-1}$ is at most $1/(2^n - (m+m'+1))$ each, and using $m+m' < 2^{n-2}$, the collision probability in this case becomes

$$\Pr\begin{bmatrix}\Delta X = \nabla X \\ \Delta Y = \nabla Y\end{bmatrix} \leq \frac{2(m+m'+1)}{2^{n+t}} + \frac{2^{n-t}}{(2^n - (m+m'+1))^2} \leq \frac{2(m+m'+1)}{2^{n+t}} + \frac{4}{2^{n+t}}.$$

For all cases except Case 1, we obtained an upper bound on the differential probability for $t \leq n$ and arbitrary $(\nabla Y, \nabla X)$, that is also an upper bound for specific differences of the form $(0^n, \langle k \rangle_t)$:

$$\Pr\left[\begin{matrix}\Delta X = \langle k \rangle_t \\ \Delta Y = 0^n\end{matrix} \middle| \neg C1\right] \leq \Pr\left[\begin{matrix}\Delta X = \nabla X \\ \Delta Y = \nabla Y\end{matrix} \middle| \neg C1\right] \leq \frac{2(m+m'+1)}{2^{n+t}} + \frac{4}{2^{n+t}}.$$

Scenario $t > n$. It remains to consider the Scenario $t > n$. In Case 1, M and M' differ in exactly one block s. The same argumentation as in Case 1 for the Scenario $t \leq n$ applies also here: the tweak inputs to $\widetilde{\pi}$ must differ in order to produce $(0, \langle k \rangle_t)$. And the bound for $Y_r = Y'_r$ is again at most $1/2^n$ since the permutations used to process $M[s]$ and $M'[s]$ are independent. In Case 2, it holds that $m = m'$ and there exist two indices $r \neq s$ s. t $M[r] \neq M'[r]$ and $M[s] \neq M'[s]$. The reasoning is analogous to the corresponding case when $t \leq n$ holds, and the bound differs only in the sense that we do not limit our interest to the most significant t bit of Y. So, the probability in this case is upper bounded by

$$\frac{2(m+m'+1)}{2^{n+t}} + \frac{1}{(2^n - (m+m'+1))^2} \leq \frac{2(m+m'+1)}{2^{n+t}} + \frac{4}{2^{2n}}.$$

A similar statement and the same bound holds for the cases when $m' \geq m$. The bound in Theorem 2 follows from summing over at most $d + d' - 1$ possible differences $(0^n, \langle k \rangle_t)$ for the case when the blocks of interest are fresh. □

It remains to derive the bound over the sequence of all q messages in the following.

Lemma 4. Let $\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}(\mathcal{T}', \{0,1\}^n)$. Given q pairwise distinct tuples $(\underline{M}^i, d^i) \in \mathcal{M} \times \mathcal{D}$ and their encodings $M^i \leftarrow \textsc{Encode}_{n,t}(\underline{M}^i, d^i)$ s. t. all M^i consist of less than $2^{\min\{n,t\}-3}$ $(n+t)$-bit blocks each, and of at most σ blocks in total, and s. t. $\sum_{i=1}^{q} d^i \leq \sigma'$. Then

$$\max_{M^1,\ldots,M^q} \sum_{i<j}^{q} \sum_{k=0}^{d^i+d^j-2} \mathsf{DP}_{\textsc{ZHash}[\widetilde{\pi}]}\left[M^i, M^j, (0^n, \langle k \rangle_t)\right] \leq \frac{2\sigma'}{2^n} + \frac{2(q-1)\sigma + q^2 + 4(q-1)\sigma'}{2^{n+\min\{n,t\}}}.$$

Proof. For Case 1, it holds that $m^i = m^j$ and there exists $s \in \{1, \ldots, m^i\}$ s. t. $M^i[s] \neq M^j[s]$ and $M^i[i] = M^j[i]$ for all $i \neq s$. As described in the proof of Theorem 2, it must hold in this case that $\Delta Y_s = 0^n$ and $\Delta M_R[s] = \langle k \rangle_t$. We reorder the query indices such that $d^1 \leq d^2 \leq \ldots \leq d^q$ holds. For each message $M^i_R[s]$, and any fixed k, there is at most one message $M^j_R[s]$ (and, since all other blocks are equal, for each message M^i and fixed k, there is at most one message M^j) that can produce the desired difference. This applies to every $k \leq d + d' \leq 2d_{\max}$. So, for a fixed message M and a fixed block index s, there are at most $d + d'$ messages that can produce our difference. It follows that

$$\begin{aligned}&\max_{M^1,\ldots,M^q} \sum_{i<j}^{q} \sum_{k=0}^{d^i+d^j-2} \mathsf{DP}_{\textsc{ZHash}[\widetilde{\pi}]}\left[M^i, M^j, (0^n, \langle k \rangle_t)\right] \\ \leq &\max_{M^1,\ldots,M^q} \sum_{i=1}^{q} \sum_{k=0}^{2(d^i-1)} \frac{1}{2^n} \leq \max_{M^1,\ldots,M^q} \sum_{i=1}^{q} \frac{2d^i}{2^n} \leq \frac{2\sigma'}{2^n}.\end{aligned}$$

Concerning the remaining terms of the other cases but Case 1, we can rewrite the bound over all message pairs as

$$\sum_{i<j}^{q}\left(\frac{2(m^i+m^j+1)}{2^{n+\min\{n,t\}}}+\frac{4(d^i+d^j)}{2^{n+\min\{n,t\}}}\right) \le \frac{2(q-1)\sigma}{2^{n+\min\{n,t\}}}+\sum_{i<j}^{q}\left(\frac{2}{2^{n+\min\{n,t\}}}+\frac{4(d^i+d^j)}{2^{n+\min\{n,t\}}}\right)$$
$$\le \frac{2(q-1)\sigma+q^2+4(q-1)\sigma'}{2^{n+\min\{n,t\}}},$$

where we used

$$\sum_{i<j}^{q}(m^i+m^j)=(q-1)\sigma \quad \text{and} \quad \sum_{i<j}^{q}(d^i+d^j)=(q-1)\sigma'.$$

The sum of all terms yields then our bound in Lemma 4. □

6 tAXU-Analysis of ZHash

Theorem 3. Let $\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}(\mathcal{T}', \{0,1\}^n)$. Let $M, M' \in (\{0,1\}^{n+t})^*$ be distinct and consist of at most m and m' $(n+t)$-bit blocks, respectively, with $1 \le m \le m' < 2^{\min\{n,t\}-3}$. Then, ZHASH$[\widetilde{\pi}]$ is (n,t,ϵ)-tAXU for

$$\epsilon \le \frac{2(m+m'+1)}{2^{n+\min\{n,t\}}}+\frac{4}{2^{\min\{n,t\}}}.$$

Proof. The proof follows a similar strategy as that of Theorem 2, where we consider the same two scenarios (1) $t \le n$ and (2) $t > n$, as well as the same four cases for each scenario as in the previous two sections. Though, this time, we are interested in upper bounding

$$\max_{\nabla X \in \{0,1\}^t} \Pr\left[\Delta X = \nabla X\right].$$

Scenario $t \le n$. Again, we start with the scenario $t \le n$. Though, we consider Case 2 first. We adopt the random variables $\mathsf{STColl}(i)$ and the composite random variables $\mathsf{STColl}(i,j) = \mathsf{STColl}(i) \vee \mathsf{STColl}(j)$ from the previous section; more precisely, in Case 2, $\mathsf{STColl}(r,s)$ is true iff (S_r, T_r) and/or (S_s, T_s) are not fresh. Again, r and s denote the smallest indices in Case 2 for which $M_r \ne M'_r$ and $M_s \ne M'_s$ holds. The probability that $\mathsf{STColl}(r,s)$ is true is at most $2(m+m'+1)/2^{n+t}$. For the (n,t,ϵ)-tAXU bound, we allow arbitrary differences ΔY. So, for Case 2, it follows that

$$\max_{\nabla X \in \{0,1\}^t} \Pr\left[\Delta X = \nabla X\right] \le \Pr\left[\mathsf{STColl}(r,s)\right] + \max_{\nabla X \in \{0,1\}^t} \Pr\left[\Delta X = \nabla X \,|\, \neg\mathsf{STColl}(r,s)\right]$$
$$\le \frac{2(m+m'+1)}{2^{n+t}}+\frac{4\cdot 2^n}{2^{n+t}} = \frac{2(m+m'+1)}{2^{n+t}}+\frac{4}{2^t}.$$

In Case 4, we use instead a random variable $\mathsf{STColl}(m', m'-1)$ that is true iff $(S_{m'}, T_{m'})$ and/or $(S_{m'-1}, T_{m'-1})$ are not fresh. Its probability and that for any difference is equal to those in Case 2. So, the same upper bound as in Case 2 holds also in Case 4.
It remains to consider Cases 1 and 3. We continue with Case 1, where $m = m'$ and there exists $s \in \{1, \ldots, m\}$ s.t. $M[s] \ne M'[s]$ and $M[i] = M'[i]$ for all $i \ne s$. We define $\mathsf{STColl}(s)$ to be true iff (S_s, T_s) is not fresh, and upper bound its probability by $(m+m'+1)/2^{n+t}$. We can then assume it is fresh in the remainder, and the point probability of Y_s for any point is at most $1/(2^n - (m+m'+1))$. In this case,

$$\Delta X_s = \mathrm{MSB}_t\left(\Delta Y_s\right) \oplus \Delta M_R[s].$$

from which it follows that for fixed messages M, and M',

$$\max_{\nabla X \in \{0,1\}^t} \Pr\left[\Delta X = \nabla X\right] = \max_{\nabla X \in \{0,1\}^t} \sum_{\substack{\Delta_1 \in \{0,1\}^n \\ \text{MSB}_t(\Delta_1) = \nabla X \oplus \Delta M_R[s]}} \Pr\left[\Delta Y = \Delta_1\right].$$

Hence,

$$\begin{aligned}
&\max_{\nabla X \in \{0,1\}^t} \Pr\left[\Delta X = \nabla X\right] \\
&\leq \Pr\left[\mathsf{STColl}(s)\right] + \max_{\nabla X \in \{0,1\}^t} \Pr\left[\Delta X = \nabla X \mid \neg \mathsf{STColl}(s)\right] \\
&\leq \Pr\left[\mathsf{STColl}(s)\right] + \max_{\nabla X \in \{0,1\}^t} \sum_{\substack{\Delta_1 \in \{0,1\}^n \\ \text{MSB}_t(\Delta_1) = \nabla X \oplus \Delta M_R[s]}} \Pr\left[\Delta Y = \Delta_1\right] \\
&\leq \frac{m + m' + 1}{2^{n+t}} + \frac{2^{n-t}}{2^n - (m + m' + 1)} \leq \frac{m + m' + 1}{2^{n+t}} + \frac{2}{2^t},
\end{aligned}$$

for $m + m' < 2^{n-1}$.
In Case 3, we can apply a similar argument as in Case 1. In Case 3, M' exceeds M by one block, $M'[m+1]$. So, we can distinguish whether $\mathsf{STColl}'(m+1)$ holds or not. Again, we obtain an upper bound of

$$\max_{\nabla X \in \{0,1\}^t} \Pr\left[\Delta X = \nabla X\right] \leq \frac{m + m' + 1}{2^{n+t}} + \frac{2^{n-t}}{2^n - (m + m' + 1)} \leq \frac{m + m' + 1}{2^{n+t}} + \frac{2}{2^t}.$$

Scenario $t > n$. The argumentation is similar for the scenario when $t > n$. In Case 1, we bound again the probability that (S_s, T_s) is not fresh and consider the opposite case.

$$\frac{m + m' + 1}{2^{2n}} + \frac{1}{2^n - (m + m' + 1)} \leq \frac{m + m' + 1}{2^{2n}} + \frac{2}{2^n}.$$

In Case 3, we bound the probability that (S'_{m+1}, T'_{m+1}) is fresh or not and obtain the same upper bound as in Case 1. In the remaining cases, the probability is at most

$$\frac{2(m + m' + 1)}{2^{2n}} + \frac{1}{(2^n - (m + m' + 1))^2} \leq \frac{2(m + m' + 1)}{2^{2n}} + \frac{4}{2^n}.$$

Our bound in Theorem 3 follows. □

Corollary 2. *Let $\widetilde{E}_K$ be defined as in Section 3, and let $K \twoheadleftarrow \mathcal{K}$. Let $\mathbf{A}$ be a* VOLPRF *adversary on* ZMAC$^+[\widetilde{E}]$ *that runs in time at most* TIME*, makes at most q queries of at most $m \leq 2^{\min\{n,t\}-3}$ $(n+t)$-bit blocks each and at most σ blocks in total, and whose output lengths d^i sum up to at most σ'. Then, it holds that*

$$\begin{aligned}
\mathbf{Adv}^{\text{VOLPRF}}_{\text{ZMAC}^+[\widetilde{E}]}(\mathbf{A}) \leq \frac{(\sigma')^2}{2^n} \cdot \left(\frac{4m+2}{2^{n+\min\{n,t\}}} + \frac{4}{2^{\min\{n,t\}}}\right) + \frac{2\sigma'}{2^n} + \\
\frac{2(q-1)\sigma + q^2 + 4(q-1)\sigma'}{2^{n+\min\{n,t\}}} + \mathbf{Adv}^{\text{TPRP}}_{\widetilde{E}}(\mathbf{A}'),
\end{aligned}$$

where $\mathbf{A}'$ is a TPRP *adversary on $\widetilde{E}$ that asks at most $\sigma + q + \sigma' + 2$ queries to its oracles and runs in time at most* TIME $+ O(\sigma + q + \sigma' + 2)$.

The bound in Corollary 2 follows from (1) using a standard argument to replace $\widetilde{E}$ with $\widetilde{\pi} \twoheadleftarrow \widetilde{\mathsf{Perm}}(\mathcal{T}', \{0,1\}^n)$, and (2) applying and condensing the bounds from Theorems 1, 2, 3, and Lemma 4. Below, we derive a bound where each query outputs $2n$ bit for comparability of the security of ZMAC$^+$ with that of ZMAC.

Table 2: Estimated performance values in cycles/byte for processing a long message M ($\geq$ 64 kByte) with ZMAC$^+[\widetilde{E}_K]$ instantiations for selected tweakable block ciphers on Intel Skylake with AES-NI enabled.

	Output length	
TBC $\widetilde{E}_K$	n bit	$\lvert M\rvert$ bit
Deoxys-BC-256	0.62	1.49
Deoxys-BC-384	0.61	1.60
Skinny-128/256	2.08	6.20
Skinny-128/384	1.62	6.42

Corollary 3. *Let $d^i = 2$ for $1 \leq i \leq q$, and all further assumptions as in Corollary 2. Then, it holds that*

$$\mathbf{Adv}^{\text{VOLPRF}}_{\text{ZMAC}^+[\widetilde{E}]}(\mathbf{A}) \leq \frac{4q^2}{2^n} \cdot \left(\frac{4m+2}{2^{n+\min\{n,t\}}} + \frac{4}{2^{\min\{n,t\}}} \right) + \frac{4q}{2^n} + \frac{2(q-1)\sigma + q^2 + 8(q-1)q}{2^{n+\min\{n,t\}}} + \mathbf{Adv}^{\text{TPRP}}_{\widetilde{E}}(\mathbf{A}'),$$

where $\mathbf{A}'$ is a TPRP *adversary on $\widetilde{E}$ that asks at most $\sigma + 3q + 2$ queries to its oracles and runs in time at most* TIME $+ O(\sigma + 3q + 2)$.

7 Potential Instantiations

The appropriate instantiations that were suggested for ZMAC [IMPS17] apply naturally also to ZMAC$^+$: one can imagine the same speed as ZMAC when instantiated with dedicated tweakable block ciphers, e. g., the Deoxys-BC variants [JNP16] Deoxys-BC-256 or Deoxys-BC-384, or the Skinny versions [BJK$^+$16] Skinny-128/256 or Skinny-128/384. For long messages, Iwata et al. [IMPS17] reported performance figures of 0.87 and 0.99 cycles/byte on a single-core Intel Skylake i5-6600 with AES-NI for encrypting long messages in a fully parallel fashion with Deoxys-BC-256 and Deoxys-BC-384, respectively, and 4.12 and 4.8 cycles/byte for Skinny-128/256 or Skinny-128/384, respectively. They point out that those figures are expected to hold for modes where the tweak is used as counter, and experienced a performance-penalty factor of 1.4 for Deoxys-BC-256, a factor of 1.8 for Deoxys-BC-384, and no significant slow-down for the Skinny versions, when used with random tweaks. Table 2 compares performance estimates when processing long messages on Intel Skylake with AES-NI enabled, for the example scenarios of single-block and long (message-length) outputs. Clearly, the difference results in the additional TBC call per output block. We stress that the performance values for ZMAC and ZAE in [IMPS17] were also only estimations, and may differ in practical implementations.

8 Conclusion

This work proposed ZMAC$^+$, a VOLPRF based on the combination of ZHash with the finalization ZFin$^+$ that replaced of the sum of permutations from the original ZMAC. We introduced a per-query parameter that allows flexible tuning of the output length. The use of ZHash allows to inherit all its advantages: it is fully parallelizable, processes $n + t$-bit per TBC call in the hashing process, and uses only a single primitive under a single key. Moreover, ZMAC$^+$ could avoid the term $O(\sigma^2/2^{n+\min\{n,t\}})$ from the security bound of ZMAC while retaining a practical tweak space of only three different permutations that can be realized e. g., by reserving two tweak bits for the domain.

References

[BDP$^+$16] Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Farfalle: Parallel Permutation-based Cryptography. *IACR Cryptology ePrint Archive*, 2016:1188, 2016.

[Ber16] Daniel J. Bernstein. Some Challenges in Heavyweight Cipher Design. Technical report, January 11 2016. `https://cr.yp.to/talks/2016.01.15/slides-djb-20160115-a4.pdf`.

[BGM04] Mihir Bellare, Oded Goldreich, and Anton Mityagin. The Power of Verification Queries in Message Authentication and Authenticated Encryption. *IACR Cryptology ePrint Archive*, 2004:309, 2004.

[BI99] Mihir Bellare and Russell Impagliazzo. A tool for obtaining tighter security analyses of pseudorandom function based constructions, with applications to PRP to PRF conversion. *IACR Cryptology ePrint Archive*, 1999:24, 1999.

[BJK$^+$16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO II*, volume 9815 of *LNCS*, pages 123–153. Springer, 2016. Full version at `https://eprint.iacr.org/2016/660`.

[BKR94] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The Security of Cipher Block Chaining. In Yvo Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 341–358. Springer, 1994.

[BKR98] Mihir Bellare, Ted Krovetz, and Phillip Rogaway. Luby-Rackoff Backwards: Increasing Security by Making Block Ciphers Non-invertible. In Kaisa Nyberg, editor, *EUROCRYPT*, volume 1403 of *Lecture Notes in Computer Science*, pages 266–280. Springer, 1998.

[BKR00] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The Security of the Cipher Block Chaining Message Authentication Code. *J. Comput. Syst. Sci.*, 61(3):362–399, 2000.

[BR00] Mihir Bellare and Phillip Rogaway. Encode-Then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography. In Tatsuaki Okamoto, editor, *ASIACRYPT*, volume 1976 of *LNCS*, pages 317–330. Springer, 2000.

[BR02] John Black and Phillip Rogaway. A Block-Cipher Mode of Operation for Parallelizable Message Authentication. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *LNCS*, pages 384–397. Springer, 2002.

[CLS17] Benoít Cogliati, Jooyoung Lee, and Yannick Seurin. New Constructions of MACs from (Tweakable) Block Ciphers. In *IACR Transactions on Symmetric Cryptology*, volume 2017, pages 27–58, 2017.

[CS14] Shan Chen and John P. Steinberger. Tight Security Bounds for Key-Alternating Ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT*, volume 8441 of *LNCS*, pages 327–350. Springer, 2014. Full version at `https://eprint.iacr.org/2013/222`.

[CW79] Larry Carter and Mark N. Wegman. Universal Classes of Hash Functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.

[DHT17] Wei Dai, Viet Tung Hoang, and Stefano Tessaro. Information-Theoretic Indistinguishability via the Chi-Squared Method. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO Part III*, volume 10403 of *LNCS*, pages 497–523. Springer, 2017. Full version at `http://eprint.iacr.org/2017/537`, latest version 20170616:190106.

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust Authenticated-Encryption AEZ and the Problem That It Solves. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT (1)*, volume 9056 of *LNCS*, pages 15–44. Springer, 2015. Full version at `https://eprint.iacr.org/2014/793`.

[IK03a] Tetsu Iwata and Kaoru Kurosawa. OMAC: One-Key CBC MAC. In Thomas Johansson, editor, *FSE*, volume 2887 of *Lecture Notes in Computer Science*, pages 129–153. Springer, 2003.

[IK03b] Tetsu Iwata and Kaoru Kurosawa. Stronger Security Bounds for OMAC, TMAC, and XCBC. In Thomas Johansson and Subhamoy Maitra, editors, *INDOCRYPT*, volume 2904 of *Lecture Notes in Computer Science*, pages 402–415. Springer, 2003.

[IMPS17] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A Fast Tweakable Block Cipher Mode for Highly Secure Message Authentication. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO, Part III*, volume 10403 of *LNCS*, pages 34–65. Springer, 2017. Full version at `https://eprint.iacr.org/2017/535`.

[JNP16] Jérémy Jean, Ivica Nikolić, and Thomas Peyrin. Deoxys v1.4. `http://competitions.cr.yp.to/caesar-submissions.html`, 2016. Third-round submission to the CAESAR competition.

[KR11] Ted Krovetz and Phillip Rogaway. The Software Performance of Authenticated-Encryption Modes. In Antoine Joux, editor, *FSE*, volume 6733 of *LNCS*, pages 306–327. Springer, 2011.

[Kra94] Hugo Krawczyk. LFSR-based Hashing and Authentication. In Yvo Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 129–139. Springer, 1994.

[LN17] Eik List and Mridul Nandi. Revisiting Full-PRF-Secure PMAC and Using It for Beyond-Birthday Authenticated Encryption. In Helena Handschuh, editor, *CT-RSA*, LNCS, pages 258–274. Springer, 2017. Full version at `https://eprint.iacr.org/2016/1174`.

[LPTY16] Atul Luykx, Bart Preneel, Elmar Tischhauser, and Kan Yasuda. A MAC Mode for Lightweight Block Ciphers. In Thomas Peyrin, editor, *FSE*, volume 9783 of *Lecture Notes in Computer Science*, pages 43–59. Springer, 2016.

[LRW02] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable Block Ciphers. In Moti Yung, editor, *CRYPTO*, volume 2442 of *LNCS*, pages 31–46. Springer, 2002.

[Luc00] Stefan Lucks. The Sum of PRPs Is a Secure PRF. In Bart Preneel, editor, *EUROCRYPT*, volume 1807 of *LNCS*, pages 470–484. Springer, 2000.

[MI15] Kazuhiko Minematsu and Tetsu Iwata. Tweak-Length Extension for Tweakable Blockciphers. In Jens Groth, editor, *IMA Int. Conf.*, volume 9496 of *LNCS*, pages 77–93. Springer, 2015. Full version at `https://eprint.iacr.org/2015/888`.

[MI17] Kazuhiko Minematsu and Tetsu Iwata. Cryptanalysis of PMACx, PMAC2x, and SIVx. *IACR Transactions on Symmetric Cryptology*, 2017(2):162–176, 2017.

[MP15] Bart Mennink and Bart Preneel. On the XOR of Multiple Random Permutations. In Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, and Michalis Polychronakis, editors, *ACNS*, volume 9092 of *Lecture Notes in Computer Science*, pages 619–634. Springer, 2015.

[Nai15] Yusuke Naito. Full PRF-Secure Message Authentication Code Based on Tweakable Block Cipher. In Man Ho Au and Atsuko Miyaji, editors, *ProvSec*, volume 9451 of *LNCS*, pages 167–182. Springer, 2015.

[Nai16] Yusuke Naito. Sandwich Construction for Keyed Sponges: Independence Between Capacity and Online Queries. In Sara Foresti and Giuseppe Persiano, editors, *CANS*, volume 10052 of *LNCS*, pages 245–261, 2016.

[NIS15] NIST. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. *Federal Information Processing Standards (FIPS) Publication*, 202, 2015. `http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf`.

[Pat08a] Jacques Patarin. A Proof of Security in O(2n) for the Xor of Two Random Permutations. In Reihaneh Safavi-Naini, editor, *ICITS*, volume 5155 of *LNCS*, pages 232–248. Springer, 2008. Full version at `https://eprint.iacr.org/2008/010`.

[Pat08b] Jacques Patarin. The "Coefficients H" Technique. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *SAC*, volume 5381 of *LNCS*, pages 328–345. Springer, 2008.

[Pat13] Jacques Patarin. Security in $O(2^n)$ for the Xor of Two Random Permutations: Proof with the standard H technique. *IACR Cryptology ePrint Archive*, 2013:368, 2013.

[Rog95] Phillip Rogaway. Bucket Hashing and its Application to Fast Message Authentication. In Don Coppersmith, editor, *CRYPTO*, volume 963 of *Lecture Notes in Computer Science*, pages 29–42. Springer, 1995.

[SO98] Richard Schroeppel and Hilarie Orman. The Hasty Pudding Cipher. *AES candidate submitted to NIST*, 1998.

[Yas11] Kan Yasuda. A New Variant of PMAC: Beyond the Birthday Bound. In Phillip Rogaway, editor, *CRYPTO*, volume 6841 of *LNCS*, pages 596–609. Springer, 2011.