Revisiting ZMAC+: Birthday-Bound Attacks and a Beyond-Birthday-Bound Repair

Yan Jia[1,2][0009−0005−4558−9394], Peng Wang[3], Lei Hu[1,2], Shuping Mao[4], Tingting Guo[5], and Jiadong Han[3]

[1] State Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{jiayan2022,hulei}@iie.ac.cn, jiayan22@mails.ucas.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] School of Cryptology, University of Chinese Academy of Sciences, Beijing, China
p-wang@mails.ucas.ac.cn
[4] Beijing Electronic Science and Technology Institute, Beijing, China
maoshuping19@mails.ucas.ac.cn
[5] Ningbo Key Laboratory of Information Technology Application Innovation and Security, School of Cyber Science and Engineering, Ningbo University of Technology
guotingting@nbut.edu.cn

Abstract. ZMAC+ is a tweakable-block-cipher-based MAC proposed by List and Nandi in ToSC 2017 with claimed beyond-birthday-bound security. We show that a tag-swap mechanism previously observed for a hypothetical two-output extension of DoveMAC also applies to the published ZMAC+ construction. For two-block tags, we give a distinguisher with constant advantage and a fresh-message forgery with constant success probability using $O(2^{n/2})$ queries in the ideal-TBC model, where n is the block length. An internal collision between two message families exchanges their tag blocks; a common extension preserves this relation and leads to a forgery with one additional authentication query. We identify a counting error in the published Lemma 4 and prove a corrected aggregate differential bound. The resulting PRF and MAC upper bounds match the birthday query threshold for bounded message lengths and two-block tags, under explicit field-order conditions. The distinguisher extends to every legal fixed output length $d \geq 2$, and the forgery to every legal even d. We also retain ZHash and separate the output domains. Under stated field-order, length, and domain-capacity conditions, this modification admits an adaptive PRF bound of order $q^2/2^{n+\min(n,t)}$ for bounded lengths. Finally, a one-query counterexample shows that the ZAE security theorem must retain the per-message length restriction required by its counter-in-tweak encryption component.

Keywords: Message authentication codes · ZMAC+ · birthday bound · tweakable block ciphers · differential probabilities · cryptanalysis

1 Introduction

Message Authentication Codes. A message authentication code (MAC) provides authentication of messages for two parties sharing a secret key. We consider stateless, deterministic MACs. A standard strategy to proving their security is to show that the tagging function is a pseudorandom function (PRF), then reduce forgery to distinguishing the PRF from a random function. For many classical constructions based on block ciphers, the security bound contains a term of order $q^2/2^n$, where n is the block length and q is the number of authentication queries [Rog04,LPTY16]. This term becomes constant when $q \approx 2^{n/2}$, the birthday bound. Beyond-birthday-bound (BBB) MACs aim to provide security beyond this bound. Examples include SUM-ECBC and PMAC+ [Yas10,Yas11].

Tweakable Block Ciphers and ZMAC+. A tweakable block cipher (TBC) is a family of permutations indexed by a secret key and a public tweak. By using both the n-bit data input and the t-bit tweak input, a MAC can process $n + t$ message bits per primitive call. ZMAC, proposed by Iwata et al. [IMPS17a], uses this approach in its parallel hash function ZHash. It then computes a tag using a sum of permutations for generating tags. List and Nandi [LN17] proposed ZMAC+, which retains ZHash but replaces the finalization with a simpler hash-then-TBC construction. ZMAC+ also allows the adversary to choose the output length for each query. For a bounded number of output blocks per query, the claimed security bound has the form

$$O\left(\frac{q}{2^n} + \frac{q(q+\sigma)}{2^{n+\min(n,t)}}\right), \tag{1}$$

where σ is the total number of encoded message blocks. If $t = n$ and each message has a bounded number of blocks, this expression remains negligible at $q \approx 2^{n/2}$. It therefore claims security beyond the birthday threshold.

Our Contributions. The tag-swap mechanism in the DoveMAC analysis [GLN19, Appendix A.4] also applies to ZMAC+. We establish its consequences for the published construction and replace the invalid security argument.

(1) We construct valid encoded message families that yield a birthday-bound distinguisher and a fresh-message forgery. We give explicit success bounds, including the probability of accidental tag matches. The distinguisher applies to every legal fixed $d \geq 2$. A common extension and one additional authentication query give a complete-tag forgery for every legal even d.

(2) We identify a counting error in the published Lemma 4: fixing a block's right component leaves its left component unrestricted. Consequently, $\Theta(q^2)$ pairs can each contribute probability 2^{-n}. Our replacement lemma gives VOLPRF and MAC bounds of order $O(q^2/2^n)$ for bounded lengths. The upper and lower bounds establish the birthday threshold for fixed two-block tags (under stated field-order conditions).

(3) We retain ZHash and assign a separate finalization domain to each output position. We prove the required hash bounds and an adaptive VOLPRF theorem, obtaining an information-theoretic term of order $O(q^2/2^{n+\min(n,t)})$ for bounded lengths under explicit conditions.

(4) We identify an omitted message-length restriction in the ZAE theorem. A payload of 2^t+1 full blocks repeats an encryption counter, giving one-query distinguishing advantage $1-2^{-n}$. This issue does not affect the original ZMAC MAC and PRF theorems.

The main attacks fix $d=2$ throughout. A distinguisher alone does not establish a forgery; we prove both results.

Related Work. The closest antecedent is Grochow, List, and Nandi's DoveMAC [GLN19]. Their Appendix A.4 gives a birthday distinguisher for a hypothetical two-output extension: two message families induce a data collision and a tweak difference of one, which exchanges the output blocks of ZFin^+. The specified DoveMAC has one output block, so that observation does not refute its stated security. Their Appendix A.2 also uses common extension to turn a collision into a fresh-message forgery against a different weakened variant. We establish these mechanisms for ZMAC+'s masked ZHash and output-length encoding, bound accidental matches, identify the error in its Lemma 4, and determine the resulting security threshold. In particular, the DoveMAC appendix does not itself provide the counterexample to the published ZMAC+ bound proved here.

Leurent, Nandi, and Sibleyras [LNS18] study generic attacks on double-block hash-then-sum MACs, including SUM-ECBC and PMAC+. They explicitly note that their method does not seem to apply to ZMAC or ZMAC+. Naito [Nai18] studies the efficiency of ZMAC-type modes. Jha and Nandi [JN22, Section 5.2, Lemma 6] revisit the single-output hash-then-TBC theorem. That result does not cover the cross-position relations of an indexed multi-output finalization; our counterexample does not invalidate their single-output lemma. Our repair extends that proof approach to independent output domains. Bonnetain et al. [BLNS21, Section 4.3] attack the original ZMAC using quantum superposition queries. Their Q2-model result concerns a different query model and does not refute the classical security theorem; our ZMAC+ attacks use classical queries.

2 Preliminaries and the ZMAC+ construction

2.1 Notation and security definitions

We write $\{0,1\}^r$ for r-bit strings, $\|$ for concatenation, and $\oplus$ for bitwise XOR. The bit length of M is denoted b y$|M|$. For $0 \le a < 2^r$, let $\langle a \rangle_r$ denote the r-bit, big-endian representation of a. We write $x \xleftarrow{\$} \mathcal{X}$ for uniform sampling from a finite set $\mathcal{X}$. Deterministic assignment is written $x \leftarrow f(y)$. All quantities associated with authentication query i carry the superscript (i); subscripts indicate positions within that query.

Let $N = 2^n$ and $\tau = \min(n, t)$. We can interpret $\{0,1\}^n$ as a element in $\mathbb{F}_{2^n}$, using a fixed public field representation. As in the original specification, 2 denotes the field element represented by the polynomial x. Multiplication by 2 and its powers is evaluated in $\mathbb{F}_{2^n}$; powers of 2 in cardinalities and security bounds denote ordinary integers. The upper bounds below require the field powers $2^0, 2^1, \ldots, 2^{m_{\max}}$ to be distinct, or equivalently that the field element 2 have multiplicative order greater than $m_{\max}$ [GJMN16]. A primitive reduction polynomial ensures this condition whenever $m_{\max} < N - 1$. For another field representation, the condition must be checked for the chosen message-length bound. Our attacks do not require distinct field powers. Define the linear map

$$\mathsf{p}_t(z) = \begin{cases} \mathsf{msb}_t(z), & t \le n, \\ z \| 0^{t-n}, & t > n. \end{cases} \tag{2}$$

The map has rank τ. We use it to state all operations on unequal-length strings explicitly and avoid the operand-order ambiguity in the original partial-XOR notation.

Definition 1 (TBC and TPRP security). *A TBC $\widetilde{E} : \mathcal{K} \times (\{0,1,2\} \times \{0,1\}^t) \times \{0,1\}^n \to \{0,1\}^n$ defines a permutation of $\{0,1\}^n$ for each fixed key and augmented tweak. We write $\widetilde{E}_K^{(a,T)}(z)$ for its evaluation at key K, augmented tweak (a,T), and data input z. An ideal TBC $\widetilde{P}$ is a family of independent uniform random permutations $\widetilde{P}^{(a,T)}$, one for each augmented tweak (a,T). For an adversary $\mathcal{D}$ with forward oracle access, define*

$$\mathsf{Adv}^{\mathrm{tprp}}_{\widetilde{E}}(\mathcal{D}) = \left| \Pr[\mathcal{D}^{\widetilde{E}_K} = 1] - \Pr[\mathcal{D}^{\widetilde{P}} = 1] \right|, \qquad K \xleftarrow{\$} \mathcal{K}.$$

We use forward TPRP security throughout. Domain labels 0, 1, and 2 are reserved for hashing, finalization, and mask generation, respectively. They select independent permutations in the ideal-TBC model.

Definition 2 (PRF and variable-output-length PRF). *For a keyed function $F_K : \mathcal{X} \to \{0,1\}^\ell$, its PRF advantage is*

$$\mathsf{Adv}^{\mathrm{prf}}_{F}(\mathcal{D}) = \left| \Pr[\mathcal{D}^{F_K} = 1] - \Pr[\mathcal{D}^{R} = 1] \right|,$$

where K is sampled uniformly from the key space and R is a uniform random function on the same domain and range. For $F_K(M,d) \in (\{0,1\}^n)^d$, define $\mathsf{Adv}^{\mathrm{volprf}}_{F}$ analogously, using an ideal oracle $\mathcal{R}$ that independently samples a uniform nd-bit string on every new pair (M,d) and reuses it on repeated pairs.

In particular, $\mathcal{R}(M,d)$ and $\mathcal{R}(M,d')$ are independent for $d \neq d'$. This is the VOLPRF notion of [LN17].

Definition 3 (MAC unforgeability). *A tagging oracle returns $F_K(M,d)$. After at most q authentication queries, an adversary outputs (M^*, d^*, u^*) and wins if $u^* = F_K(M^*, d^*)$ and (M^*, d^*) was never authenticated. We denote its success probability by $\mathsf{Adv}^{\mathrm{mac}}_{F}$. For a fixed output length, d is a public scheme parameter and freshness reduces to freshness of M^*.*

We consider one final forgery attempt and no verification oracle. In the ideal VOLPRF experiment, a fresh target with output length d^* has a uniform tag, so its guessing probability is 2^{-nd^*}. Our attack fixes $d = 2$ and forges a new message.

Definition 4 (Hash differences). *For a keyed hash H with output $(h, g) \in \{0,1\}^n \times \{0,1\}^t$, define*

$$\mathsf{DP}_H(\mu, \nu; (\eta, \delta)) = \Pr[H(\mu) \oplus H(\nu) = (\eta, \delta)], \quad \mu \neq \nu.$$

The probability is taken over the hash key or, in the ideal model, over the hidden primitives that define the hash. The following properties must hold for every pair of distinct inputs. The hash is ε-almost universal (AU) if $\mathsf{DP}_H(\mu, \nu; (0^n, 0^t)) \leq \varepsilon$; it is ε-almost XOR universal (AXU) if the same bound holds for every output difference. It is (n, t, ε)-partial-AXU if it holds for every $(\eta, 0^t)$, and (n, t, ε)-truncated-AXU if

$$\sum_{\eta \in \{0,1\}^n} \mathsf{DP}_H(\mu, \nu; (\eta, \delta)) \leq \varepsilon \quad \textit{for every } \delta \in \{0,1\}^t.$$

The last property states that the g component is AXU. These definitions follow the hash properties used in [LN17]. Our upper bound will use the collision probability of the h component directly.

2.2 Encoding and specification corrections

The parameter d specifies the number of n-bit blocks in the tag. It must fit in the n-bit length field appended to the message, and the d finalization positions must have distinct t-bit indices. We therefore use the admissible output-length domain

$$\mathcal{D} = \{1, \ldots, \min(2^n - 1, 2^t)\}. \tag{3}$$

The bound $d \leq 2^n - 1$ ensures that d can be represented by $\langle d \rangle_n$. The bound $d \leq 2^t$ ensures that each index $\langle j-1 \rangle_t$, for $1 \leq j \leq d$, is well defined. The original output-length domain includes $d = 2^n$ when $t \geq n$, although an n-bit field cannot encode that integer. Equation (3) removes this invalid endpoint when necessary.

To encode a message $M \in \{0,1\}^*$ and a requested length $d \in \mathcal{D}$, first append a delimiter bit 1, then p zero bits, and finally the n-bit representation of d. Choose p as the smallest nonnegative number that makes the resulting bit length divisible by $n + t$. Denote the encoded string by μ and its number of complete blocks by m:

$$\begin{aligned} p &= (-|M| - n - 1) \bmod (n + t), \qquad 0 \leq p < n + t, \\ \mu &= \mathsf{enc}(M, d) = M \| 1 \| 0^p \| \langle d \rangle_n, \\ m &= \frac{|\mu|}{n + t} = \frac{|M| + 1 + p + n}{n + t}. \end{aligned} \tag{4}$$

The least nonnegative residue uniquely determines p, which may be zero. The encoded length is a *multiple* of $n + t$, not necessarily one block. Parse μ into $m \geq 1$ complete blocks $\mu_s = a_s \| b_s$, where $a_s \in \{0,1\}^n$ enters the data path and $b_s \in \{0,1\}^t$ enters the tweak path of the hash.

The encoding is injective: the last n bits determine d, and removing that field, the trailing zeros, and the delimiter recovers M. We retain the original convention of encoding d itself. The correction to its admissible range does not affect our attack, which uses $d = 2$ for all $n \geq 2$ and $t \geq 1$.

We correct two further typographical errors in [LN17]: the encoding in Definition 9 must have both parameters n, t, and the second message in Theorem 2 must be $\mathsf{enc}(M', d')$, where the printed statement repeats (M, d). Finally, both masks generated by the TBC have n bits. We therefore apply (2) whenever a mask or cipher output enters the t-bit path. These conventions specify the intended construction without changing its cryptographic structure.

2.3 The algorithm

Mask Generation. For each key, precompute the two masks

$$L = \widetilde{E}_K^{(2,0^t)}(\langle 1 \rangle_n), \qquad R = \widetilde{E}_K^{(2,\langle 1 \rangle_t)}(\langle 1 \rangle_n). \tag{5}$$

ZHash. For an encoded message μ, initialize $(h_0, g_0) = (0^n, 0^t)$ and compute

$$\begin{aligned} z_s &= a_s \oplus 2^{s-1}L, & w_s &= b_s \oplus \mathsf{p}_t(2^{s-1}R), \\ v_s &= \widetilde{E}_K^{(0,w_s)}(z_s), & g_s &= g_{s-1} \oplus b_s \oplus \mathsf{p}_t(v_s), \\ h_s &= 2(h_{s-1} \oplus v_s), & & 1 \leq s \leq m. \end{aligned} \tag{6}$$

The output is $(h, g) = (h_m, g_m)$, or equivalently

$$h = \bigoplus_{s=1}^{m} 2^{m+1-s} v_s, \qquad g = \bigoplus_{s=1}^{m} \big(b_s \oplus \mathsf{p}_t(v_s)\big). \tag{7}$$

ZFin+. The finalization uses h as the data input in every call and XORs the output index into g to obtain the tweak. It returns $u = (u_1, \ldots, u_d)$, where

$$u_j = \widetilde{E}_K^{(1,g \oplus \langle j-1 \rangle_t)}(h), \qquad 1 \leq j \leq d. \tag{8}$$

We write $\mathsf{ZMAC}_K^+(M, d) = \mathsf{ZFin}_K^+(\mathsf{ZHash}_K(\mathsf{enc}(M, d)), d)$. Algorithm 1 gives the complete procedure. Figure 1 shows the attack's two-block encoding. The hash calls depend only on the blocks, positions, and masks, so they are parallelizable. A common block at the same position produces the same cipher output in every query.

Algorithm 1 ZMAC+, with explicit lengths and partial XOR

Require: $M \in \{0,1\}^*$, $d \in \mathcal{D}$, secret key K; cached masks L, R
1: $\mu \leftarrow \mathsf{enc}(M, d)$; parse $\mu = (a_1 \| b_1) \| \cdots \| (a_m \| b_m)$
2: $(h, g) \leftarrow (0^n, 0^t)$
3: **for** $s = 1, \ldots, m$ **do**
4: $\quad z \leftarrow a_s \oplus 2^{s-1}L$; $w \leftarrow b_s \oplus \mathsf{p}_t(2^{s-1}R)$
5: $\quad v \leftarrow \widetilde{E}_K^{(0,w)}(z)$
6: $\quad g \leftarrow g \oplus b_s \oplus \mathsf{p}_t(v)$; $h \leftarrow 2(h \oplus v)$
7: **end for**
8: **for** $j = 1, \ldots, d$ **do**
9: $\quad u_j \leftarrow \widetilde{E}_K^{(1, g \oplus \langle j-1 \rangle_t)}(h)$
10: **end for**
11: **return** $(u_1, \ldots, u_d)$

3 Birthday-bound attacks on ZMAC+

Throughout this section, the underlying primitive is an ideal TBC $\widetilde{P}$. The masks L, R are independent uniform n-bit strings, and the hash and finalization domains are independent of each other and of these masks. We assume $n \geq 2$, $t \geq 1$, and first fix $d = 2$. This attack satisfies the original encoding rules; at $t = n$ it also avoids any ambiguity about unequal-length XOR. Its tag-swap mechanism is the one observed for the DoveMAC extension in [GLN19, Appendix A.4].

3.1 Two query families and tag exchange

The attack makes the data components of two hashes equal while forcing their tweak components to differ by $\langle 1 \rangle_t$. For $d = 2$, this difference exchanges the two finalization tweaks. The adversary can detect the resulting relation by comparing complete tags.

Choose $1 \leq k \leq N$ distinct values $a^{(1)}, \ldots, a^{(k)} \in \{0,1\}^n$ and distinct values $a^{(k+1)}, \ldots, a^{(2k)} \in \{0,1\}^n$. The two lists may overlap. Define

$$b^{(i)} = \begin{cases} 0^t, & 1 \leq i \leq k, \\ \langle 1 \rangle_t, & k < i \leq 2k, \end{cases} \qquad M^{(i)} = a^{(i)} \| b^{(i)} \| 0^{t-1}, \quad d^{(i)} = 2. \tag{9}$$

Each message has $n + 2t - 1$ bits, so $p = 0$ in (4). The delimiter and output-length field give exactly two encoded blocks:

$$\mu^{(i)} = (a^{(i)} \| b^{(i)}) \| Q, \qquad Q = 0^{t-1} \| 1 \| \langle 2 \rangle_n \in \{0,1\}^{n+t}. \tag{10}$$

Write $Q = c \| e$, with $|c| = n$ and $|e| = t$. The first cipher output depends on the query, while the second is a common value $W = v_2^{(i)}$ for all i:

$$\begin{aligned} v_1^{(i)} &= \widetilde{P}^{(0, b^{(i)} \oplus \mathsf{p}_t(R))}(a^{(i)} \oplus L), \\ W &= \widetilde{P}^{(0, e \oplus \mathsf{p}_t(2R))}(c \oplus 2L). \end{aligned} \tag{11}$$

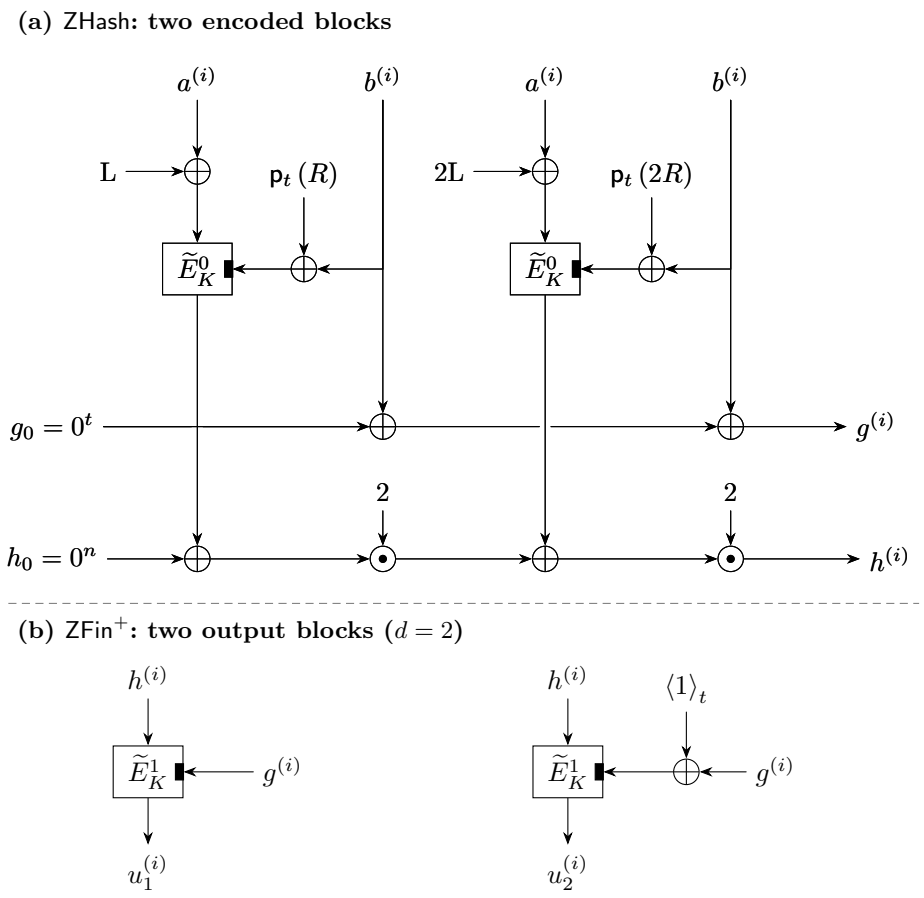

Fig. 1. ZMAC+ on the attack encoding $\mu^{(i)} = (a^{(i)} \| b^{(i)}) \| (c \| e)$. **Top:** (a) uses the common block $Q = c \| e = 0^{t-1} \| 1 \| \langle 2 \rangle_n$; its output W is shared by all queries. The projection branches enter $g^{(i)}$, while field doubling enters $h^{(i)}$. **Bottom:** (b) reuses $h^{(i)}$ with tweaks $g^{(i)}$ and $g^{(i)} \oplus \langle 1 \rangle_t$. In the artwork, the superscript denotes the domain and the side arrow supplies the tweak; solid dots denote branches. Masks are defined in (5).

By (7), the hash of query i is

$$h^{(i)} = 2^2 v_1^{(i)} \oplus 2W, \qquad g^{(i)} = b^{(i)} \oplus e \oplus \mathsf{p}_t(v_1^{(i)} \oplus W). \tag{12}$$

Conditional on L, R, the sets $\mathcal{V}_0 = \{v_1^{(i)} : 1 \le i \le k\}$ and $\mathcal{V}_1 = \{v_1^{(j)} : k < j \le 2k\}$ are independent, uniform k-subsets of $\{0,1\}^n$. Within each family, distinct values of $a^{(i)}$ give distinct inputs to one random permutation. The two families use different tweaks, so their permutations are independent. The common value W may depend on these sets, but it cancels in the hash difference.

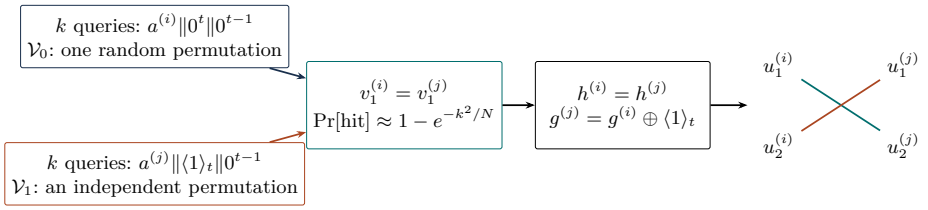

Fig. 2. Tag exchange from a cross-family collision. Equality of the n-bit first-block outputs gives a fixed t-bit hash difference, which exchanges the two finalization tweaks and reverses the $2n$-bit tag.

Lemma 1 (Forced exchange). *For $i \le k < j$, if $v_1^{(i)} = v_1^{(j)}$, then*

$$h^{(i)} = h^{(j)}, \quad g^{(i)} \oplus g^{(j)} = \langle 1\rangle_t, \quad (u_1^{(i)}, u_2^{(i)}) = (u_2^{(j)}, u_1^{(j)}). \tag{13}$$

Proof. The first two equalities follow directly from (12). In (8), the first primitive call for query i is therefore identical to the second call for query j, and conversely. Thus the tags are reversals of each other for every choice of the finalization permutations.

3.2 Distinguishing probability and complexity

Let $p_k = \Pr[\mathcal{V}_0 \cap \mathcal{V}_1 \neq \varnothing]$. Its exact value is

$$p_k = 1 - \frac{\binom{N-k}{k}}{\binom{N}{k}} = 1 - \prod_{r=0}^{k-1} \left(1 - \frac{k}{N-r}\right), \quad 2k \le N; \qquad p_k = 1 \quad (2k > N). \tag{14}$$

To see this, fix $\mathcal{V}_0$. Among the $\binom{N}{k}$ possible sets $\mathcal{V}_1$, exactly $\binom{N-k}{k}$ are disjoint from it. If $2k > N$, an intersection is unavoidable.

Theorem 1 (Birthday distinguisher). *There is a nonadaptive adversary $\mathcal{A}_{\text{dist}}$ making $2k$ queries, each with two encoded blocks and two output blocks, such that*

$$\mathsf{Adv}^{\text{volprf}}_{\mathsf{ZMAC}^+[\widetilde{P}]}(\mathcal{A}_{\text{dist}}) \ge p_k - \frac{k^2}{N^2}. \tag{15}$$

It also distinguishes the fixed-output-length tagging function for $d = 2$.

Proof. The adversary queries all messages in (9) and returns 1 exactly when a pair from different families has reversed tags. In the real experiment, Lemma 1 guarantees acceptance whenever $\mathcal{V}_0 \cap \mathcal{V}_1 \neq \varnothing$. The acceptance probability is therefore at least p_k.

In the ideal experiment, the $2k$ distinct messages receive independent uniform tags in $(\{0,1\}^n)^2$. A fixed pair has reversed tags with probability N^{-2}. A union bound over the k^2 cross-family pairs bounds the ideal acceptance probability by k^2/N^2. Taking the difference proves the claim.

For $k = c\sqrt{N} + O(1)$ and fixed $c > 0$,

$$p_k = 1 - e^{-c^2} + O(N^{-1/2}), \qquad \mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{ZMAC}^+[\widetilde{P}]}(\mathcal{A}_{\mathrm{dist}}) \geq 1 - e^{-c^2} - O(N^{-1/2}). \quad (16)$$

The asymptotic expression follows by taking logarithms of the product in (14). To implement the attack, the adversary stores the tags of the first family in a dictionary and looks up the reversal of each second-family tag. It stores $O(k)$ tags and performs an expected $O(k)$ dictionary operations. A sorting-based implementation uses $O(k \log k)$ comparisons. Each stored tag has $2n$ bits.

3.3 From exchange detection to a fresh-message forgery

A detected tag reversal suggests that the first cipher outputs collided. We use a common extension to transfer this relation to fresh messages, then bound the probability of selecting an accidental reversal. For each message in (9), define

$$M^{(i),+} = a^{(i)} \| b^{(i)} \| 0^{n+2t-1}, \qquad \mathsf{enc}(M^{(i),+}, 2) = (a^{(i)} \| b^{(i)}) \| 0^{n+t} \| Q. \quad (17)$$

This is the original raw message followed by $n + t$ zero bits, encoded again with $d = 2$. The resulting encoding has three blocks. Its first block is unchanged, and its two remaining blocks are common to all queries. If two first-block cipher outputs collide, the extended hashes therefore have equal h components and g difference $\langle 1 \rangle_t$. Their tags are again reversals of each other. This argument extends the raw message and then reapplies the encoding, including the delimiter and output-length field.

Lemma 2 (Accidental exchange). *For any fixed $i \leq k < j$ in the initial query set,*

$$\Pr[v_1^{(i)} \neq v_1^{(j)} \text{ and } u^{(i)} = \mathsf{rev}(u^{(j)})] \leq \frac{1}{N(N-1)}, \qquad \mathsf{rev}(x, y) = (y, x).$$

Proof. Condition on the masks and the entire hash-domain permutation family. If $v_1^{(i)} \neq v_1^{(j)}$, then $h^{(i)} \neq h^{(j)}$ by (12). Put $\delta = g^{(i)} \oplus g^{(j)}$. If $\delta = \langle 1 \rangle_t$, a reversed equality would require one permutation to map two distinct data inputs to the same output; its probability is zero. If $\delta = 0^t$, the two queries use the same two independent permutations. Evaluate each permutation at the two distinct data inputs. Each resulting ordered pair is uniform among the $N(N-1)$ pairs of distinct outputs, and the two pairs are independent. A tag reversal requires one pair to be the reversal of the other, which has probability $1/(N(N-1))$. For $\delta \notin \{0^t, \langle 1 \rangle_t\}$, all four tweaks are distinct, giving probability N^{-2}. In all cases the conditional probability is at most $1/(N(N-1))$. Averaging over the fixed masks and hash permutations proves the bound.

Theorem 2 (Birthday MAC forgery). *There is a forger $\mathcal{A}_{\mathrm{forge}}$ for fixed $d = 2$ using at most $2k + 1$ authentication queries, each with at most three encoded blocks, such that*

$$\mathsf{Adv}^{\mathrm{mac}}_{\mathsf{ZMAC}^+[\widetilde{P}]}(\mathcal{A}_{\mathrm{forge}}) \geq p_k - \frac{k^2}{N(N-1)}. \quad (18)$$

Algorithm 2 Birthday forgery against fixed two-block ZMAC+ tags

Query all $M^{(1)}, \ldots, M^{(2k)}$ from (9), obtaining $u^{(1)}, \ldots, u^{(2k)}$
Store $(u^{(i)}, i)$ for $1 \le i \le k$ in a dictionary
Scan $k < j \le 2k$ for a lookup of $\mathsf{rev}(u^{(j)})$; retain the first matching (i, j)
if a pair (i, j) was found **then**
    $u^{(2k+1)} \leftarrow \mathsf{Tag}(M^{(i),+}, 2)$
    return $(M^{(j),+}, 2, \mathsf{rev}(u^{(2k+1)}))$
end if
return failure

In particular, its success probability is at least $1 - e^{-1} - o(1)$ *for* $k = \lfloor \sqrt{N} \rfloor$.

Proof. Search for reversed tags among the $2k$ initial queries. If a pair (i, j) is found, authenticate $M^{(i),+}$, then output the reversal of its tag as a forgery for $M^{(j),+}$ with output length 2. The target is fresh: all initial messages are shorter, and the only longer authenticated message is in the other family.

Let E be the event $\mathcal{V}_0 \cap \mathcal{V}_1 \neq \varnothing$, and let F be the event that some initial cross-family pair has reversed tags without a first-block collision. On $E \setminus F$, the search succeeds and every matching pair has equal first-block cipher outputs. The common extension then yields a valid forgery with certainty. This implication is deterministic, so it remains valid even when the longer queries reuse previous primitive inputs. Lemma 2 and a union bound give $\Pr[F] \le k^2/(N(N-1))$. Since $\Pr[E] = p_k$, $\Pr[\text{win}] \ge \Pr[E] - \Pr[F]$ proves the theorem.

The attack requests complete two-block tags and forges a previously unauthenticated message. Algorithm 2 gives the procedure.

Only the additional authentication query depends on the observed tags. The bound holds for any rule used to select a matching pair.

3.4 Other fixed output lengths

Proposition 1. *For every legal fixed* $d \ge 2$*, the distinguishing bound in* (15) *holds with* $2k$ *queries. For every legal even* d*, the complete-tag forgery bound in* (18) *holds with at most* $2k + 1$ *authentication queries and at most three encoded blocks per message.*

Proof. Replace the common block by $Q_d = 0^{t-1} \| 1 \| \langle d \rangle_n$. The first-block distributions are unchanged, and the common second-block output still cancels. A cross-family collision therefore gives $\Delta h = 0^n$ and $\Delta g = \langle 1 \rangle_t$. Comparing only the first two tag blocks proves the same distinguishing bound.

For even d, XOR by one permutes the index set $\{0, \ldots, d-1\}$. Thus a genuine collision exchanges every adjacent pair of tag blocks: $(u_1, u_2, \ldots, u_{d-1}, u_d)$ becomes $(u_2, u_1, \ldots, u_d, u_{d-1})$. Detect a candidate using its first two blocks, authenticate one common extension, and apply this permutation to its entire tag to forge the other extension. Lemma 2 still bounds false candidates, and the freshness argument of Theorem 2 is unchanged.

Table 1. Exact-formula evaluation at $n = t = 16$ and $k = 256$. The TPRP term is zero in the ideal-TBC model. The forgery uses one additional query.

Quantity	Value
Internal intersection probability p_k	0.633558523466
Our VOLPRF advantage lower bound	0.633543264677
Our MAC forgery success lower bound	0.633543264444
Published Corollary 3: VOLPRF upper bound	0.033018626273
Published Lemma 4: claimed aggregate upper bound	0.032042026520
Aggregate contribution of our cross-family pairs	1

For odd d, the last index is mapped outside the output range, so this argument does not give a complete-tag forgery. For growing d, processing the returned tags costs $O(kd)$; matching birthday query thresholds are claimed here for bounded output lengths.

3.5 Comparison with the published bounds

Set $n = t = 16$, $k = 256$, and $d^{(i)} = 2$. Then $q = 512$, $m_{\max} = 2$, $\sigma = 1024$ encoded blocks, and $D = 1024$ output blocks. These parameters satisfy the original length restrictions. Table 1 compares our exact probability formulas with the published upper bounds. Our distinguishing lower bound is more than 19 times the published upper bound. Thus the contradiction holds for concrete parameters and does not rely on an asymptotic approximation.

The upper bound claimed in the published Lemma 4 is

$$\frac{2D}{N} + \frac{2(q-1)\sigma + q^2 + 4(q-1)D}{2^{n+\tau}}. \tag{19}$$

Corollary 3 adds the term

$$\frac{4q^2}{N}\left(\frac{4m_{\max}+2}{2^{n+\tau}} + \frac{4}{2^{\tau}}\right)$$

to (19), with $D = 2q$. Our distinguishing bound contradicts Corollary 3. The aggregate contribution of the cross-family pairs also contradicts Lemma 4, as explained next.

3.6 The counting error in Lemma 4

Consider two encodings of the same length that differ only at position s. This is Case 1 in the proof of Theorem 2 of [LN17], which is used again in its Lemma 4. Write Δ for the XOR difference of the corresponding values. Equation (7) gives

$$\Delta h = 2^{m+1-s}\Delta v_s, \qquad \Delta g = \Delta b_s \oplus \mathsf{p}_t(\Delta v_s). \tag{20}$$

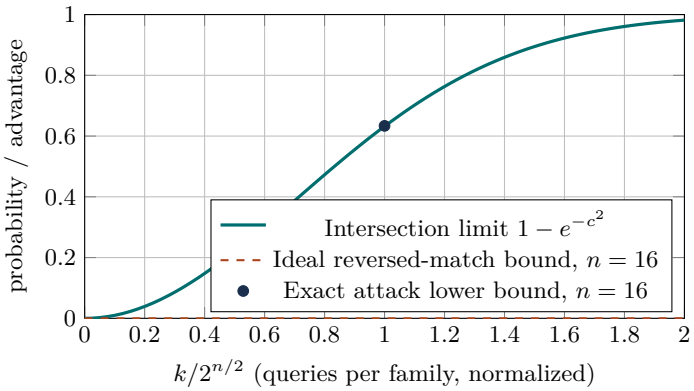

Fig. 3. Distinguishing advantage at the birthday scale. The intersection probability tends to $1 - e^{-c^2}$, whereas the ideal reversed-match bound is $O(2^{-n})$. The marked point uses the exact formula at $n = 16$.

Since multiplication by 2^{m+1-s} is invertible, $\Delta h = 0^n$ implies $\Delta v_s = 0^n$, and hence $\Delta g = \Delta b_s$. If $b_s = b'_s$ but $a_s \neq a'_s$, the two primitive calls use the same tweak and different data inputs, so their outputs cannot agree. If $b_s \neq b'_s$, their tweaks differ and the outputs agree with probability exactly $1/N$. The correct per-pair differential probability is therefore

$$\mathsf{DP}_{\mathsf{ZHash}[\widetilde{P}]}(\mu, \nu; (0^n, \delta)) = \begin{cases} N^{-1}, & b_s \neq b'_s \text{ and } \delta = b_s \oplus b'_s, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

The contributions of all other positions cancel. This cancellation holds even if a common block shares a primitive input with the differing block.

The counting error occurs when the published proof sums these probabilities over message pairs. On printed page 319, it argues that fixing one message, the position s, and the difference δ determines at most one other message. These choices determine only $b'_s = b_s \oplus \delta$; the left component $a'_s \in \{0,1\}^n$ remains unrestricted. Our two families give k^2 pairs with the same allowed difference $\langle 1 \rangle_t$, each contributing $1/N$. Their total contribution is k^2/N, which equals 1 for the parameters in Table 1. Reordering the queries by output length, as in the published proof, leaves this contribution unchanged.

For general output lengths, define the set of possible XOR differences between finalization indices by

$$\mathcal{K}_{ij} = \{\langle a \rangle_t \oplus \langle b \rangle_t : 0 \leq a < d^{(i)},\ 0 \leq b < d^{(j)}\} \subseteq \{0,1\}^t. \tag{22}$$

A primitive input is shared by finalization calls from distinct queries i, j exactly when $\Delta h = 0^n$ and $\Delta g \in \mathcal{K}_{ij}$. The set counts each possible difference once, and every element is a well-defined t-bit string. For $d^{(i)} = d^{(j)} = 2$, $\mathcal{K}_{ij} = \{0^t, \langle 1 \rangle_t\}$. The published interval also contains $\langle 2 \rangle_t$ when it is representable; that extra nonnegative term cannot reduce the aggregate sum.

3.7 A corrected aggregate differential bound

We now give a conservative replacement for Lemma 4 of [LN17]. We first bound collisions in the h component for any fixed pair of encoded messages. We then sum over all pairs. This proof is independent of the remaining case estimates in the original differential analysis.

Lemma 3 (Data-component collision bound). *Let μ, ν be distinct encoded strings with $m, m' \le m_{\max}$ blocks, and put $r = m + m' < N + 1$. Suppose the field powers $2^0, 2^1, \ldots, 2^{m_{\max}}$ are distinct. Then*

$$\Pr[h(\mu) = h(\nu)] \le \frac{1}{N - r + 1} + \frac{\binom{r}{2}}{2^{n+\tau}}. \tag{23}$$

For equal-length encodings differing in exactly one block, the sharper bound (21) *applies to the full differential.*

Proof. Collisions of masked inputs. Label each hash call by its message and position. Two calls at the same position with the same message block have identical primitive inputs; treat them as a single call. At the same position, different message blocks always give different primitive inputs: a difference in the left component changes z, and a difference in the right component changes w.

For calls at distinct positions $s \ne s'$, an input collision requires

$$(2^{s-1} \oplus 2^{s'-1})L = a_s \oplus a'_{s'}, \qquad \mathsf{p}_t((2^{s-1} \oplus 2^{s'-1})R) = b_s \oplus b'_{s'}. \tag{24}$$

The coefficient is nonzero, so the first equation fixes L and holds with probability N^{-1}. Multiplication by that coefficient is invertible; the second equation therefore has rank τ and probability at most $2^{-\tau}$. Independence of L, R gives a joint probability at most $2^{-(n+\tau)}$. A union bound over at most $\binom{r}{2}$ pairs bounds the event $\mathsf{Bad}_{\mathrm{mask}}$ of any cross-position input collision by the second term of (23).

A nonzero output coefficient. Fix masks for which $\mathsf{Bad}_{\mathrm{mask}}$ does not occur. After merging identical calls, all remaining primitive inputs are distinct. Expand $h(\mu) \oplus h(\nu)$ using (7) and combine the coefficients of identical calls. For equal lengths, a differing block gives an unmerged call with nonzero coefficient 2^{m+1-s}. For unequal lengths, an extra position in the longer encoding gives such a call. Thus a nonzero coefficient always exists.

Bounding the output relation. Choose a call with a nonzero coefficient using only the fixed messages and masks, and expose the outputs of all other distinct calls. At most $r - 1$ outputs have been exposed under the selected call's tweak. Its output is therefore uniform over at least $N - r + 1$ unused values. Because its coefficient is nonzero, the equation $h(\mu) \oplus h(\nu) = 0^n$ determines at most one such value. The conditional collision probability is at most $1/(N - r + 1)$. Adding the bound on $\Pr[\mathsf{Bad}_{\mathrm{mask}}]$ proves the lemma.

Lemma 4 (Corrected aggregate bound; cf. original Lemma 4). *Let $(M^{(i)}, d^{(i)})$, $1 \le i \le q$, be fixed, distinct query pairs, with encodings $\mu^{(i)}$ of*

lengths $m^{(i)} \le m_{\max}$ *and* $\sigma = \sum_i m^{(i)}$. *Assume* $2m_{\max} < N + 1$ *and distinct field powers as in Lemma 3. Define*

$$S = \sum_{i<j} \sum_{\delta \in \mathcal{K}_{ij}} \mathsf{DP}_{\mathsf{ZHash}[\widetilde{P}]}(\mu^{(i)}, \mu^{(j)}; (0^n, \delta)). \tag{25}$$

Let c_1 *be the number of pairs whose encodings have the same length and differ at exactly one position. Then*

$$S \le \frac{c_1}{N} + \sum_{\substack{i<j \\ \text{other pairs}}} \left(\frac{1}{N - m^{(i)} - m^{(j)} + 1} + \frac{\binom{m^{(i)}+m^{(j)}}{2}}{2^{n+\tau}} \right), \tag{26}$$

where the sum ranges over the pairs not counted by c_1. *In particular,*

$$S \le \mathcal{B}(q, \sigma, m_{\max}) := \frac{\binom{q}{2}}{N - 2m_{\max} + 1} + \frac{(2m_{\max} - 1)(q-1)\sigma}{2^{n+\tau+1}}. \tag{27}$$

For $q = 2k$ *messages from* (9), $S \ge k^2/N$.

Proof. For a fixed pair, the events corresponding to different values of δ are disjoint. The inner sum in (25) is therefore $\Pr[\Delta h = 0^n,\ \Delta g \in \mathcal{K}_{ij}]$, which is at most $\Pr[\Delta h = 0^n]$. Equation (21) bounds each of the c_1 pairs by $1/N$. Applying Lemma 3 to the remaining pairs gives (26).

For the simpler bound (27), apply the general pair bound to every pair. Bound each denominator below by $N - 2m_{\max} + 1$ and use

$$\binom{m^{(i)} + m^{(j)}}{2} \le \frac{2m_{\max} - 1}{2}(m^{(i)} + m^{(j)}), \qquad \sum_{i<j}(m^{(i)} + m^{(j)}) = (q-1)\sigma.$$

For the last claim, each of the k^2 cross-family pairs differs only in its first block, with $\Delta b_1 = \langle 1 \rangle_t \in \mathcal{K}_{ij}$, and contributes $1/N$.

For bounded message lengths, Lemma 4 gives $S = O(q^2/2^n)$. Our message families show that the quadratic dependence on q is necessary. Sharper estimates for other pairs may improve the constants or the dependence on message length, but cannot restore a BBB bound for the present two-block finalization.

4 Security analysis

We use Lemma 4 to prove VOLPRF security, then derive MAC security by a standard reduction. These upper bounds match the query threshold of our attacks for fixed two-block tags and bounded message lengths. The theorems allow adaptive queries with output lengths in (3). Any advantage bound greater than one may be replaced by one.

4.1 VOLPRF security

Theorem 3 (Corrected VOLPRF security). *Let an adversary $\mathcal{A}$ make at most $q \geq 1$ distinct ZMAC+ queries, with at most $m_{\max}$ encoded blocks per query, at most σ encoded blocks in total, and at most D output blocks in total. Under the assumptions of Lemma 4, there is a TPRP adversary $\mathcal{D}$ using at most $\sigma + D + 2$ primitive queries such that*

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{ZMAC}^+[\widetilde{E}]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{tprp}}_{\widetilde{E}}(\mathcal{D}) + \frac{\binom{D}{2}}{N} + \mathcal{B}(q, \sigma, m_{\max}). \tag{28}$$

For $q = 0$, the VOLPRF advantage is zero; the budget formula is used only for $q \geq 1$.

Proof. Step 1: Primitive replacement. Replace $\widetilde{E}_K$ by an ideal TBC $\widetilde{P}$. A TPRP adversary simulates ZMAC+ using its oracle and returns the output of $\mathcal{A}$. It uses two calls to generate the masks, at most σ calls for hashing, and at most D calls for finalization. The change in advantage is at most $\mathsf{Adv}^{\mathrm{tprp}}_{\widetilde{E}}(\mathcal{D})$.

Step 2: Permutation/function switching. Replace the domain-1 permutations $\widetilde{P}^{(1,T)}$ by independent random functions $F_T : \{0,1\}^n \to \{0,1\}^n$. The usual online permutation/function coupling fails only when a fresh output of one F_T repeats a previous output under the same tweak. At most D evaluations are made, so this switch costs at most $\binom{D}{2}/N$. This remains valid for adaptively chosen inputs and repeated evaluations. Domains 0 and 2, and hence the hash, are unchanged.

Step 3: Finalization-input collisions. Compare the resulting hash-then-random-function system with $\mathcal{R}$. Within one query, finalization uses distinct tweaks. Between queries i, j, repeated finalization inputs occur exactly when $h^{(i)} = h^{(j)}$ and $g^{(i)} \oplus g^{(j)} \in \mathcal{K}_{ij}$. Until the first such repetition, couple every fresh $F_T(h)$ to a fresh independent output block of $\mathcal{R}$; both systems then return identical answers.

To handle adaptivity, first run the entire ideal interaction with $\mathcal{R}$, independently of the hidden domain-0 and domain-2 primitives. Condition on this ideal transcript and on the adversary's random coins. The query list is then fixed and independent of the hidden hash. A union bound over its pairs bounds the probability of a repeated finalization input by S in (25). Lemma 4 bounds S uniformly over all such lists. The coupled systems agree until the first repetition, so the same bound controls their distinguishing gap. This argument applies to adaptive adversaries without conditioning the hash on real responses. Adding the three bounds proves (28).

For fixed output length d and $\sigma \leq q m_{\max}$, this gives

$$\begin{aligned}\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{ZMAC}^+[\widetilde{E}]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{tprp}}_{\widetilde{E}}(\mathcal{D}) + \frac{dq(dq-1)}{2N} + \frac{q(q-1)}{2(N - 2m_{\max} + 1)} \\ + \frac{(2m_{\max} - 1)m_{\max} q(q-1)}{2^{n+\tau+1}}.\end{aligned} \tag{29}$$

For bounded d and $m_{\max}$, the terms other than the TPRP advantage are $O(q^2/N)$. This estimate is conservative: the permutation/function switch alone contributes $\binom{D}{2}/N$. We claim a matching query threshold for two-block tags, but do not claim tightness for single-block tags or for output lengths that grow with n.

4.2 MAC security

Theorem 4 (Corrected MAC security). *Suppose a forger $\mathcal{A}$ makes at most q authentication queries and one forgery attempt with output length at least $d_{\min}$. Include the forgery target in the resource counts: at most $m_{\max}$ encoded blocks per message, Σ encoded blocks in total, and Δ output blocks in total. Let $Q = q + 1$ and assume the conditions of Lemma 4. There is a TPRP adversary $\mathcal{D}$ with at most $\Sigma + \Delta + 2$ primitive queries for which*

$$\mathsf{Adv}^{\mathrm{mac}}_{\mathsf{ZMAC}^+[\widetilde{E}]}(\mathcal{A}) \le 2^{-nd_{\min}} + \mathsf{Adv}^{\mathrm{tprp}}_{\widetilde{E}}(\mathcal{D}) + \frac{\binom{\Delta}{2}}{N} + \mathcal{B}(Q, \Sigma, m_{\max}). \tag{30}$$

For fixed two-block tags, take $d_{\min} = 2$ and $\Delta \le 2Q$.

Proof. Construct a VOLPRF distinguisher that forwards the forger's authentication queries to its oracle. When the forger outputs a target, the distinguisher rejects if the target is not fresh. Otherwise, it queries the target and returns 1 exactly when the oracle's tag equals the proposed tag. In the real experiment, its acceptance probability is the forgery success probability. In the ideal experiment, the fresh target has an independent uniform tag of at least $nd_{\min}$ bits, so acceptance has probability at most $2^{-nd_{\min}}$. Apply Theorem 3 with at most $Q = q + 1$ queries and total budgets Σ, Δ.

Corollary 1 (Birthday security threshold). *Assume the field and length conditions of Theorem 3. In the ideal-TBC model, for fixed $d = 2$ and messages of at most three encoded blocks, ZMAC+ has both PRF and MAC upper bounds of order $q^2/2^n$ (plus the MAC guessing term). With $q = \Theta(2^{n/2})$, Theorems 1 and 2 give constant distinguishing advantage and constant forgery success, respectively. Thus ZMAC+ has birthday-bound security in this regime and is not a BBB-secure MAC.*

Proof. Set $D = 2q$, $\sigma \le 3q$ in (28), and $\Delta \le 2(q + 1)$, $\Sigma \le 3(q + 1)$ in (30). For fixed $m_{\max} = 3$ and admissible parameters, the resulting terms are $O(q^2/N)$, so they vanish for $q = o(\sqrt{N})$. Conversely, set $k = \lfloor \sqrt{N} \rfloor$ in Theorems 1 and 2. The distinguishing advantage and forgery probability are each at least $1 - e^{-1} - o(1)$ at the birthday scale. The upper and lower bounds therefore establish the same query threshold.

The attack needs only an n-bit data collision and a controlled tweak difference, despite the $2n$-bit tag and $(n + t)$-bit hash. For $t = n$, it contradicts (1); increasing t does not prevent the exchange. For a concrete TBC, each lower bound decreases by at most the TPRP advantage of the corresponding simulator. The forgery simulator includes the primitive calls needed to verify the final forgery.

Table 2. Ideal-TBC experiments at $n = 8$. Each row uses 3000 trials. The first row uses seed 20261015; subsequent rows increment the seed by one. Probability columns report empirical frequencies, except for p_k, which is computed exactly.

t	k	Exact p_k	Internal hit	Reversed tag	Fresh forgery	Ideal match
4	16	0.6554	0.6570	0.6583	0.6563	0.0047
8	16	0.6554	0.6683	0.6700	0.6667	0.0033
12	16	0.6554	0.6543	0.6560	0.6537	0.0033
8	4	0.0614	0.0587	0.0587	0.0587	0.0000
8	8	0.2271	0.2337	0.2340	0.2333	0.0000
8	24	0.9164	0.9180	0.9187	0.9167	0.0067

4.3 Experimental verification

The accompanying program `verify_degradation.py` implements ideal tweaked permutations by lazy sampling. For each new data input under a fixed (a, T), it chooses an output uniformly among unused n-bit values; repeated inputs reuse the recorded output. Domain-2 calls generate L, R, so the program follows the same model as the proofs. It implements the projection in (2) and the complete bit-level encoding.

For every first-block collision observed in an experiment, the program checks the forced tag reversal. It also verifies that the extension attack produces a valid tag for a fresh message, and records accidental reversals separately. At $n = 8$ and $k = 16$, the experiments cover $t < n$, $t = n$, and $t > n$. Further experiments vary k with $t = n = 8$. Table 2 reports one reproducible run. The values for $n = t = 16$ in Table 1 are evaluated directly from the exact probability formulas and published bounds, without simulation.

The experiments check the implementation and the deterministic tag relations at small block lengths. The security claims follow from the proofs above. The program requires only Python's standard library, and the source package records the seeds, trial counts, field polynomials, and numerical results needed to reproduce the tables. At a success probability near 0.65, 3000 trials have standard error about 0.009; four decimal places do not imply that statistical precision. An additional 900 trials cover $d \in \{3, 4, 6\}$ and $t \in \{4, 8, 12\}$, checking the first-two-block exchange and, for even d, complete-tag permutation and fresh-message forgery.

5 A repair by separating output domains

The attack exchanges two finalization positions within a shared domain. We retain enc and ZHash and assign each output position a separate domain, defining $\mathsf{ZMAC}^{+,\mathrm{sep}}$ by

$$(h, g) = \mathsf{ZHash}_K(\mathsf{enc}(M, d)), \qquad u_j = \widetilde{E}_K^{(\mathsf{F}_j, g)}(h), \quad 1 \le j \le d. \tag{31}$$

The labels $\mathsf{F}_1, \ldots, \mathsf{F}_{d_{\max}}$ must be distinct and separate from the hash and mask domains. Thus this construction requires an augmented tweak space supporting $d_{\max}+2$ domain labels, with an injective encoding of each label–tweak pair. The TBC in the following theorem is defined on this enlarged space.

Lemma 5 (Hash properties for the repair). *Assume $m_{\max} \le N/4$ and that the field element 2 has order greater than $m_{\max}$. For distinct encodings μ, ν of lengths $m, m' \le m_{\max}$, put $r = m + m'$ and $H = (h, g) = \mathsf{ZHash}[\widetilde{P}]$. Then, for every $\delta \in \{0,1\}^t$,*

$$\Pr[H(\mu) = H(\nu)] \le \frac{2r+2}{2^{n+\tau}}, \tag{32}$$

$$\Pr[g(\mu) \oplus g(\nu) = \delta] \le \frac{5}{2^{\tau+1}}. \tag{33}$$

Proof. Merge calls with the same position and message block, as in Lemma 3. Write the hash differences as

$$\Delta h = \bigoplus_c A_c V_c, \qquad \Delta g = B \oplus \mathsf{p}_t\left(\bigoplus_c \beta_c V_c\right), \qquad \beta_c \in \{0,1\}.$$

The coefficients depend only on the two encodings. A call appearing in only one encoding has $\beta_c = 1$. Such a call always exists. Protect one chosen call from collisions with other primitive inputs. Same-position distinct blocks cannot collide; each cross-position collision has probability at most $2^{-(n+\tau)}$ by (24). The failure probability is at most $(r-1)/2^{n+\tau}$. On its complement, fix the masks and expose all other outputs. At most $2^{n-\tau}$ values of the chosen output satisfy the projected difference, so

$$\Pr[\Delta g = \delta] \le \frac{r-1}{2^{n+\tau}} + \frac{2^{n-\tau}}{N-r+1} \le \frac{5}{2^{\tau+1}}. \tag{34}$$

The last inequality uses $r \le N/2$.

For the full-hash collision, equal-length encodings differing in exactly one block have probability zero by (21) with $\delta = 0^t$. In every other case, choose two calls u, v such that $A_u\beta_v \oplus A_v\beta_u \ne 0$. If the lengths are equal, choose calls at two differing positions of one encoding: their β coefficients are one and their A coefficients differ. If the lengths differ and a common same-position call exists, it has $\beta = 0$ and $A \ne 0$; pair it with an extra-position call of the longer encoding, whose β is one. If no common call exists, choose two positions in the longer encoding. Distinct field powers justify each nonzero determinant.

Protect these two inputs from all other inputs, at cost at most $2(r-1)/2^{n+\tau}$; collisions among unselected calls need not be excluded. After fixing the masks and exposing the other outputs, the equations $\Delta h = 0^n$, $\Delta g = 0^t$ impose $n+\tau$ independent binary linear constraints on (V_u, V_v). Indeed, the invertible two-column matrix maps this pair to a fixed n-bit value and a value with a fixed projection. There are at most $2^{n-\tau}$ satisfying pairs. Each has conditional probability at most $1/((N-r+2)(N-r+1))$, whether the two calls use the same

permutation or different ones. Hence

$$\Pr[\Delta h = 0^n, \Delta g = 0^t] \le \frac{2(r-1)}{2^{n+\tau}} + \frac{2^{n-\tau}}{(N-r+2)(N-r+1)} \le \frac{2r+2}{2^{n+\tau}}. \quad (35)$$

The AU bound concerns only a full-hash collision. It is consistent with the probability $1/N$ of the nonzero difference $(0^n, \langle 1 \rangle_t)$ used in our attack, and does not rely on the published aggregate Lemma 4.

Theorem 5 (Security of the domain-separated repair). *Under the conditions of Lemma 5, consider an adaptive adversary making at most $q \ge 1$ distinct queries, totaling at most σ encoded blocks and D output blocks. These budgets hold in every execution. There is a TPRP adversary $\mathcal{D}$ making at most $\sigma + D + 2$ primitive queries such that*

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{ZMAC}^{+,\mathrm{sep}}[\widetilde{E}]}(\mathcal{A}) \le \mathsf{Adv}^{\mathrm{tprp}}_{\widetilde{E}}(\mathcal{D}) + \frac{2(q-1)\sigma + q(q-1) + \frac{5}{4}(q-1)D}{2^{n+\tau}}. \quad (36)$$

Proof. Replace the TBC by ideal independent permutations at the stated TPRP cost. We extend the single-output hash-then-TBC argument of [JN22, Section 5.2]. Append the hidden hash randomness to the public transcript; in the ideal experiment sample this randomness independently of the random answers. Call a transcript bad if two queries have identical full hashes, or if two outputs at the same position have both equal hash tweaks and equal output values.

For a good extended transcript, all inputs and all outputs within a finalization group (F_j, g) are distinct. If $a_{j,g}$ calls use that group and the transcript contains D_ω output blocks, the real-to-ideal probability ratio is $N^{D_\omega} / \prod_{j,g} (N)_{a_{j,g}} \ge 1$, where $(N)_a = N(N-1)\cdots(N-a+1)$. The hash randomness and the adversary's choices have the same factors in both probabilities. The statistical distance is consequently at most the ideal probability of a bad transcript, as in the H-coefficient argument.

Condition on the ideal public transcript. Its encodings are independent of the hidden hash randomness. Equation (32) bounds the first bad event by

$$\sum_{i<i'} \frac{2m^{(i)} + 2m^{(i')} + 2}{2^{n+\tau}} \le \frac{2(q-1)\sigma + q(q-1)}{2^{n+\tau}}.$$

For the second event, the same conditioning gives a factor at most $5/2^{\tau+1}$ for every pair with equal public outputs at a shared position. The expected number of such equal-output pairs is at most $(q-1)D/(2N)$. To see this for adaptive queries, observe that each new query's length is chosen before its fresh independent outputs. Each comparison with an earlier output then has probability $1/N$, and pathwise

$$\sum_{i<i'} \min(d^{(i)}, d^{(i')}) \le \frac{1}{2} \sum_{i<i'} (d^{(i)} + d^{(i')}) \le \frac{(q-1)D}{2}.$$

Multiplying these bounds proves the last term in (36).

MAC security and scope. Let $\mathcal{C}(q, \sigma, D)$ denote the fraction in (36). The reduction of Theorem 4 immediately gives the one-attempt forgery bound

$$2^{-nd_{\min}} + \mathsf{Adv}^{\mathrm{tprp}}_{\widetilde{E}}(\mathcal{D}) + \mathcal{C}(q+1, \Sigma, \Delta),$$

where the fresh target is included in Σ, Δ and in the per-message length limit; the reduction uses at most $\Sigma + \Delta + 2$ primitive queries. For bounded message and output lengths, $\mathcal{C} = O(q^2/2^{n+\tau})$. In particular, $d = 2$ and $m_{\max} = 3$ give $\mathcal{C} \le (19/2)q(q-1)/2^{n+\tau}$. This is a BBB guarantee when τ grows with the security parameter; for constant τ the improvement is only a constant factor. For $d = 1$, this is the original finalization, so the proof also covers ZMAC+ restricted to single-block tags.

The repair keeps $m + d$ calls per message and two cached mask-generation calls per key. For fixed $d = 2$, an available fourth domain label suffices; general output lengths need enough distinct labels. XORing a different constant into g does not provide independent domains, and truncating g to make room for the index is not covered by this proof. Encoding d remains necessary for the VOLPRF notion: otherwise distinct requests $(M, 1)$ and $(M, 2)$ would share their first output block.

6 Counter reuse in the ZAE extension of ZMAC

Iwata et al. also propose ZAE, a deterministic authenticated encryption (DAE) scheme based on ZMAC [IMPS17b, Section 5]. Its encryption counter repeats after 2^t blocks. This gives a one-query distinguisher when the message domain is taken as stated in Figure 6 and Theorem 2 of the full version (revision of 15 December 2017, pages 23–25). The issue is a missing message-length restriction in the ZAE composition. It does not affect the MAC or PRF theorem for the original ZMAC. Throughout this section, $\widetilde{E}$ uses the augmented tweak space of original ZMAC/ZAE, including domain 10 for encryption.

Construction and security experiment. Given associated data A and a message M, ZAE first applies an injective encoding $\mathsf{encode}(A, M)$. It computes the $2n$-bit synthetic IV $V = \mathsf{ZMAC}_K(\mathsf{encode}(A, M))$, encrypts M as $C = \mathsf{IVCTRT}_K(V, M)$, and returns (V, C). This follows the SIV paradigm [RS06], with counter-in-tweak encryption as the encryption component [PS16]. Parse $V = V_L \| V_R$ into n-bit halves and set $T = \mathsf{p}_t(V_L)$ and $S = V_R$. Let $\boxplus$ denote addition modulo 2^t. For each full n-bit message block, the algorithm in Figure 6 of [IMPS17b] computes

$$C_s = M_s \oplus \widetilde{E}_K^{(10, T \boxplus s)}(S). \tag{37}$$

For a partial final block, the keystream is truncated to the required length. Our attack uses only full blocks. Domain 10 separates these encryption calls from the calls used by ZMAC. The DAE experiment compares the real encryption and decryption oracles with an encryption oracle that returns random strings of the required length and a decryption oracle that always rejects. Our adversary makes

one encryption query and no decryption queries, so it satisfies the experiment's restrictions on repeated queries and decryption queries.

Theorem 6 (One-query counter-period distinguisher for ZAE). *Suppose the ZAE message domain allows a message of $2^t + 1$ full n-bit blocks. There is a DAE adversary $\mathcal{A}_{\mathrm{ZAE}}$ making one encryption query such that*

$$\mathsf{Adv}^{\mathrm{dae}}_{\mathsf{ZAE}[\widetilde{E}]}(\mathcal{A}_{\mathrm{ZAE}}) = 1 - 2^{-n}, \tag{38}$$

for every underlying TBC $\widetilde{E}$, including an ideal TBC.

Proof. The adversary chooses empty associated data and $M^{(1)} = 0^{n(2^t+1)}$. It parses the response as the $2n$-bit IV followed by ciphertext blocks $C^{(1)}_1, \ldots, C^{(1)}_{2^t+1}$. It returns 1 if $C^{(1)}_1 = C^{(1)}_{2^t+1}$. Since $T \boxplus (1+2^t) = T \boxplus 1$, the first and last encryption calls have the same domain, tweak, and data input. Equation (37) therefore gives $C^{(1)}_1 = C^{(1)}_{2^t+1}$ for every key and every synthetic IV. In the ideal experiment, the response is a uniform string, and these two n-bit blocks agree with probability 2^{-n}. Subtracting the two probabilities yields (38).

With the default $n/2$-bit length fields, this query is encodable when n is even and $n(2^t+1) < 2^{n/2}$. The example below satisfies this condition. One query still requires $2^t + 1$ payload blocks. More generally, whenever both positions exist as full blocks,

$$C_s \oplus C_{s+2^t} = M_s \oplus M_{s+2^t}. \tag{39}$$

Thus counter repetition reveals the XOR of plaintext blocks one counter period apart. The relation holds for every synthetic IV, regardless of the hash function used to generate it.

Comparison with the stated security bound. Theorem 2 of [IMPS17b] permits $q \le 2^{n-4}$ queries and bounds their total encoded length by σ_Z blocks of n bits. It states

$$\mathsf{Adv}^{\mathrm{dae}}_{\mathsf{ZAE}[\widetilde{E}]} \le \mathsf{Adv}^{\mathrm{tprp}}_{\widetilde{E}} + \frac{3.5\sigma_Z^2}{2^{n+\tau}} + 4\left(\frac{q}{2^n}\right)^{3/2} + \frac{q}{2^{2n}}. \tag{40}$$

The underlying counter-mode result already has a length condition: Theorem 1 of Peyrin and Seurin [PS16] assumes that each message contains at most $|\mathcal{T}|$ blocks. For t-bit tweaks, this means $\lceil |M|/n \rceil \le 2^t$. Their full-version Appendix B derives the IVCTRT bound by adapting that proof. However, neither Figure 6 nor Theorem 2 of the ZMAC full version states the corresponding restriction for ZAE. The reduction therefore omits a hypothesis of the encryption component's security analysis. In particular, its step bounding the IVCTRT advantage by $\sigma_Z^2/2^{n+\tau}$ cannot apply to the unrestricted message domain: Theorem 6 holds for every IV source. If the component's restriction is instead understood as implicitly inherited, the attack lies outside that restricted domain; the issue is then an unstated hypothesis in the ZAE specification and theorem. It does not refute the CTRT theorem under its stated length condition.

For a concrete contradiction in a range where the claimed bound is small, take $n = 32$, $t = 4$, $q = 1$, and $|M| = 17 \cdot 32 = 544$. The two $n/2$-bit length fields in the original Figure 6 can represent $|A| = 0$ and $|M| = 544$. To count the encoded bits, we use the intended one-zero padding: leave a nonempty string whose length is a multiple of $n + t$ unchanged; otherwise append $1\|0^c$, where $c = (-|X| - 1) \bmod (n + t)$. The printed formula $c = |X| \bmod (n + t) - 1$ on page 5 does not generally yield complete blocks. We correct this padding typo when computing the resource count; it has no effect on the counter relation (39). The padded associated data and message have 36 and 576 bits, and the length fields contribute 32 bits. Hence

$$|\mathsf{encode}(A, M)| = 36 + 576 + 32 = 644, \qquad \sigma_Z = \lceil 644/32 \rceil = 21. \tag{41}$$

The subsequent ZMAC padding extends this string to 648 bits, which still occupies 21 blocks of n bits when rounded up. With an ideal TBC, the right-hand side of (40) evaluates to

$$\frac{3.5 \cdot 21^2}{2^{36}} + \frac{4}{2^{48}} + \frac{1}{2^{64}} \approx 2.2460895366 \cdot 10^{-8}, \qquad 1 - 2^{-32} = 0.999999999767\ldots. \tag{42}$$

The claimed upper bound is therefore incompatible with the distinguishing advantage on this message domain. The verification program checks both the numerical comparison and the repetition of primitive inputs.

Message-length restriction. The construction and its security statement must, at a minimum, limit each plaintext or encrypted payload to 2^t blocks, excluding the synthetic IV. This limit must apply to both encryption and decryption queries. For a payload of $\ell \le 2^t$ blocks, the counters $T \boxplus 1, \ldots, T \boxplus \ell$ are distinct, even when modular addition wraps around zero. Supporting longer payloads requires a counter encoding that avoids repeated primitive inputs and a new security analysis. Restoring the length restriction excludes Theorem 6; a complete verification of the ZAE bound on the restricted domain is a separate task. For $t \ge n$, the published bound is already vacuous at the length required by our attack, even if that length is admissible. The numerical contradiction is therefore relevant when $t < n$. This counter-period issue is independent of the birthday attack on ZMAC+.

7 Conclusion and outlook

The tag-swap mechanism previously observed for a DoveMAC extension also applies to the published ZMAC+. Its Lemma 4 undercounts the message pairs producing the required hash difference. Our replacement lemma, distinguisher, and fresh-message forgery establish the birthday query threshold for two-block tags and bounded message lengths under the stated field conditions. Distinguishing extends to every legal $d \ge 2$, and complete-tag forgery to every legal even d.

Separating output domains preserves ZHash and the number of primitive calls. Under explicit field-order and length conditions, our independent hash analysis and adaptive composition proof give a bound of order $q^2/2^{n+\tau}$ for bounded lengths, including the single-output case. This guarantee requires the stated domain capacity. Tighter dependence on message length, matching attacks on the repaired construction, and complete-tag forgery bounds for odd $d > 1$ remain open questions.

The ZAE theorem must explicitly retain its encryption component's per-message length restriction. Without it, one sufficiently long query reveals repeated keystream blocks. Verifying the complete ZAE bound on the restricted domain remains separate from the ZMAC+ repair; the original ZMAC MAC and PRF theorems are unaffected.

References

[BLNS21] Xavier Bonnetain, Gaëtan Leurent, María Naya-Plasencia, and André Schrottenloher. Quantum Linearization Attacks. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021*, Lecture Notes in Computer Science, pages 422–452, Cham, 2021. Springer International Publishing.

[GJMN16] Robert Granger, Philipp Jovanovic, Bart Mennink, and Samuel Neves. Improved Masking for Tweakable Blockciphers with Applications to Authenticated Encryption. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, Lecture Notes in Computer Science, pages 263–293, Berlin, Heidelberg, 2016. Springer.

[GLN19] Tony Grochow, Eik List, and Mridul Nandi. DoveMAC: A TBC-based PRF with Smaller State, Full Security, and High Rate. *IACR Transactions on Symmetric Cryptology*, pages 43–80, September 2019.

[IMPS17a] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A Fast Tweakable Block Cipher Mode for Highly Secure Message Authentication. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, Lecture Notes in Computer Science, pages 34–65, Cham, 2017. Springer International Publishing.

[IMPS17b] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A Fast Tweakable Block Cipher Mode for Highly Secure Message Authentication, 2017. Added new possible instances using "key + tweak" constructions, corrected Skinny performance estimations.

[JN22] Ashwin Jha and Mridul Nandi. A Survey on Applications of H-Technique: Revisiting Security Analysis of PRP and PRF. *Entropy*, 24(4):462, April 2022.

[LN17] Eik List and Mridul Nandi. ZMAC+ – An Efficient Variable-output-length Variant of ZMAC. *IACR Transactions on Symmetric Cryptology*, pages 306–325, December 2017.

[LNS18] Gaëtan Leurent, Mridul Nandi, and Ferdinand Sibleyras. Generic Attacks Against Beyond-Birthday-Bound MACs. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 306–336, Cham, 2018. Springer International Publishing.

[LPTY16] Atul Luykx, Bart Preneel, Elmar Tischhauser, and Kan Yasuda. A MAC Mode for Lightweight Block Ciphers. In Thomas Peyrin, editor, *Fast Software Encryption*, pages 43–59, Berlin, Heidelberg, 2016. Springer.

[Nai18] Yusuke Naito. On the Efficiency of ZMAC-Type Modes. In Jan Camenisch and Panos Papadimitratos, editors, *Cryptology and Network Security*, pages 190–210, Cham, 2018. Springer International Publishing.

[PS16] Thomas Peyrin and Yannick Seurin. Counter-in-Tweak: Authenticated Encryption Modes for Tweakable Block Ciphers. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, Lecture Notes in Computer Science, pages 33–63, Berlin, Heidelberg, 2016. Springer.

[Rog04] Phillip Rogaway. Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004*, Lecture Notes in Computer Science, pages 16–31, Berlin, Heidelberg, 2004. Springer.

[RS06] Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, Lecture Notes in Computer Science, pages 373–390, Berlin, Heidelberg, 2006. Springer.

[Yas10] Kan Yasuda. The Sum of CBC MACs Is a Secure PRF. In Josef Pieprzyk, editor, *Topics in Cryptology - CT-RSA 2010*, Lecture Notes in Computer Science, pages 366–381, Berlin, Heidelberg, 2010. Springer.

[Yas11] Kan Yasuda. A New Variant of PMAC: Beyond the Birthday Bound. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, Lecture Notes in Computer Science, pages 596–609, Berlin, Heidelberg, 2011. Springer.