IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2019, No. 2, pp. 1–54. DOI:10.13154/tosc.v2019.i2.1-54

ZOCB and ZOTR: Tweakable Blockcipher Modes for Authenticated Encryption with Full Absorption

Zhenzhen Bao[1,2], Jian Guo[1], Tetsu Iwata[3] and Kazuhiko Minematsu[4]

[1] Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, Singapore
{zzbao,guojian}@ntu.edu.sg
[2] Strategic Centre for Research in Privacy-Preserving Technologies and Systems, Nanyang Technological University, Singapore, Singapore
[3] Nagoya University, Nagoya, Japan
tetsu.iwata@nagoya-u.jp
[4] NEC corporation, Tokyo, Japan
k-minematsu@nec.com

Abstract. We define ZOCB and ZOTR for nonce-based authenticated encryption with associated data, and analyze their provable security. These schemes use a tweakable blockcipher (TBC) as the underlying primitive, and fully utilize its input to process a plaintext and associated data (AD). This property is commonly referred to as full absorption, and this has been explored for schemes based on a permutation or a pseudorandom function (PRF). Our schemes improve the efficiency of TBC-based counterparts of OCB and OTR called ΘCB3 (Krovetz and Rogaway, FSE 2011) and $\mathbb{OTR}$ (Minematsu, EUROCRYPT 2014). Specifically, ΘCB3 and $\mathbb{OTR}$ have an independent part to process AD, and our schemes integrate this process into the encryption part of a plaintext by using the tweak input of the TBC. Up to a certain length of AD, ZOCB and ZOTR completely eliminate the independent process for it. Even for longer AD, our schemes process it efficiently by fully using the tweak input of the TBC. For this purpose, based on previous tweak extension schemes for TBCs, we introduce a scheme called XTX^*. To our knowledge, ZOCB and ZOTR are the first efficiency improvement of ΘCB3 and $\mathbb{OTR}$ in terms of the number of TBC calls. Compared to Sponge-based and PRF-based schemes, ZOCB and ZOTR allow fully parallel computation of the underlying primitive, and have a unique design feature that an authentication tag is independent of a part of AD. We present experimental results illustrating the practical efficiency gain and clarifying the efficiency cost for it with a concrete instantiation. The results show that for long input data, our schemes have gains, while we have efficiency loss for short input data.

Keywords: ZOCB · ZOTR · Authenticated encryption · Associated data · Tweakable blockcipher · Provable security

1 Introduction

Nonce-Based Authenticated Encryption. Authenticated encryption (AE) is a symmetric-key cryptographic function for simultaneously providing confidentiality and integrity of plaintexts. Starting from the formalization by Katz and Yung [KY00] and Bellare and Namprempre [BN00, BN08], and the seminal constructions by Jutla [Jut01, Jut08], the practical significance of AE has been widely recognized, and it grows as one of the most active research areas in symmetric-key cryptography.

Received: 2018-09-01, Revised: 2019-03-01, Accepted: 2019-05-01, Published: 2019-06-11

Associated data (AD) is data that is authenticated but not encrypted, and AE with associated data (AEAD), formalized by Rogaway [Rog02], takes both a plaintext and AD as input. AEAD provides confidentiality of plaintexts and integrity of both plaintexts and AD. The most basic form of AEAD is based on a nonce, and is called nonce-based AEAD (NAE)[1]. A nonce is a non-repeating value for each encryption, and can be realized for instance with a counter. NAE is commonly built as a mode of operation of a blockcipher. However, there is often an inherent limitation on the security caused by the birthday paradox on the input or output of a blockcipher, which ensures only $n/2$-bit security of NAE if a blockcipher with n-bit blocks is used. The $n/2$-bit security is commonly referred to as up to the birthday bound (upBB) security. Possible solutions to break this barrier exist, i.e. NAE with beyond the birthday bound (BBB) security, however, they come with an extra computational cost.

One direction to overcome the obstacle is to use a tweakable blockcipher (TBC) as the underlying primitive instead of classical blockciphers. A TBC was formalized by Liskov, Rivest, and Wagner [LRW02, LRW11], and it has an extra t-bit tweak input to provide variability, i.e., it realizes a family of 2^t independent blockciphers indexed by the tweak. Starting from the early Hasty Pudding Cipher [Sch98], we see a growing number of concrete proposals, including Threefish (in Skein [FLS$^+$10]), Deoxys-BC [JNP14a], Joltik-BC [JNP14b], and KIASU-BC [JNP14c] from the TWEAKEY framework [JNP14d], and Scream [GLS$^+$14], where the last four schemes were submitted to CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) [CAE]. We also see other examples including SKINNY [BJK$^+$16], QARMA [Ava17], and CRAFT [BLMR19]. See also TBCs in the proposals for the NIST Lightweight Cryptography project [NIS].

TBC-Based NAE. One of the most well known TBC-based NAE schemes is ΘCB3 by Krovetz and Rogaway [KR11]. The basic form of ΘCB3 called TAE was already proposed by Liskov et al. in 2002 [LRW02, LRW11], and OCB1 was proposed by Rogaway in 2004 [Rog04][2]. The seminal OCB in [Rog04] is a blockcipher-based realization of OCB1 and ΘCB3, and has been standardized in ISO [ISO09] and IETF [KR14]. The latest version called OCB3 in [KR11] is provably secure with upBB security, and its security is well understood, as it has evolved over the years and multiple security proofs are published [RBBK01, Rog02, RBB03, Rog04, KR11][3]. See also a series of papers [AY13, ZWH16, ADL17, BN17] for analyses of the provable security aspect of OCB.

ΘCB3 is simple, parallelizable, and efficient. It uses one call to a TBC to process one n-bit input block, where n is the block length in bits of the TBC. We note that ΘCB3 itself was not designed as a standalone TBC-based NAE scheme, but it was introduced as an abstraction of OCB to simplify the security proof. However, due to the strong advantages, a number of concrete NAE proposals employ ΘCB3 with dedicated TBCs. For example, there are multiple CAESAR submissions [CAE] using ΘCB3, including Deoxys [JNP14a], Joltik [JNP14b], KIASU [JNP14c], and Scream [GLS$^+$14]. See also proposals for the NIST Lightweight Cryptography project [NIS], OPP [GJMN16] for permutation-based instantiations of ΘCB3 that uses a (tweakable) Even-Mansour construction[4], and a construction by Naito [Nai17].

Despite its wide adoption in concrete constructions and in-depth analyses of the provable security aspect, there is no essential efficiency improvement of TBC-based NAE in terms of the number of TBC calls since the appearance of TAE and OCB1 in the early 2000s. From a different perspective, a blockcipher-based NAE scheme OTR and its TBC-based counterpart $\mathbb{OTR}$ were designed by Minematsu [Min14]. These schemes improve OCB and

[1] We use NAE instead of NAEAD.

[2] TAE is defined as a plain AE scheme that does not take AD as input, and ΘCB3 is a refinement of OCB1.

[3] We note that issues in the security proof in [Rog04] are reported [IM18, Poe18, Iwa18, IIMP19].

[4] OPP is closer to OCB as the underlying permutation has no native tweak input.

ΘCB3 by removing the necessity of the decryption routine of the underlying blockcipher or TBC (this property is often called as the inverse-freeness). However, OTR and $\mathbb{OTR}$ still do not improve the number of blockcipher or TBC calls of OCB and ΘCB3.

ZOCB and ZOTR. We present the first improvement to ΘCB3 in terms of the number of TBC calls. We follow the natural approach of using the input of the underlying primitive as much as possible. This approach is often called full absorption of the underlying primitive, which has been studied for NAE based on a permutation or a pseudorandom function (PRF). For instance for Sponge-based constructions [BDPA08, BDPA11], full absorption by integrating the AD process into the encryption process has been studied by Sasaki and Yasuda [SY15] and by Mennink et al. [MRV15]. The work by Reyhanitabar et al. [RVV15] shows full absorption for PRF-based NAE. In the authentication-only scenario, ZMAC [IMPS17] can be seen as a TBC-based MAC with full absorption, while the related deterministic AEAD (DAE) called ZAE [IMPS17] does not achieve full absorption, and as far as we know, the possibility of TBC-based NAE with full absorption has not been explored prior to our work.

Our first proposal, which we call ZOCB, achieves full absorption by effectively integrating the authentication process of AD into the encryption process of a plaintext. The difficulty of TBC-based NAE to improve ΘCB3 is that invertibility of the TBC has to be maintained for decryption, which is only ensured for one of the two inputs of the TBC. This issue was irrelevant in the previous works of ZMAC or ZAE. Likewise, the issue of invertibility is non-existent in Sponge-based or PRF-based constructions.

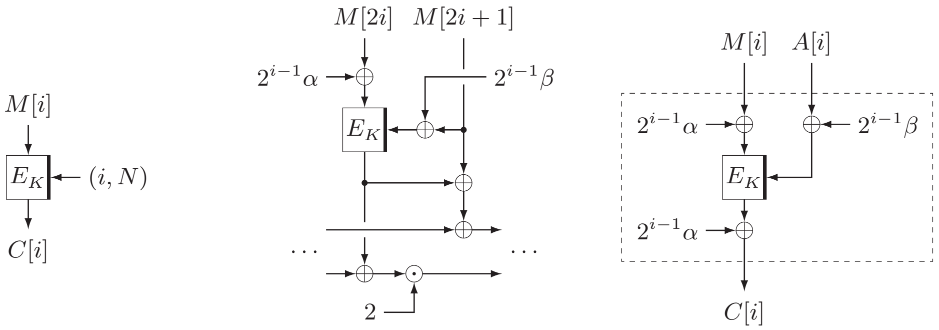
Suppose that we have a TBC with n-bit blocks and t-bit tweaks. Our design approach is to use the t-bit tweak input space of the TBC to process a block of AD, while we use the n-bit input space to process a block of plaintext, as invertibility is not necessary in the process of AD but is needed for the process of a plaintext. The t-bit tweak space was used only for a block counter and domain separation in ΘCB3. This approach in turn implies that a block counter has to be realized by some other means, and for this purpose we follow the design of ZMAC and ZAE to use the so-called masks to realize the block counter. We introduce a *tweak extension scheme*, called XTX^*, which combines XTX [MI15] and XT [IMPS17], and uses the idea of XEX* [Rog04].

Our design, like ZMAC and ZAE, combines a TBC and a mask to obtain the efficiency improvement. See Fig. 1 for the schematic comparison between ΘCB3, ZMAC/ZAE, and ZOCB, and Fig. 2 for the design approaches in [SY15, MRV15, RVV15]. In ZOCB, the i-th plaintext block $M[i]$ and AD block $A[i]$ are processed simultaneously in a single primitive call as in Sponge-based and PRF-based NAE [SY15, MRV15, RVV15], while ZOCB is structurally different from them in that ZOCB is fully parallelizable, and hence there is no chaining value that can be utilized as in [SY15, MRV15, RVV15]. Therefore, the design of ZOCB can be characterized as having advantages of both:

- full absorption by processing $M[i]$ and $A[i]$ simultaneously in one primitive call as in [SY15, MRV15, RVV15], which was not explored for TBC-based NAE, and
- full parallelizability as in ΘCB3 and ZMAC/ZAE, which is not achieved by [SY15, MRV15, RVV15].

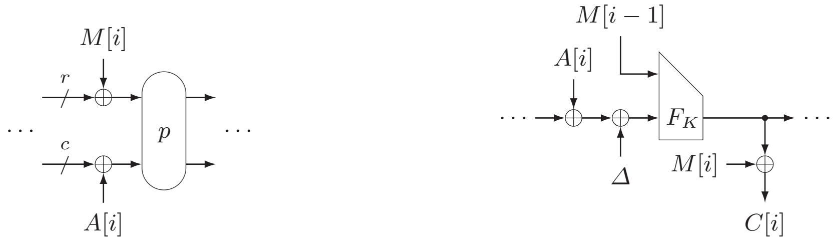
It also has a unique design feature that a tag is independent of a part of the AD blocks, which is needed to maintain the parallelizability for the lack of chaining values. As we detail below, our contribution is to show that the approach does work in the context of TBC-based NAE, both in terms of provable security and implementation aspects.

Given a TBC with n-bit blocks and t-bit tweaks, when AD is shorter than mt bits, where m is the length of a plaintext in n-bit blocks, ZOCB needs only about m calls of the underlying TBC, which improves ΘCB3 that needs $m + a$ calls, where a is the length of AD in n-bit blocks. Even if AD is longer than mt bits, ZOCB is still faster than ΘCB3, as



(a) ΘCB3 **(b)** ZMAC/ZAE **(c)** ZOCB

Figure 1: **(a)** The processing of the i-th plaintext block $M[i]$ in ΘCB3, where E_K takes both a nonce N and a block counter i as the tweak input. **(b)** The processing of two consecutive plaintext blocks $M[2i]$ and $M[2i+1]$ in ZMAC/ZAE. **(c)** The processing of a plaintext block $M[i]$ and an AD block $A[i]$ in ZOCB. They are processed simultaneously in a single TBC call. In these figures, E_K is the underlying TBC. For simplicity, the domain separation is omitted and the figures assume $n = t$. α and β in ZMAC/ZAE and ZOCB are n-bit strings that depend on the key and/or nonce. The dashed box in ZOCB can be seen as a TBC with large tweak space that takes $A[i]$, i, and N as the tweak input. Observe the difference to ΘCB3, where the TBC in the dashed box takes $A[i]$ as the tweak input in addition to (i, N). The difference to ZMAC/ZAE is the simultaneous processing of $M[i]$ and $A[i]$.



(a) Sponge-based scheme **(b)** PRF-based scheme

Figure 2: **(a)** The Sponge-based scheme with full absorption [SY15, MRV15]. p is a public permutation. **(b)** The PRF-based scheme with full absorption [RVV15]. F_K is a PRF and $\varDelta$ is a mask that depends on the key, nonce, and the block counter. These schemes are non-parallelizable, and full absorption is achieved by utilizing the chaining values.

we use a standalone upBB secure AD authentication from [IMPS17] that can process $n + t$ bits per one TBC call. Therefore, our proposal uniformly reduces the required number of TBC calls for any input length, neglecting the constant number of calls during the setup.

We also define ZOTR as an improvement of $\mathbb{OTR}$. These schemes do not need a decryption routine of a TBC, and ZOTR uses the same number of TBC calls as ZOCB. See Table 1 for the comparison with other schemes.

With respect to the tag generation process in ZOCB and ZOTR, in the schemes listed in Table 1 and in the previous designs [SY15, MRV15, RVV15], a tag is a function of the entire AD. We also see that all the secure ones (A1–A8) in [NRS14], a tag is a function of the entire AD. In contrast to this, in ZOCB and ZOTR, a tag is obtained as the TBC-encryption of a checksum of plaintext blocks, and the tag is independent of a part of the AD blocks. As we outline below, ZOCB and ZOTR achieve provable authenticity despite the use of this type of AD-independent tags. We remark that a tag of A9 in [NRS14] is

Table 1: Comparison of our schemes with other BC/TBC-based NAE/DAE schemes and FKD. BC stands for a blockcipher. The block length of BC/TBC is n bits, and the tweak length of the TBC is t bits. “Func.” is the functionality, “Prim.” is the primitive, and “# of calls” refers to the number of BC/TBC calls for at-bit AD and mn-bit plaintexts for the case $t = n$, neglecting constant number of calls. “Inv.” denotes the inverse-freeness, and “Para.” shows if the primitive can be called in parallel. “Mul” in GCM denotes $\mathrm{GF}(2^n)$ multiplication. Full-state Keyed Duplexing (“FKD”) is based on a permutation (“Perm”), where b denotes the width of the permutation, c is a security related parameter called capacity, and $r = b - c$. # of calls is written for AD of $|A|$ bits and a plaintext of $|M|$ bits.

Scheme	Func.	Prim.	# of calls		Inv.	Para.	Security	Ref.
			$a < m$	$a \geq m$				
GCM	NAE	BC, Mul	$(a + m)$ Mul $+ m$ BC		Y	Y	$n/2$	[MV04, NIS07]
OCB3	NAE	BC	$a + m$		N	Y	$n/2$	[KR11]
OTR	NAE	BC	$a + m$		Y	Y	$n/2$	[Min14]
ΘCB3	NAE	TBC	$a + m$		N	Y	n	[Rog04, KR11]
𝕆𝕋ℝ	NAE	TBC	$a + m$		Y	Y	n	[Min14]
SCT	DAE	TBC	$a + 2m$		Y	Y	$n/2$ (n as NAE)	[PS16]
ZAE	DAE	TBC	$a/2 + 3m/2$		Y	Y	$\min\{n, (n + t)/2\}$	[IMPS17]
FKD	NAE	Perm	$\lvert M\rvert/r$ (if $\lvert A\rvert/c \leq \lvert M\rvert/r$) $(\lvert A\rvert + \lvert M\rvert)/b$ (else)		Y	N	$\min\{\lvert K\rvert, c/2\}$	[BDPA11, MRV15]
ZOCB	NAE	TBC	m	$(a + m)/2$	N	Y	$\min\{n, (n + t)/2\}$	Sect. 5
ZOTR	NAE	TBC	m	$(a + m)/2$	Y	Y	$\min\{n, (n + t)/2\}$	Sect. 6

independent of AD, however, it does not achieve a desirable provable security result.

We point out that, unlike Sponge-based and PRF-based schemes, the number of TBC calls was not reduced without cost. The use of a mask requires a doubling operation for each block, and this can add efficiency cost in practice. We also note that the tweak does not behave like a counter, and this can add the computational cost to update the tweak depending on the TBC in use. We discuss in Sect. 9 a possible solution to mitigate the cost.

The length of AD is usually shorter than a plaintext, in which case the efficiency gain of our schemes becomes marginal, or can be worse if the cost for the doubling operation is larger than the efficiency gain. Still, real-world protocols tend to require long header information for handling complex network setting, say IPv6 header has a minimum of 40 bytes (whereas it was 20 bytes for IPv4) that can be arbitrarily extended, and is attached to any packet [Hag09].

Provable Security. We analyze the provable security of ZOCB and ZOTR. Under the standard assumption about the underlying TBC, we show that they have $\min\{n, (n+t)/2\}$-bit security both in privacy and authenticity, well beyond the upBB security ($n/2$-bit security) for a reasonable tweak length. As mentioned above, in ZOCB and ZOTR, a tag is independent of a part of AD blocks. The provable security of authenticity works because in *the decryption*, the tag is a function of both the ciphertext and AD, and hence changing one of the AD blocks in decryption will affect the tag. The proofs are not mathematically involved, however, they still require careful analysis due to the integration mechanism of the AD process.

We remark that the provable security bounds are worse than those of ΘCB3 and 𝕆𝕋ℝ that are perfectly secure in privacy and have n-bit security in authenticity. This implies that, as NAE schemes, they have n-bit security by taking the minimum of the privacy and authenticity security bounds. For ZOCB and ZOTR, by using a TBC with $n \leq t$, they achieve n-bit security as NAE schemes and hence in this case our schemes do not have security loss compared to ΘCB3 and $\mathbb{OTR}$. We also note that our security analysis is in the classical single key scenario, and the security against related-key attacks, side-channel attacks, or multi-key security is not a part of the design criteria. We also do not consider nonce-misuse scenario or decryption misuse scenario. In particular, as NAE schemes, our proposals are insecure if a nonce is repeated in encryption.

Instantiation and Implementation. In order to see the practical efficiency gain, we instantiate ZOCB and ZOTR to measure the speed. We propose an AES-based TBC called TAES, which is a natural construction of a TBC where we simply put the concatenation of a 128-bit tweak and 128-bit key into the key-scheduling function of AES-256. AES-256 was not designed to be used as a TBC, but TAES immediately gives a very efficient TBC that could be used in various other TBC-based schemes. We claim 128-bit security of TAES in the single-key setting, and we discuss the reason why the most efficient attack on AES-256 is not applicable to TAES. We propose TAES as a target for cryptanalysts, and we leave further detailed security analysis as an open problem.

We implemented TAES-ZOCB/ZOTR on an Intel Skylake CPU using AES-NI and compared them with TAES-ΘCB3. We also tested an Intel Haswell CPU. Furthermore, to see the advantage with an existing TBC, we implemented SKINNY-ZOCB/ZOTR and compared them with SKINNY-ΘCB3, where we use the version called SKINNY-128-256. The results show that our schemes have practical efficiency gains for long input data. We also see that the efficiency loss for short input data is limited.

2 Preliminaries

Notation. We first introduce notation that we use throughout this paper. Let $\{0,1\}^*$ be the set of all finite bit strings, including the empty string ε. For $X \in \{0,1\}^*$, $|X|$ is its length in bits. For an integer $i \geq 0$, $\{0,1\}^i$ is the set of all bit strings of i bits, and $\{0,1\}^{\leq i}$ is the set of all bit strings of at most i bits, including ε. For an integer $\ell \geq 1$, $|X|_\ell$ is the length of $X \in \{0,1\}^*$ in ℓ-bit blocks, which is defined as $|X|_\ell = \lceil |X|/\ell \rceil$ if $X \neq \varepsilon$, and $|X|_\ell = 1$ if $X = \varepsilon$. For two bit strings X and Y, $X \,\|\, Y$ is their concatenation. We also write this as XY if it is clear from the context. Let 0^i be the string of i zero bits, and for instance we write 10^i for $1 \,\|\, 0^i$. For $X \in \{0,1\}^*$ and $\ell \geq 1$, we define the one-zero-padding function, $\mathsf{ozp}_\ell(X)$, as $\mathsf{ozp}_\ell(X) = X$ if $X \neq \varepsilon$ and $|X| \bmod \ell = 0$, i.e., if $|X|$ is a positive multiple of ℓ bits. Otherwise, $\mathsf{ozp}_\ell(X) = X \,\|\, 10^{\ell-1-(|X| \bmod \ell)}$. For $X \in \{0,1\}^*$ with $|X| \geq i$, $\mathsf{msb}_i(X)$ is the first (left) i bits of X, and $\mathsf{lsb}_i(X)$ is the last (right) i bits of X.

For $X \in \{0,1\}^*$ and $\ell \geq 1$, we define the parsing operation of X into ℓ-bit blocks as $(X[1], \ldots, X[x]) \xleftarrow{\ell} X$, where for $X \neq \varepsilon$, we let $x = |X|_\ell$, $X[1] \,\|\, \ldots \,\|\, X[x] = X$, $|X[i]| = \ell$ for $1 \leq i \leq x-1$, and $1 \leq |X[x]| \leq \ell$. For $X = \varepsilon$, we let $x = 1$, $X[x] \xleftarrow{\ell} X$, and $X[1] = \varepsilon$. For $X \in \{0,1\}^{\ell+i}$ with $\ell \geq 1$ and $i \geq 0$, we define another type of parsing operation $(X[1], X[2]) \xleftarrow{\ell,i} X$ as $X[1] = \mathsf{msb}_\ell(X)$ and $X[2] = \mathsf{lsb}_i(X)$. We also write this as $(X[1], X[2]) \xleftarrow{\ell,*} X$ for $X[1] = \mathsf{msb}_\ell(X)$ and $X[2] = \mathsf{lsb}_{|X|-\ell}(X)$. Note that when $|X| = \ell$, we have $(X[1], X[2]) \xleftarrow{\ell,*} X$, where $X[1] = X$ and $X[2] = \varepsilon$.

We use a tweakable blockcipher (TBC) as the underlying primitive. A TBC is a keyed function $E : \mathcal{K} \times \mathcal{W} \times \{0,1\}^n \to \{0,1\}^n$, where $\mathcal{K}$ is the key space, $\mathcal{W}$ is the tweak space, and n is the block length, such that for any key and tweak $(K, W) \in \mathcal{K} \times \mathcal{W}$, $E(K, W, \cdot)$ is a permutation over $\{0,1\}^n$. We emphasize that E in this paper is a TBC (which is often denoted as $\widetilde{E}$). Following [PS16, IMPS17], we consider the case where $\mathcal{W} = \mathcal{I} \times \{0,1\}^t$ for some $\mathcal{I} \subset \mathbb{N} = \{0, 1, \ldots\}$ and $t \geq 0$, where t is called the (effective) tweak length. For $(i, W) \in \mathcal{I} \times \{0,1\}^t = \mathcal{W}$, we write $E_K((i, W), M)$, $E_K^i(W, M)$, or

$E_K^{i,W}(M)$ for $E(K,(i,W),M)$ interchangeably. For $(i,W) \in \mathcal{W}$, the inverse permutation $D(K,(i,W),M)$ of $E(K,(i,W),M)$ is written as $D_K((i,W),M)$, $D_K^i(W,M)$, or $D_K^{i,W}(M)$. For $i,\ell \in \mathbb{N}$ with $0 \le i \le 2^\ell - 1$, we write the standard ℓ-bit representation of i as $[i]_\ell$. For instance we write $E_K^{3,[1]_t}(0^n)$ to mean $E(K,(3,0^{t-1}1),0^n)$.

Given a TBC $E : \mathcal{K} \times \mathcal{W} \times \{0,1\}^n \to \{0,1\}^n$ with $\mathcal{W} = \mathcal{I} \times \{0,1\}^t$, for $X \in \{0,1\}^*$, we define the msb-or-padding function $\mathsf{mop}_t(X)$ as

$$\mathsf{mop}_t(X) \overset{\text{def}}{=} \begin{cases} \mathsf{msb}_t(X) & \text{if } |X| \ge t\,, \\ X \,\|\, 0^{t-|X|} & \text{if } |X| < t\,. \end{cases}$$

Observe that for any $X \in \{0,1\}^*$, we have $|\mathsf{mop}_t(X)| = t$. We note that, for $Y \in \{0,1\}^t$ and $X \in \{0,1\}^n$, $Y \oplus \mathsf{mop}_t(X)$ corresponds to $Y \oplus_t X$ used in ZMAC [IMPS17].

We use a multiplication in the Galois Field $\mathrm{GF}(2^n)$ of 2^n elements. The element is represented as an n-bit string $a_{n-1}\dots a_1a_0$, or an integer $\sum_{0\le i\le n-1} a_i 2^i$. We assume that $2 \in \mathrm{GF}(2^n)$, which is $0^{n-1}10$ in an n-bit string, generates all the non-zero elements, and thus the multiplicative order of 2 is $2^n - 1$. We write $2 \cdot a$ to mean the multiplication of $a \in \mathrm{GF}(2^n)$ by 2, which is called the doubling operation, and we write $2^i \cdot a$ or $2^i a$ if we apply the doubling operation i times on a. Implementation of doubling over $\mathrm{GF}(2^n)$ is simple. It is one-bit logical shift of an n-bit variable (which represents the coefficient vector of a polynomial in $\mathrm{GF}(2^n)$) followed by XOR of a constant representing the irreducible polynomial defining the field. See e.g. [Rog04].

TPRP and TSPRP Notions. We assume that TBC $E : \mathcal{K} \times \mathcal{W} \times \{0,1\}^n \to \{0,1\}^n$ is a tweakable pseudorandom permutation (TPRP) [LRW02, LRW11]. Let $\mathrm{Perm}(\mathcal{W},n)$ be the set of all functions $\mathsf{E} : \mathcal{W} \times \{0,1\}^n \to \{0,1\}^n$ such that for any $W \in \mathcal{W}$, $\mathsf{E}(W,\cdot)$ is a permutation over $\{0,1\}^n$. We say that E is a tweakable uniform random permutation (TURP) if $\mathsf{E} \overset{\$}{\leftarrow} \mathrm{Perm}(\mathcal{W},n)$, and for an adversary $\mathcal{B}$, we define

$$\mathbf{Adv}_E^{\mathrm{tprp}}(\mathcal{B}) \overset{\text{def}}{=} \Pr\left[\mathcal{B}^{E_K(\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{B}^{\mathsf{E}(\cdot,\cdot)} \Rightarrow 1\right]\,,$$

where the first probability is taken over $K \overset{\$}{\leftarrow} \mathcal{K}$ and the randomness of $\mathcal{B}$, and the last one is taken over $\mathsf{E} \overset{\$}{\leftarrow} \mathrm{Perm}(\mathcal{W},n)$ and $\mathcal{B}$.

We also assume that TBC $E : \mathcal{K} \times \mathcal{W} \times \{0,1\}^n \to \{0,1\}^n$ is a tweakable strong pseudorandom permutation (TSPRP) [LRW02, LRW11]. For TURP $\mathsf{E} \overset{\$}{\leftarrow} \mathrm{Perm}(\mathcal{W},n)$, let D be the set of inverse permutations, i.e., for any $W \in \mathcal{W}$, $\mathsf{D}(W,\cdot)$ is the inverse permutation of $\mathsf{E}(W,\cdot)$. For an adversary $\mathcal{B}$, we define

$$\mathbf{Adv}_E^{\mathrm{tsprp}}(\mathcal{B}) \overset{\text{def}}{=} \Pr\left[\mathcal{B}^{E_K(\cdot,\cdot),D_K(\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{B}^{\mathsf{E}(\cdot,\cdot),\mathsf{D}(\cdot,\cdot)} \Rightarrow 1\right]\,,$$

where the probabilities are over $K \overset{\$}{\leftarrow} \mathcal{K}$ and $\mathcal{B}$ for the first one, and $\mathsf{E} \overset{\$}{\leftarrow} \mathrm{Perm}(\mathcal{W},n)$ and $\mathcal{B}$ for the last one.

AEAD Scheme. Let $\Pi = (\Pi.\mathrm{Enc}, \Pi.\mathrm{Dec})$ be an AEAD scheme. The deterministic encryption algorithm $\Pi.\mathrm{Enc}$ takes a key $K \in \mathcal{K}_\Pi$ and a tuple (N,A,M) of a nonce $N \in \mathcal{N}_\Pi$, associated data (AD) $A \in \mathcal{A}_\Pi$, and a plaintext $M \in \mathcal{M}_\Pi$ as input, and returns a ciphertext and a tag $(C,T) \in \mathcal{C}_\Pi \times \mathcal{T}_\Pi$, where $\mathcal{K}_\Pi$ is the key space, $\mathcal{N}_\Pi$ is the nonce space, $\mathcal{A}_\Pi$ is the AD space, $\mathcal{M}_\Pi$ is the plaintext space, $\mathcal{C}_\Pi$ is the ciphertext space, and $\mathcal{T}_\Pi$ is the tag space. We write $(C,T) \leftarrow \Pi.\mathrm{Enc}_K(N,A,M)$ for $(C,T) \leftarrow \Pi.\mathrm{Enc}(K,N,A,M)$. The deterministic decryption algorithm $\Pi.\mathrm{Dec}$ takes $K \in \mathcal{K}_\Pi$ and $(N,A,C,T) \in \mathcal{N}_\Pi \times \mathcal{A}_\Pi \times \mathcal{C}_\Pi \times \mathcal{T}_\Pi$ as input, and returns $M \in \mathcal{M}_\Pi$ or the distinguished reject symbol $\perp$. We write $M \leftarrow \Pi.\mathrm{Dec}_K(N,A,C,T)$ or $\perp \leftarrow \Pi.\mathrm{Dec}_K(N,A,C,T)$ for $M \leftarrow \Pi.\mathrm{Dec}(K,N,A,C,T)$ or $\perp \leftarrow \Pi.\mathrm{Dec}(K,N,A,C,T)$, and require that $M \leftarrow \Pi.\mathrm{Dec}_K(N,A,\Pi.\mathrm{Enc}_K(N,A,M))$ holds for all $(K,N,A,M) \in \mathcal{K}_\Pi \times \mathcal{N}_\Pi \times \mathcal{A}_\Pi \times \mathcal{M}_\Pi$.

Privacy Notion. The privacy notion we consider captures the indistinguishably of a ciphertext by a nonce-respecting adversary with a chosen plaintext attack [Rog02]. A privacy adversary $\mathcal{A}$ has access to the encryption oracle $\Pi.\mathrm{Enc}_K$ or a random-bits oracle which we write the \$-oracle. The encryption oracle takes $(N, A, M) \in \mathcal{N}_\Pi \times \mathcal{A}_\Pi \times \mathcal{M}_\Pi$ as input and returns $(C, T) \leftarrow \Pi.\mathrm{Enc}_K(N, A, M)$. The \$-oracle takes $(N, A, M) \in \mathcal{N}_\Pi \times \mathcal{A}_\Pi \times \mathcal{M}_\Pi$ as input and returns a uniform random string $(C, T) \xleftarrow{\$} \{0,1\}^{|\Pi.\mathrm{Enc}_K(N,A,M)|}$, i.e., a uniform random string of the same length as the output of $\Pi.\mathrm{Enc}_K(N, A, M)$. The privacy advantage is defined as

$$\mathbf{Adv}_{\Pi}^{\mathrm{priv}}(\mathcal{A}) \overset{\mathrm{def}}{=} \Pr\left[\mathcal{A}^{\Pi.\mathrm{Enc}_K(\cdot,\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\$(\cdot,\cdot,\cdot)} \Rightarrow 1\right] ,$$

where the first probability is over $K \xleftarrow{\$} \mathcal{K}_\Pi$ and $\mathcal{A}$, and the last is over the \$-oracle and $\mathcal{A}$. We say that $\mathcal{A}$ in the privacy game is nonce-respecting if it does not make two queries with the same nonce, regardless of its internal coin or the response from the oracle. We consider only nonce-respecting privacy adversaries.

Authenticity Notion. The authenticity notion we consider captures the unforgeability of a nonce-respecting adversary with a chosen ciphertext attack [Rog02]. Let $\mathcal{A}$ be an authenticity adversary that has access to the encryption oracle $\Pi.\mathrm{Enc}_K$ and the decryption oracle $\Pi.\mathrm{Dec}_K$. The encryption oracle is as in the privacy notion. The decryption oracle takes $(N, A, C, T) \in \mathcal{N}_\Pi \times \mathcal{A}_\Pi \times \mathcal{C}_\Pi \times \mathcal{T}_\Pi$ as input and returns $M \leftarrow \Pi.\mathrm{Dec}_K(N, A, C, T)$ or $\perp \leftarrow \Pi.\mathrm{Dec}_K(N, A, C, T)$. We define the authenticity advantage as

$$\mathbf{Adv}_{\Pi}^{\mathrm{auth}}(\mathcal{A}) \overset{\mathrm{def}}{=} \Pr\left[\mathcal{A}^{\Pi.\mathrm{Enc}_K(\cdot,\cdot,\cdot),\Pi.\mathrm{Dec}_K(\cdot,\cdot,\cdot,\cdot)} \text{ forges}\right] ,$$

where the probability is over $K \xleftarrow{\$} \mathcal{K}_\Pi$ and $\mathcal{A}$, and we say that the adversary forges if the decryption oracle returns a bit string (other than $\perp$) for a query (N, A, C, T), where (C, T) was not previously returned to $\mathcal{A}$ from the encryption oracle for a query (N, A, M). The adversary $\mathcal{A}$ in the authenticity game is assumed to be nonce-respecting with respect to the encryption oracle, and $\mathcal{A}$ can repeat the same nonce for the decryption oracle. More precisely, suppose that $\mathcal{A}$ has made i encryption queries with nonces $N_1, \dots, N_i$ and j decryption queries with nonces $N'_1, \dots, N'_j$. If the next query is an encryption query with a nonce N_{i+1}, then it holds that $N_{i+1} \notin \{N_1, \dots, N_i\}$, but $N_{i+1} \in \{N'_1, \dots, N'_j\}$ may hold. If the next query is a decryption query with a nonce N'_{j+1}, then both $N'_{j+1} \in \{N_1, \dots, N_i\}$ and $N'_{j+1} \in \{N'_1, \dots, N'_j\}$ may hold. Without loss of generality, we assume that $\mathcal{A}$ does not make redundant queries, i.e., if the encryption oracle returns (C, T) for a query (N, A, M), then $\mathcal{A}$ does not subsequently make a decryption query (N, A, C, T), and $\mathcal{A}$ does not repeat a query.

3 iZOCB and iZOTR

3.1 iZOCB

Overview. We first present the specification of iZOCB, the idealized version of ZOCB, to highlight the feasibility of the block-by-block processing of AD and AD-independent tag approach, and the structural difference from ΘCB3.[5]

Intuitively, iZOCB corresponds to the rightmost figure of ZOCB in Fig. 1, where the dashed box is replaced with an ideally secure TBC $\widetilde{\mathsf{E}}$ with large tweak space $\mathcal{G}$. We call $\mathcal{G}$ the global tweak space, and a global tweak includes the i-th t-bit AD block $A[i]$, a block counter i, a nonce N, and values for the domain separation. For a plaintext parsed into

[5] See Appendix D for an informal introduction to iZOCB and ZOCB.

n-bit blocks $(M[1],\ldots,M[m]) \xleftarrow{n} M$ and AD parsed into t-bit blocks $(A[1],\ldots,A[a]) \xleftarrow{t} A$, the i-th call of the TBC $\widetilde{\mathsf{E}}$ to compute a ciphertext block $C[i]$ can be described as $C[i] \leftarrow \widetilde{\mathsf{E}}((N, A[i], i), M[i])$. This allows us to encrypt an mn-bit plaintext with at-bit AD for any $a < m$ at the cost of $m+1$ calls of $\widetilde{\mathsf{E}}$. When $a \geq m$, $(a-m)t$ bits of AD remain unprocessed, and they are authenticated with a TBC-based (computational) universal hash function that takes $(n+t)$-bit input per $\widetilde{\mathsf{E}}$ call. This is an n-bit output version of $\mathbb{ZHASH}$ [IMPS17]. A tag is the encryption of a checksum, which is the XOR of all the plaintext blocks and is independent of the first $(m-1)t$ bits of AD, where m is the block length of the plaintext.

Compared to ΘCB3 that has an independent PMAC-style process for AD and hence tags depend on the entire AD, the process for $A[i]$ is integrated into the encryption process of a plaintext block $M[i]$. Therefore, iZOCB is structurally different from ΘCB3, and this is made possible since we are assuming global tweak space $\mathcal{G}$ for the underlying TBC.

Idealized ZOCB. We write $\mathsf{iZOCB}[\mathrm{Perm}(\mathcal{G}, n)] = (\mathsf{iZOCB.Enc}, \mathsf{iZOCB.Dec})$ for the encryption and decryption algorithms of iZOCB that uses $\mathrm{Perm}(\mathcal{G}, n)$ as the underlying TBC. $\mathcal{G}$ is global tweak space defined as

$$\mathcal{G} \stackrel{\text{def}}{=} \{\mathsf{C}, \mathsf{H}\} \times \mathcal{I}' \times \{0,1\}^n \times \{0,1\}^t \times \mathbb{Z}_\rho\,, \tag{1}$$

where C and H are used to separate the Core functions and the Hash function, $\mathcal{I}' \stackrel{\text{def}}{=} \{0,1,2\}$, $\mathbb{Z}_\rho = \{0,\ldots,\rho-1\}$, $\rho = 2^{(n+\min\{n,t\})/2} - 1$, and $t \geq 1$ is a constant, with the constraint that for a global tweak $(\mathsf{b}, \nu, N, B, i) \in \mathcal{G}$, N must satisfy $N = 0^n$ when $\mathsf{b} = \mathsf{H}$. In other words, $\mathcal{G}$ can alternatively be defined as

$$\mathcal{G} \stackrel{\text{def}}{=} \left(\{\mathsf{C}\} \times \mathcal{I}' \times \{0,1\}^n \times \{0,1\}^t \times \mathbb{Z}_\rho\right) \cup \left(\{\mathsf{H}\} \times \mathcal{I}' \times \{0^n\} \times \{0,1\}^t \times \mathbb{Z}_\rho\right)\,.$$

Here, $\{\mathsf{C}, \mathsf{H}\} \times \mathcal{I}'$ corresponds to the domain separation, $\{0,1\}^n$ corresponds to the space for nonces, $\{0,1\}^t$ is the space for t-bit AD blocks, and $\mathbb{Z}_\rho$ corresponds to the block counter. The value of ρ specifies the maximum input length.

Now iZOCB takes a TURP $\widetilde{\mathsf{E}} \in \mathrm{Perm}(\mathcal{G}, n)$ as a key. Let $\widetilde{\mathsf{D}}$ be the decryption of $\widetilde{\mathsf{E}}$. iZOCB has the signature $\mathcal{K}_{\mathsf{iZOCB}} = \mathrm{Perm}(\mathcal{G}, n)$, $\mathcal{N}_{\mathsf{iZOCB}} = \{0,1\}^n$, $\mathcal{A}_{\mathsf{iZOCB}} = \mathcal{M}_{\mathsf{iZOCB}} = \mathcal{C}_{\mathsf{iZOCB}} = \{0,1\}^{\leq n\rho}$, and $\mathcal{T}_{\mathsf{iZOCB}} = \{0,1\}^n$. The specification of $\mathsf{iZOCB}[\mathrm{Perm}(\mathcal{G}, n)] = (\mathsf{iZOCB.Enc}, \mathsf{iZOCB.Dec})$ is presented in Fig. 3 and the encryption algorithm is illustrated in Figs. 4 and 5.

In the encryption algorithm, $\mathsf{iZOCB.Enc}_{\widetilde{\mathsf{E}}}(N, A, M)$, we first check the block length m of M (in n-bit blocks), and if the length of A is less than mt bits, then we let $B \leftarrow \mathsf{ozp}_{mt}(A)$, $(C, Y) \leftarrow \mathsf{iCore.Enc}_{\widetilde{\mathsf{E}}}(N, B, M)$, and return (C, T) with $T = Y$. The Core function is the main routine for processing M and B to compute C and Y. M is processed as in ΘCB3, while we use the tweak input to process B. If $|A| \geq mt$, then we let B be the first mt bits of A, and $\widehat{B}$ be the remaining bits of A. M and B are processed as $(C, Y) \leftarrow \mathsf{iCore.Enc}_{\widetilde{\mathsf{E}}}(N, B, M)$, and for the remaining bits $\widehat{B}$ of A, they are processed with the Hash function as $\widehat{Y} \leftarrow \mathsf{iHash}_{\widetilde{\mathsf{E}}}(\widehat{B})$. The hash function works similarly to $\mathbb{ZHASH}$ of ZMAC. The output is (C, T) with $T = Y \oplus \widehat{Y}$.

Observe that, in Fig. 4, Y is independent of $B[1],\ldots,B[m-1]$, the first $(m-1)t$ bits of AD. However, when we decrypt, changes in $B[1],\ldots,B[m-1]$ will affect Y, which intuitively explains why our AD-independent tag approach works.

Security of iZOCB. The next theorem shows that iZOCB is provably secure.

Theorem 1. *Let* $\mathrm{Perm}(\mathcal{G}, n)$ *be the underlying TBC of* iZOCB*, where* $\mathcal{G}$ *is defined as in* (1)*. For any privacy adversary* $\mathcal{A}$ *that makes* q *encryption queries, we have*

```
**Algorithm** $\mathsf{iZOCB.Enc}_{\widetilde{\mathsf{E}}}(N, A, M)$

$m \leftarrow |M|_n$
**if** $|A| < mt$ **then**
    $B \leftarrow \mathsf{ozp}_{mt}(A)$
    $\widehat{Y} \leftarrow 0^n$
**else**
    $(B, \widehat{B}) \xleftarrow{mt,*} A$
    $\widehat{Y} \leftarrow \mathsf{iHash}_{\widetilde{\mathsf{E}}}(\widehat{B})$
$(C, Y) \leftarrow \mathsf{iCore.Enc}_{\widetilde{\mathsf{E}}}(N, B, M)$
$T \leftarrow \widehat{Y} \oplus Y$
**return** $(C, T)$
```

```
**Algorithm** $\mathsf{iCore.Enc}_{\widetilde{\mathsf{E}}}(N, B, M)$
// $M \in \{0,1\}^*$, $|M|_n = m$, and $|B| = mt$

$S \leftarrow 0^n$
$(M[1], \dots, M[m]) \xleftarrow{n} M$
$(B[1], \dots, B[m]) \xleftarrow{t} B$
**for** $i = 1$ **to** $m - 1$ **do**    // $m \geq 2$
    $S \leftarrow S \oplus M[i]$
    $C[i] \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},0,N,B[i],i-1}(M[i])$
$Z \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},0,N,0^t,m-1}(0^n)$
$C[m] \leftarrow M[m] \oplus \mathsf{msb}_{|M[m]|}(Z)$
$S \leftarrow S \oplus \mathsf{ozp}_n(M[m])$
**if** $|M[m]| \neq n$ **then**
    $Y \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},1,N,B[m],m-1}(S)$
**else**
    $Y \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},2,N,B[m],m-1}(S)$
$C \leftarrow (C[1], \dots, C[m])$
**return** $(C, Y)$
```

```
**Algorithm** $\mathsf{iHash}_{\widetilde{\mathsf{E}}}(\widehat{B})$

$\widehat{Y} \leftarrow 0^n$
$(\widehat{B}[1], \dots, \widehat{B}[a]) \xleftarrow{n+t} \widehat{B}$
**for** $i = 1$ **to** $a - 1$ **do**    // $a \geq 2$
    $(P[i], Q[i]) \xleftarrow{n,t} \widehat{B}[i]$
    $\widehat{Y} \leftarrow \widehat{Y} \oplus \widetilde{\mathsf{E}}^{\mathsf{H},0,0^n,Q[i],i-1}(P[i])$
$(P[a], Q[a]) \xleftarrow{n,t} \mathsf{ozp}_{n+t}(\widehat{B}[a])$
**if** $|\widehat{B}[a]| \neq n + t$ **then**
    $\widehat{Y} \leftarrow \widehat{Y} \oplus \widetilde{\mathsf{E}}^{\mathsf{H},1,0^n,Q[a],a-1}(P[a])$
**else**
    $\widehat{Y} \leftarrow \widehat{Y} \oplus \widetilde{\mathsf{E}}^{\mathsf{H},2,0^n,Q[a],a-1}(P[a])$
**return** $\widehat{Y}$
```

```
**Algorithm** $\mathsf{iZOCB.Dec}_{\widetilde{\mathsf{E}}}(N, A, C, T)$

$m \leftarrow |C|_n$    ◁
**if** $|A| < mt$ **then**
    $B \leftarrow \mathsf{ozp}_{mt}(A)$
    $\widehat{Y} \leftarrow 0^n$
**else**
    $(B, \widehat{B}) \xleftarrow{mt,*} A$
    $\widehat{Y} \leftarrow \mathsf{iHash}_{\widetilde{\mathsf{E}}}(\widehat{B})$
$(M, Y) \leftarrow \mathsf{iCore.Dec}_{\widetilde{\mathsf{E}}}(N, B, C)$    ◁
$T^* \leftarrow \widehat{Y} \oplus Y$    ◁
**if** $T^* = T$ **then return** $M$    ◁
**else return** $\perp$    ◁
```

```
**Algorithm** $\mathsf{iCore.Dec}_{\widetilde{\mathsf{E}}}(N, B, C)$
// $C \in \{0,1\}^*$, $|C|_n = m$, and $|B| = mt$

$S \leftarrow 0^n$
$(C[1], \dots, C[m]) \xleftarrow{n} C$    ◁
$(B[1], \dots, B[m]) \xleftarrow{t} B$
**for** $i = 1$ **to** $m - 1$ **do**    // $m \geq 2$
    $M[i] \leftarrow \widetilde{\mathsf{D}}^{\mathsf{C},0,N,B[i],i-1}(C[i])$    ◁
    $S \leftarrow S \oplus M[i]$    ◁
$Z \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},0,N,0^t,m-1}(0^n)$
$M[m] \leftarrow C[m] \oplus \mathsf{msb}_{|C[m]|}(Z)$    ◁
$S \leftarrow S \oplus \mathsf{ozp}_n(M[m])$
**if** $|C[m]| \neq n$ **then**    ◁
    $Y \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},1,N,B[m],m-1}(S)$
**else**
    $Y \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},2,N,B[m],m-1}(S)$
$M \leftarrow (M[1], \dots, M[m])$    ◁
**return** $(M, Y)$    ◁
```

Figure 3: Definitions of $\mathsf{iZOCB.Enc}_{\widetilde{\mathsf{E}}}(N, A, M)$ and $\mathsf{iZOCB.Dec}_{\widetilde{\mathsf{E}}}(N, A, C, T)$. The blue lines with a blue triangle in $\mathsf{iZOCB.Dec}_{\widetilde{\mathsf{E}}}(N, A, C, T)$ and $\mathsf{iCore.Dec}_{\widetilde{\mathsf{E}}}(N, B, C)$ indicate the difference from the corresponding encryption algorithms (also in other figures).

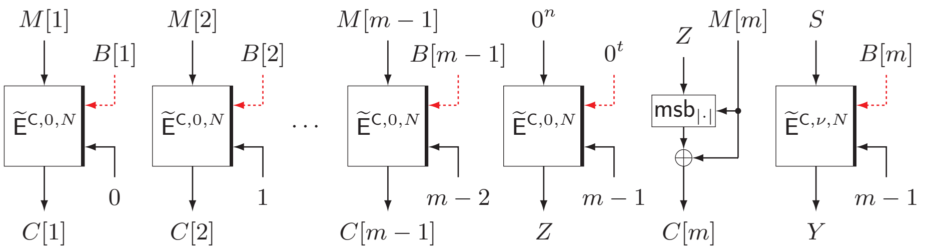

$M[1]$
$B[1]$
$\widetilde{\mathsf{E}}^{\mathsf{C},0,N}$
0
$C[1]$
$M[2]$
$B[2]$
$\widetilde{\mathsf{E}}^{\mathsf{C},0,N}$
1
$C[2]$
$\cdots$
$M[m-1]$
$B[m-1]$
$\widetilde{\mathsf{E}}^{\mathsf{C},0,N}$
$m-2$
$C[m-1]$
0^n
0^t
$\widetilde{\mathsf{E}}^{\mathsf{C},0,N}$
$m-1$
Z
Z
$M[m]$
$\mathsf{msb}_{|\cdot|}$
$C[m]$
S
$B[m]$
$\widetilde{\mathsf{E}}^{\mathsf{C},\nu,N}$
$m-1$
Y

Figure 4: $\mathsf{iCore.Enc}_{\widetilde{\mathsf{E}}}(N, B, M)$ of iZOCB, where $(B[1], \dots, B[m]) \xleftarrow{t} B$. The value of ν in the last step is 1 if $|M[m]| \neq n$ and 2 otherwise. S is $M[1] \oplus \cdots \oplus M[m-1] \oplus \mathsf{ozp}_n(M[m])$. The red dashed lines are t bits wide (also in other figures).

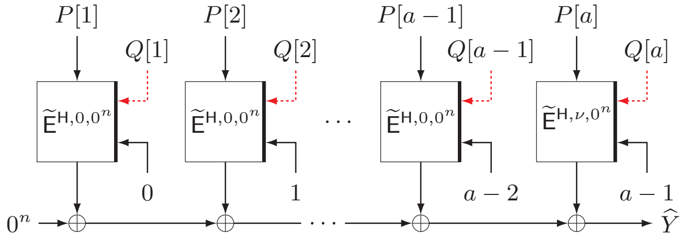

$P[1]$
$Q[1]$
$\widetilde{\mathsf{E}}^{\mathsf{H},0,0^n}$
0
$P[2]$
$Q[2]$
$\widetilde{\mathsf{E}}^{\mathsf{H},0,0^n}$
1
$\cdots$
$P[a-1]$
$Q[a-1]$
$\widetilde{\mathsf{E}}^{\mathsf{H},0,0^n}$
$a-2$
$P[a]$
$Q[a]$
$\widetilde{\mathsf{E}}^{\mathsf{H},\nu,0^n}$
$a-1$
0^n
$\cdots$
$\widehat{Y}$

Figure 5: $\mathsf{iHash}_{\widetilde{\mathsf{E}}}(\widehat{B})$ of iZOCB, where $(\widehat{B}[1], \dots, \widehat{B}[a]) \xleftarrow{n+t} \widehat{B}$ and $(P[i], Q[i]) \xleftarrow{n,t} \widehat{B}[i]$. ν in the last step is 1 if $|\widehat{B}[a]| \neq n+t$ and 2 otherwise.

$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{iZOCB}[\mathrm{Perm}(\mathcal{G},n)]}(\mathcal{A}) = 0$. *For any authenticity adversary* $\mathcal{A}$ *that makes* q *encryption and* q' *decryption queries, we have* $\mathbf{Adv}^{\mathrm{auth}}_{\mathsf{iZOCB}[\mathrm{Perm}(\mathcal{G},n)]}(\mathcal{A}) \leq 4q'/2^n$*.*

The privacy bound follows from the fact that $\mathsf{iCore.Enc}_{\widetilde{\mathsf{E}}}(N, B, M)$ and $\mathsf{iHash}_{\widetilde{\mathsf{E}}}(\widehat{B})$ are independent due to domain separation, and in $\mathsf{iCore.Enc}_{\widetilde{\mathsf{E}}}(N, B, M)$, each call of $\widetilde{\mathsf{E}}^{\mathsf{C},\cdot,\cdot,\cdot,\cdot}(\cdot)$ takes distinct global tweaks due to the nonce. The authenticity bound requires careful case analyses, and a complete proof is presented in Sect. 4.1.

3.2 iZOTR

Overview. We next present the idealized version of ZOTR, which we write iZOTR. This is based on $\mathbb{OTR}$ [Min14], and the PMAC-style process for AD in $\mathbb{OTR}$ is integrated into the encryption process.[6] iZOCB and iZOTR share the same overall structure, and the difference from iZOCB is the definition of the Core functions, where we use a two-round Feistel permutation that uses a TBC as round functions. This eliminates the need of the decryption of the TBC.

Idealized ZOTR. We write $\mathsf{iZOTR}[\mathrm{Perm}(\mathcal{G}, n)] = (\mathsf{iZOTR.Enc}, \mathsf{iZOTR.Dec})$ for the encryption and decryption algorithms of iZOTR that uses $\mathrm{Perm}(\mathcal{G}, n)$ as the underlying TBC. Here, $\mathcal{G}$ is global tweak space defined as

$$\mathcal{G} \overset{\mathrm{def}}{=} \{\mathsf{C}, \mathsf{H}\} \times \mathcal{I}' \times \{0,1\}^n \times \{0,1\}^t \times \mathbb{Z}_\rho\,, \tag{2}$$

where $\mathcal{I}' \overset{\mathrm{def}}{=} \{0, 1, \dots, 5\}$, $\mathbb{Z}_\rho = \{0, \dots, \rho - 1\}$, $\rho = 2^{(n+\min\{n,t\})/2} - 1$, and $t \geq 1$ is a constant, with the constraint that a global tweak $(\mathsf{b}, \nu, N, B, i) \in \mathcal{G}$ must satisfy $N = 0^n$ when $\mathsf{b} = \mathsf{H}$.

[6] See Appendix E for an informal introduction to iZOTR and ZOTR.

```
Algorithm iZOTR.Enc$_{\widetilde{\mathsf{E}}}(N, A, M)$

 1. $m \leftarrow |M|_n$
 2. if $|A| < mt$ then
 3.     $B \leftarrow \mathsf{ozp}_{mt}(A)$
 4.     $\widehat{Y} \leftarrow 0^n$
 5. else
 6.     $(B, \widehat{B}) \xleftarrow{mt,*} A$
 7.     $\widehat{Y} \leftarrow \mathsf{iHash}_{\widetilde{\mathsf{E}}}(\widehat{B})$
 8. $(C, Y) \leftarrow \mathsf{iCore.Enc}_{\widetilde{\mathsf{E}}}(N, B, M)$
 9. $T \leftarrow \widehat{Y} \oplus Y$
10. return $(C, T)$
```

```
Algorithm iZOTR.Dec$_{\widetilde{\mathsf{E}}}(N, A, C, T)$

 1. $m \leftarrow |C|_n$                                        ◁
 2. if $|A| < mt$ then
 3.     $B \leftarrow \mathsf{ozp}_{mt}(A)$
 4.     $\widehat{Y} \leftarrow 0^n$
 5. else
 6.     $(B, \widehat{B}) \xleftarrow{mt,*} A$
 7.     $\widehat{Y} \leftarrow \mathsf{iHash}_{\widetilde{\mathsf{E}}}(\widehat{B})$
 8. $(M, Y) \leftarrow \mathsf{iCore.Dec}_{\widetilde{\mathsf{E}}}(N, B, C)$   ◁
 9. $T^* \leftarrow \widehat{Y} \oplus Y$                              ◁
10. if $T^* = T$ then return $M$                            ◁
11. else return $\perp$                                       ◁
```

Figure 6: Definitions of $\mathsf{iZOTR.Enc}_{\widetilde{\mathsf{E}}}(N, A, M)$ and $\mathsf{iZOTR.Dec}_{\widetilde{\mathsf{E}}}(N, A, M)$

iZOTR uses $\mathrm{Perm}(\mathcal{G}, n)$ as the underlying TBC, and it takes a TURP $\widetilde{\mathsf{E}} \in \mathrm{Perm}(\mathcal{G}, n)$ as a key. It has the signature $\mathcal{K}_{\mathsf{iZOTR}} = \mathrm{Perm}(\mathcal{G}, n)$, $\mathcal{N}_{\mathsf{iZOTR}} = \{0,1\}^n$, $\mathcal{A}_{\mathsf{iZOTR}} = \mathcal{M}_{\mathsf{iZOTR}} = \mathcal{C}_{\mathsf{iZOTR}} = \{0,1\}^{\le n\rho}$, and $\mathcal{T}_{\mathsf{iZOTR}} = \{0,1\}^n$.

The specification of $\mathsf{iZOTR}[\mathrm{Perm}(\mathcal{G}, n)] = (\mathsf{iZOTR.Enc}, \mathsf{iZOTR.Dec})$ is in Figs. 6, 7, 8, and 9, and the encryption is illustrated in Figs. 10, 11, and 12. We note that the pseudocode of $\mathsf{iZOTR.Enc}_{\widetilde{\mathsf{E}}}(N, A, M)$, $\mathsf{iZOTR.Dec}_{\widetilde{\mathsf{E}}}(N, A, C, T)$, and $\mathsf{iHash}_{\widetilde{\mathsf{E}}}(\widehat{B})$ is the same as the corresponding pseudocode of iZOCB. iZOCB and iZOTR differ only in $\mathsf{iCore.Enc}_{\widetilde{\mathsf{E}}}(N, B, M)$ and $\mathsf{iCore.Dec}_{\widetilde{\mathsf{E}}}(N, B, M)$, where the Core functions work as in $\mathbb{OTR}$, while AD is processed with the tweak input of the underlying TBC.

Security of iZOTR. We show that iZOTR is provably secure. A proof is in Sect. 4.2.

Theorem 2. *Let $\mathrm{Perm}(\mathcal{G}, n)$ be the underlying TBC of* iZOTR*, where $\mathcal{G}$ is defined as in (2). For any privacy adversary $\mathcal{A}$ that makes q encryption queries, we have $\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{iZOTR}[\mathrm{Perm}(\mathcal{G},n)]}(\mathcal{A}) = 0$. For any authenticity adversary $\mathcal{A}$ that makes q encryption and q' decryption queries, we have $\mathbf{Adv}^{\mathrm{auth}}_{\mathsf{iZOTR}[\mathrm{Perm}(\mathcal{G},n)]}(\mathcal{A}) \le 6q'/2^n$.*

4 Security Proofs of iZOCB and iZOTR

4.1 Security Proof of iZOCB (Theorem 1)

The overall proof approach follows those of OCB1 and ΘCB3. However, we need an additional analysis due to the integration mechanism of the AD process.

We first consider the privacy bound, and then the authenticity bound.

The Privacy Bound. We observe that, for an encryption query (N, A, M), any output block of iCore.Enc involves exactly one call of $\widetilde{\mathsf{E}}$ taking a unique tweak that contains N. Therefore, (C, Y) in line 8 of iZOCB.Enc is a uniformly random string of $|M| + n$ bits. The tag T in line 9 is also uniformly random since iHash is an independent procedure to iCore.Enc from the domain separation. This means that (C, T) is completely random and the privacy bound is 0.

The Authenticity Bound. We first consider the case $q' = 1$ (i.e. single decryption query). We consider the most effective information theoretic adversary, i.e., we assume that the adversary makes all the q encryption queries before making the single verification query.

```
**Algorithm** $\mathsf{iCore.Enc}_{\widetilde{\mathsf{E}}}(N, B, M)$        // $M \in \{0,1\}^*$, $|M|_n = m$, and $|B| = mt$
    $S \leftarrow 0^n$
    $(M[1], \ldots, M[m]) \xleftarrow{n} M$
    $\ell \leftarrow \lceil m/2 \rceil$
    $(B[1], \ldots, B[m]) \xleftarrow{t} B$
    **for** $i = 1$ **to** $\ell - 1$ **do**                                     // $\ell \geq 2$
        $C[2i-1] \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},0,N,B[2i-1],i-1}(M[2i-1]) \oplus M[2i]$
        $C[2i] \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},1,N,B[2i],i-1}(C[2i-1]) \oplus M[2i-1]$
        $S \leftarrow S \oplus M[2i]$
    **if** $m \bmod 2 = 0$ **then**
        $Z \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},0,N,B[m-1],\ell-1}(M[m-1])$
        $C[m] \leftarrow \mathsf{msb}_{|M[m]|}(Z) \oplus M[m]$
        $C[m-1] \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},1,N,0^t,\ell-1}(\mathsf{ozp}_n(C[m])) \oplus M[m-1]$
        $S \leftarrow S \oplus \mathsf{ozp}_n(C[m]) \oplus Z$
    **if** $m \bmod 2 = 1$ **then**
        $C[m] \leftarrow \mathsf{msb}_{|M[m]|}(\widetilde{\mathsf{E}}^{\mathsf{C},0,N,0^t,\ell-1}(0^n)) \oplus M[m]$
        $S \leftarrow S \oplus \mathsf{ozp}_n(M[m])$
    **if** $m \bmod 2 = 0$ **and** $|M[m]| \neq n$ **then** $Y \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},2,N,B[m],\ell-1}(S)$
    **if** $m \bmod 2 = 0$ **and** $|M[m]| = n$ **then** $Y \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},3,N,B[m],\ell-1}(S)$
    **if** $m \bmod 2 = 1$ **and** $|M[m]| \neq n$ **then** $Y \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},4,N,B[m],\ell-1}(S)$
    **if** $m \bmod 2 = 1$ **and** $|M[m]| = n$ **then** $Y \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},5,N,B[m],\ell-1}(S)$
    $C \leftarrow (C[1], \ldots, C[m])$
    **return** $(C, Y)$
```

Figure 7: Definition of $\mathsf{iCore.Enc}_{\widetilde{\mathsf{E}}}(N, B, M)$ of iZOTR

Let $\mathcal{S}_e = (N_i, A_i, M_i, C_i, T_i)_{i=1,\ldots,q}$ be the query-response pairs of the q encryption queries, and let (N', A', C', T') be the single decryption query. Note that the adversary can be assumed to be deterministic, and this implies that (N', A', C', T') is a function of $\mathcal{S}_e$.

Now from the definition, we have $N_i \neq N_j$ for all $1 \leq i < j \leq q$ and $(N', A', C', T') \neq (N_i, A_i, C_i, T_i)$ for all $1 \leq i \leq q$. We follow the notation used in the corresponding pseudocode. All the internal variables for the i-th encryption query have subscript i, and those for the decryption query have a prime symbol. For example, M_i is parsed as $(M_i[1], \ldots, M_i[m_i]) \xleftarrow{n} M_i$. Let $T^* \in \{0,1\}^n$ be the true tag value for the decryption query (N', A', C', T'), i.e., T^* is the value defined in line 9 of the computation of $\mathsf{iZOCB.Dec}_{\widetilde{\mathsf{E}}}(N', A', C', T')$ in Fig. 3.

We define $\nu' \in \{1, 2\}$ as $\nu' = 1$ when $|C'[m']| \neq n$, and $\nu' = 2$ when $|C'[m']| = n$. This is used in the encryption of S'. For the i-th encryption query, ν_i is defined similarly.

For a random variable X over finite space $\mathcal{X}$, we write

$$\mathrm{pp}(X) \overset{\mathrm{def}}{=} \max_{x \in \mathcal{X}} \Pr[X = x]$$

to denote the maximum point probability of X. The maximum point probability conditioned on an event $\mathcal{E}$ is defined as $\mathrm{pp}(X \mid \mathcal{E}) = \max_{x \in \mathcal{X}} \Pr[X = x \mid \mathcal{E}]$.

Let $p \overset{\mathrm{def}}{=} \Pr[T' = T^* \mid \mathcal{S}_e]$. We evaluate p, which directly gives the maximum of forgery probability, by a case analysis. As we arbitrarily fix $\mathcal{S}_e$, (N', A', C', T') is also fixed, and we rely on the remaining randomness of $\mathsf{iZOCB.Dec}_{\widetilde{\mathsf{E}}}$.

As a decryption query (N', A', C', T') such that $(N', A', C') = (N_i, A_i, C_i)$ for some $1 \leq i \leq q$ and $T' \neq T_i$ always fails, we may assume that $(N', A', C') \neq (N_i, A_i, C_i)$ holds

```
Algorithm iCore.Dec_Ẽ(N, B, C)        // $C \in \{0,1\}^*$, $|C|_n = m$, and $|B| = mt$
    $S \leftarrow 0^n$
    $(C[1], \ldots, C[m]) \xleftarrow{n} C$                                   ◁
    $\ell \leftarrow \lceil m/2 \rceil$
    $(B[1], \ldots, B[m]) \xleftarrow{t} B$
    for $i = 1$ to $\ell - 1$ do                                                // $\ell \geq 2$
        $M[2i-1] \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},1,N,B[2i],i-1}(C[2i-1]) \oplus C[2i]$             ◁
        $M[2i] \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},0,N,B[2i-1],i-1}(M[2i-1]) \oplus C[2i-1]$           ◁
        $S \leftarrow S \oplus M[2i]$
    if $m$ mod $2 = 0$ then
        $M[m-1] \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},1,N,0^t,\ell-1}(\mathsf{ozp}_n(C[m])) \oplus C[m-1]$  ◁
        $Z \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},0,N,B[m-1],\ell-1}(M[m-1])$                          ◁
        $M[m] \leftarrow \mathsf{msb}_{|C[m]|}(Z) \oplus C[m]$                             ◁
        $S \leftarrow S \oplus \mathsf{ozp}_n(C[m]) \oplus Z$
    if $m$ mod $2 = 1$ then
        $M[m] \leftarrow \mathsf{msb}_{|C[m]|}(\widetilde{\mathsf{E}}^{\mathsf{C},0,N,0^t,\ell-1}(0^n)) \oplus C[m]$   ◁
        $S \leftarrow S \oplus \mathsf{ozp}_n(M[m])$
    if $m$ mod $2 = 0$ and $|C[m]| \neq n$ then $Y \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},2,N,B[m],\ell-1}(S)$  ◁
    if $m$ mod $2 = 0$ and $|C[m]| = n$ then $Y \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},3,N,B[m],\ell-1}(S)$   ◁
    if $m$ mod $2 = 1$ and $|C[m]| \neq n$ then $Y \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},4,N,B[m],\ell-1}(S)$  ◁
    if $m$ mod $2 = 1$ and $|C[m]| = n$ then $Y \leftarrow \widetilde{\mathsf{E}}^{\mathsf{C},5,N,B[m],\ell-1}(S)$   ◁
    $M \leftarrow (M[1], \ldots, M[m])$                                         ◁
    return $(M, Y)$                                                         ◁
```

Figure 8: Definition of iCore.Dec$_{\widetilde{\mathsf{E}}}(N, B, C)$ of iZOTR

```
Algorithm iHash_Ẽ($\widehat{B}$)
    $Y \leftarrow 0^n$
    $(\widehat{B}[1], \ldots, \widehat{B}[a]) \xleftarrow{n+t} \widehat{B}$
    for $i = 1$ to $a - 1$ do          // $a \geq 2$
        $(P[i], Q[i]) \xleftarrow{n,t} \widehat{B}[i]$
        $Y \leftarrow Y \oplus \widetilde{\mathsf{E}}^{\mathsf{H},0,0^n,Q[i],i-1}(P[i])$
    $(P[a], Q[a]) \xleftarrow{n,t} \mathsf{ozp}_{n+t}(\widehat{B}[a])$
    if $|\widehat{B}[a]| \neq n + t$ then
        $Y \leftarrow Y \oplus \widetilde{\mathsf{E}}^{\mathsf{H},1,0^n,Q[a],a-1}(P[a])$
    else
        $Y \leftarrow Y \oplus \widetilde{\mathsf{E}}^{\mathsf{H},2,0^n,Q[a],a-1}(P[a])$
    return $Y$
```

Figure 9: Definition of iHash$_{\widetilde{\mathsf{E}}}(\widehat{B})$ of iZOTR

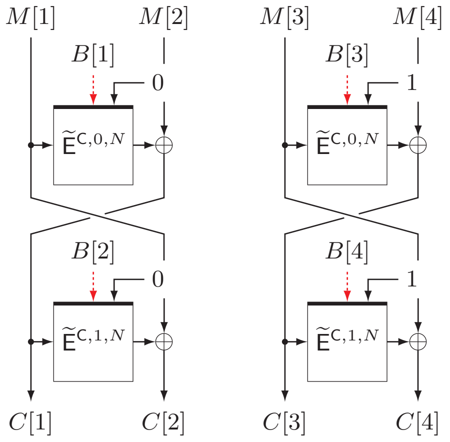

$M[1]$
$M[2]$
$B[1]$
0
$\widetilde{\mathsf{E}}^{\mathsf{C},0,N}$
$B[2]$
0
$\widetilde{\mathsf{E}}^{\mathsf{C},1,N}$
$C[1]$
$C[2]$
$M[3]$
$M[4]$
$B[3]$
1
$\widetilde{\mathsf{E}}^{\mathsf{C},0,N}$
$B[4]$
1
$\widetilde{\mathsf{E}}^{\mathsf{C},1,N}$
$C[3]$
$C[4]$

Figure 10: iCore.Enc$_{\widetilde{\mathsf{E}}}(N, B, M)$ of iZOTR for the process of $M[1], \ldots, M[4]$ and $B[1], \ldots, B[4]$ when $|M|_n = m \geq 5$

for all $1 \leq i \leq q$. Recall that $m' = |C'|_n$ and $m_i = |C_i|_n$. Our analysis is divided into the following cases.

Case 1. $N' \neq N_i$ for all $1 \leq i \leq q$

Case 2. For some $1 \leq i \leq q$, $N' = N_i$ and $C' \neq C_i$

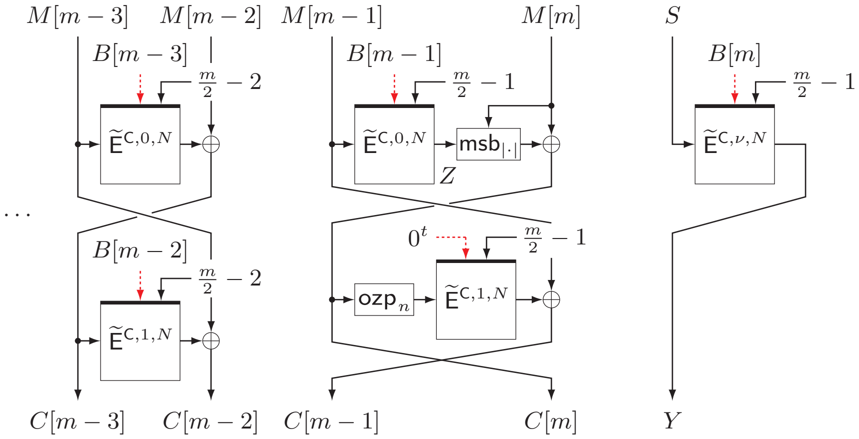


Figure 11: $\mathsf{iCore.Enc}_{\widetilde{\mathsf{E}}}(N, B, M)$ of iZOTR for the case $m \bmod 2 = 0$. This illustrates the process of $M[m-3], \ldots, M[m]$ and $B[m-3], \ldots, B[m]$, where $(B[1], \ldots, B[m]) \xleftarrow{t} B$. ν in the last step is 2 if $|M[m]| \neq n$ and 3 otherwise. S is $M[2] \oplus M[4] \oplus \cdots \oplus M[m-2] \oplus \mathsf{ozp}_n(C[m]) \oplus Z$.

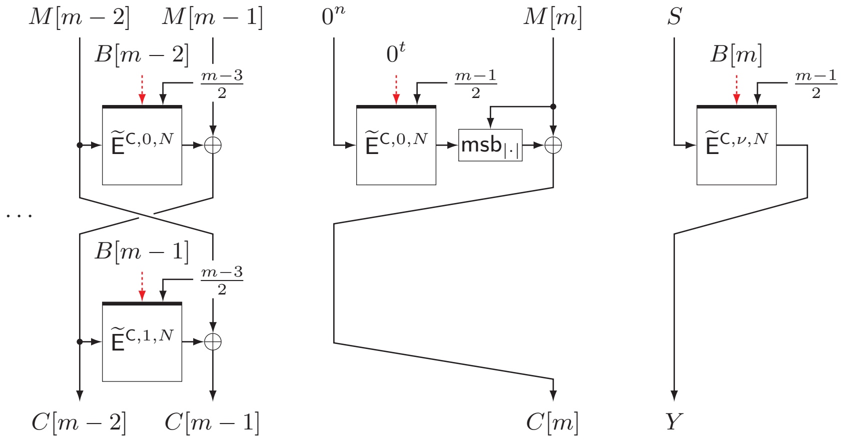


Figure 12: $\mathsf{iCore.Enc}_{\widetilde{\mathsf{E}}}(N, B, M)$ of iZOTR for the case $m \bmod 2 = 1$. This processes $M[m-2], \ldots, M[m]$ and $B[m-2], \ldots, B[m]$, where $(B[1] \ldots, B[m]) \xleftarrow{t} B$. ν in the last step is 4 if $|M[m]| \neq n$ and 5 otherwise. S is $M[2] \oplus M[4] \oplus \cdots \oplus M[m-1] \oplus \mathsf{ozp}_n(M[m])$.

Case 2.1. $m' \neq m_i$

Case 2.2. $m' = m_i$

Case 3. For some $1 \leq i \leq q$, $N' = N_i$, $C' = C_i$, and $A' \neq A_i$

Case 3.1. $B' \neq B_i$

Case 3.2. $B' = B_i$ and $|A'|, |A_i| < m't$

Case 3.3. $B' = B_i$, $|A'| < m't$, and $|A_i| \geq m't$

Case 3.4. $B' = B_i$, $|A'| \geq m't$, and $|A_i| < m't$

Case 3.5. $B' = B_i$ and $|A'|, |A_i| \geq m't$

Observe that we cover all the cases. The details of the analysis follow.

Case 1. Y' is uniformly random over $\{0,1\}^n$, and from $T^* = Y' \oplus \widehat{Y}'$, we have $p = 1/2^n$.

Case 2.1. $Y' = \widetilde{\mathsf{E}}^{\mathsf{C},\nu',N',B'[m'],m'-1}(S')$ (for some $\nu' \in \{1,2\}$) is uniformly random since $(\nu', N', B'[m'], m'-1)$ is a unique tweak among all calls to $\widetilde{\mathsf{E}}^{\mathsf{C},\cdot,\cdot,\cdot,\cdot}(\cdot)$ for encryption queries due to the uniqueness of $(\nu', m'-1)$. Therefore, $T^* = Y' \oplus \widehat{Y}'$ is also uniform and we have $p = 1/2^n$.

Case 2.2. This case is similar to the case of ΘCB3 and is divided into two subcases. For simplicity we write $m = m' = m_i$. Since $\widehat{Y}'$ and $\widehat{Y}_i$ are independent of Y' and Y_i, the probability of $T^* \oplus T_i = Y' \oplus Y_i \oplus \widehat{Y}' \oplus \widehat{Y}_i = 0^n$ is bounded by $\mathrm{pp}(Y' \oplus Y_i)$.

Subcase (1) is the case $C'[j] \neq C_i[j]$ for some $1 \leq j < m$ with $m \geq 2$. Then $M'[j]$ is uniform over $\{0,1\}^n$ (when $B'[j] \neq B_i[j]$) or $\{0,1\}^n \setminus \{M_i[j]\}$ (when $B'[j] = B_i[j]$). This implies that

$$S' \oplus S_i = \left(\bigoplus_{h=1,\ldots,m-1} M'[h] \oplus M_i[h] \right) \oplus \mathsf{ozp}_n(M'[m]) \oplus \mathsf{ozp}_n(M_i[m]) \tag{3}$$

contains an independent term $M'[j] \oplus M_i[j]$ which is uniform over $\{0,1\}^n$ or uniform over $\{0,1\}^n \setminus \{0^n\}$. Therefore, $\Pr[S' \oplus S_i = 0^n]$ is at most $1/(2^n - 1)$, and we have $\mathrm{pp}(Y' \oplus Y_i \mid S' \neq S_i) \leq 1/(2^n - 1)$. This yields $p \leq 1/(2^n - 1) + 1/(2^n - 1) = 2/(2^n - 1)$.

Subcase (2) is the case $C'[j] = C_i[j]$ for all $1 \leq j < m$ and $C'[m] \neq C_i[m]$. If $B'[j] \neq B_i[j]$ for some $1 \leq j < m$, then (3) contains an independent term $M'[j]$ which is uniform over $\{0,1\}^n$, and hence we have $\Pr[S' \oplus S_i = 0^n] = 1/2^n$, from which $p \leq 2/(2^n - 1)$ follows.

If $B'[j] = B_i[j]$ for all $1 \leq j < m$, then we have

$$S' \oplus S_i = \mathsf{ozp}_n(M'[m]) \oplus \mathsf{ozp}_n(M_i[m])\,.$$

First, when $\nu' \neq \nu_i$, the final tweaks for S' and S_i are different and thus $Y' \oplus Y_i$ is uniform. Next, consider the case $\nu' = \nu_i$. When $|C'[m]| \neq |C_i[m]|$ (which must be the case that both are shorter than n bits), we have $|M'[m]| \neq |M_i[m]|$ and $S' \oplus S_i \neq 0^n$ from the property of ozp. When $|C'[m]| = |C_i[m]| = \ell$ for some $1 \leq \ell \leq n$, we have

$$\begin{aligned} S' \oplus S_i &= \mathsf{ozp}_n(C'[m] \oplus \mathsf{msb}_\ell(Z)) \oplus \mathsf{ozp}_n(C_i[m] \oplus \mathsf{msb}_\ell(Z)) \\ &= \mathsf{ozp}_n(C'[m] \oplus C_i[m])\,, \end{aligned}$$

where $Z = Z' = Z_i$, and this must be non-zero. Thus, $\mathrm{pp}(Y' \oplus Y_i) \leq 1/(2^n - 1)$ holds, and this gives $p \leq 2/(2^n - 1)$ for Case 2.2.

Case 3.1. For some $1 \leq j \leq m$, where $m = m' = m_i$, we have $B'[j] \neq B_i[j]$. Therefore, when $1 \leq j < m$ (and $m \geq 2$), $M'[j]$ is uniform and independent of $M_i[j]$, thus $S' \oplus S_i = 0^n$ with probability $1/2^n$. When $j = m$, S' and S_i are encrypted with different tweaks, making Y' independent and uniform over $\{0,1\}^n$. From the same analysis as in Case 2.2 including the observation that we can focus on $\mathrm{pp}(Y' \oplus Y_i)$ and ignore $\widehat{Y}'$ and $\widehat{Y}_i$, we have $p \leq 2/(2^n - 1)$.

Case 3.2. Let $m = m' = m_i$. We have $B' = \mathsf{ozp}_{mt}(A')$ and $B_i = \mathsf{ozp}_{mt}(A_i)$, and $A' \neq A_i$ implies that this case is impossible.

Case 3.3. We have $T^* = Y' = Y_i$ and $T_i = Y_i \oplus \widehat{Y}_i$. Here, $\widehat{Y}_i$ is $\mathsf{iHash}_{\widetilde{\mathsf{E}}}(\widehat{B}_i)$ and is uniformly random since this involves at least one call of $\widetilde{\mathsf{E}}^{\mathsf{H},\cdot,\cdot,\cdot,\cdot}(\cdot)$ (note that $\widehat{B}_i$ can be empty), and we may have $\widehat{B}_i = \widehat{B}_j$ for some (possibly multiple) indices $1 \le j \le q$, in which case we have $\widehat{Y}_j = \widehat{Y}_i$ with probability one. Let $\mathcal{J}$ be the set of such indices $j \in \{1, \dots, q\} \setminus \{i\}$.

We observe that Y_i is uniform given $\mathcal{S}_e$, since $T_i = Y_i \oplus \widehat{Y}_i$ completely hides Y_i for the randomness of $\widehat{Y}_i$, and $\widehat{Y}_j$ for any $j \in \mathcal{J}$ is masked by Y_j, which itself is independent and uniformly random. Therefore, T^* is completely random, implying $p = 1/2^n$.

Case 3.4. We have $T^* = Y' \oplus \widehat{Y}' = Y_i \oplus \widehat{Y}'$ and $T_i = Y_i$. From the same analysis as in Case 3.3, $\widehat{Y}'$ is either independent and uniformly random (when $\widehat{B}' \ne \widehat{B}_j$ for all $1 \le j \le q$), or uniformly random but not independent (when $\widehat{B}' = \widehat{B}_j$ for some $j \in \{1, \dots, q\} \setminus \{i\}$). However, for the latter case, $\widehat{Y}' = \widehat{Y}_j$ is masked by Y_j. Hence $\widehat{Y}'$ is uniform given $\mathcal{S}_e$ and we have $p = 1/2^n$.

Case 3.5. The analysis is similar to that of authentication part of ΘCB3. Let $m = m' = m_i$. We have $T^* = Y' \oplus \widehat{Y}'$ and $T_i = Y_i \oplus \widehat{Y}_i$, where $Y' = Y_i$. As in the analysis of Case 3.4, $\widehat{Y}'$ may be identical to some $\widehat{Y}_j$ for $j \in \{1, \dots, q\} \setminus \{i\}$. However, we may ignore them and focus on $\mathrm{pp}(\widehat{Y}' \oplus \widehat{Y}_i)$.

Let $|\widehat{B}'|_{n+t} = a'$ and $|\widehat{B}_i|_{n+t} = a_i$. As the first mt bits of A' and A_i are the same, we must have $\widehat{B}' \ne \widehat{B}_i$. Following the pseudocode, let $(P'[j], Q'[j]) \xleftarrow{n,t} \widehat{B}'[j]$ for $1 \le j < a'$ and $(P'[a'], Q'[a']) \xleftarrow{n,t} \mathsf{ozp}_{n+t}(\widehat{B}'[a'])$. Similarly, we define $(P_i[j], Q_i[j])$, where $1 \le j \le a_i$, for $\widehat{B}_i$. We divide the analysis into three subcases.

Subcase (1) is the case $a' = a_i \ge 2$ and $\widehat{B}'[j] \ne \widehat{B}[j]$ for some $1 \le j < a'$. We observe that $\widehat{Y}' \oplus \widehat{Y}_i$ contains an independent term $\mathsf{E}^{\mathsf{H},0,0^n,Q'[j],j-1}(P'[j]) \oplus \mathsf{E}^{\mathsf{H},0,0^n,Q_i[j],j-1}(P_i[j])$, and this term has the maximum point probability of $1/(2^n - 1)$, which implies $\mathrm{pp}(\widehat{Y}' \oplus \widehat{Y}_i) \le 1/(2^n - 1)$. We therefore have $p \le 1/(2^n - 1)$.

Subcase (2) is the case $a' = a_i$ and $\widehat{B}'[a'] \ne \widehat{B}_i[a']$. In this case, we have either $\mathsf{ozp}_{n+t}(\widehat{B}'[a']) \ne \mathsf{ozp}_{n+t}(\widehat{B}_i[a'])$ or $\mathsf{ozp}_{n+t}(\widehat{B}'[a']) = \mathsf{ozp}_{n+t}(\widehat{B}_i[a'])$ but $|\widehat{B}'[a']| \ne |\widehat{B}_i[a']|$. The former case is the same as Subcase (1). For the latter case, due to the difference in the second argument of the tweak $\nu \in \{1, 2\}$, $\widehat{Y}' \oplus \widehat{Y}_i$ is uniformly random. Therefore, $p \le 1/(2^n - 1)$.

Subcase (3) is the case $a' < a_i$ or $a_i < a'$. If $a' < a_i$, we see that $\widehat{Y}' \oplus \widehat{Y}_i$ contains an independent term $\mathsf{E}^{\mathsf{H},0,0^n,Q_i[a_i],a_i-1}(P_i[a_i])$, and we have $p = 1/2^n$. The same analysis holds for the case $a_i < a'$.

Overall, $p \le 2/(2^n - 1)$ holds for all the cases, which gives

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathsf{iZOCB}[\mathrm{Perm}(\mathcal{G},n)]}(\mathcal{A}) \le \frac{2}{2^n - 1} \le \frac{4}{2^n}$$

for any adversary $\mathcal{A}$ that makes q encryption queries and single decryption query. Using the generic transformation from single to multiple decryption queries given by Bellare et al. [BGM04], we obtain the claimed authenticity bound.

4.2 Security Proof of iZOTR (Theorem 2)

The proof is similar to those of iZOCB and 𝕆𝕋ℝ [Min14][7]. We analyze the privacy first, and then the authenticity.

[7] 𝕆𝕋ℝ uses a tweakable random function, while iZOTR uses a tweakable random permutation, and this makes the difference of the constant in the security bound.

The Privacy Bound. We see that any output block of $\mathsf{iCore.Enc}$ has exactly one call of $\widetilde{\mathsf{E}}$ taking a unique tweak, and this contains a nonce. Therefore, (C, Y) is uniform, and from the independence of iHash, (C, T) is also uniform. This proves that the privacy bound is zero.

The Authenticity Bound. The basic strategy is the same as the case of iZOCB. We first focus on the case $q' = 1$. We employ the same notation as was used in the proof of Theorem 1. Additionally, we use $\ell' = \lceil m'/2 \rceil$ and $\nu' \in \{2, 3, 4, 5\}$ following the pseudocode, i.e., for the decryption query, we have $\nu' = 2$ when $m' \bmod 2 = 0$ and $|C'[m']| \neq n$, $\nu' = 3$ when $m' \bmod 2 = 0$ and $|C'[m']| = n$, $\nu' = 4$ when $m' \bmod 2 = 1$ and $|C'[m']| \neq n$, and $\nu' = 5$ when $m' \bmod 2 = 1$ and $|C'[m']| = n$. For the i-th encryption query, we define ℓ_i and ν_i analogously, and we also use subscript i for other internal variables. Let $p = \Pr[T' = T^* \mid \mathcal{S}_e]$ be the advantage of the most effective adversary, where $\mathcal{S}_e$ denote the transcript of the q encryption queries. Following the proof of Theorem 1, we use $\mathrm{pp}(X)$ to denote the maximum point probability of random variable X.

The case analysis for deriving p is as follows.

Case 1. $N' \neq N_i$ for all $1 \leq i \leq q$

Case 2. For some $1 \leq i \leq q$, $N' = N_i$ and $C' \neq C_i$

Case 2.1. $(\nu', \ell', B'[m']) \neq (\nu_i, \ell_i, B_i[m_i])$

Case 2.2. $(\nu', \ell', B'[m']) = (\nu_i, \ell_i, B_i[m_i])$ and $\nu' = \nu_i \in \{2, 3\}$

Case 2.3. $(\nu', \ell', B'[m']) = (\nu_i, \ell_i, B_i[m_i])$ and $\nu' = \nu_i \in \{4, 5\}$

Case 3. For some $1 \leq i \leq q$, $N' = N_i$, $C' = C_i$, and $A' \neq A_i$

Case 3.1. $(\nu', \ell', B'[m']) \neq (\nu_i, \ell_i, B_i[m_i])$

Case 3.2. $(\nu', \ell', B'[m']) = (\nu_i, \ell_i, B_i[m_i])$ and $B' \neq B_i$

Case 3.3. $(\nu', \ell', B'[m']) = (\nu_i, \ell_i, B_i[m_i])$, $B' = B_i$, and $|A'|, |A_i| < m't$

Case 3.4. $(\nu', \ell', B'[m']) = (\nu_i, \ell_i, B_i[m_i])$, $B' = B_i$, $|A'| < m't$, and $|A_i| \geq m't$

Case 3.5. $(\nu', \ell', B'[m']) = (\nu_i, \ell_i, B_i[m_i])$, $B' = B_i$, $|A'| \geq m't$, and $|A_i| < m't$

Case 3.6. $(\nu', \ell', B'[m']) = (\nu_i, \ell_i, B_i[m_i])$, $B' = B_i$ and $|A'|, |A_i| \geq m't$

In what follows, we present the details of the analysis.

Case 1. Y' is random, and from $T^* = Y' \oplus \widehat{Y}'$, we have $p = 1/2^n$.

Case 2.1. The final tweaks used to encrypt S' and S_i are different, and both are used exactly once. Therefore, Y' is independent and uniform, and we have $p = 1/2^n$.

Case 2.2. As $\nu' = \nu_i \in \{2, 3\}$, we have $m' \bmod 2 = m_i \bmod 2 = 0$. The analysis of this case mostly follows the proof of $\mathbb{OTR}$ [Min14]. From $(\nu', \ell') = (\nu_i, \ell_i)$, $m' = m_i$ holds true. For simplicity let $m = m' = m_i$ and $\ell = \ell' = \ell_i$. Note that by definition, m cannot be zero, and we have $m \geq 2$. Let $(V'[1], \ldots, V'[\ell]) \xleftarrow{2n} C'$ and $(V_i[1], \ldots, V_i[\ell]) \xleftarrow{2n} C_i$, and we call $V'[j]$ or $V_i[j]$ a chunk. From $C' \neq C_i$, there exists $1 \leq j \leq \ell$ such that $V'[j] \neq V_i[j]$.

As in the analysis of iZOCB, we evaluate $\mathrm{pp}(Y' \oplus Y_i)$ and this can be used as the bound of p from the independence of iHash. As in the proof of Theorem 1, we observe that

$$\begin{aligned}\mathrm{pp}(Y' \oplus Y_i) &\leq \mathrm{pp}(Y' \oplus Y_i \mid S' \neq S_i) + \Pr[S' = S_i] \\ &\leq \frac{1}{2^n - 1} + \Pr[S' = S_i]\,. \end{aligned} \tag{4}$$

We evaluate $\mathrm{pp}(Y' \oplus Y_i)$ in two subcases.

Subcase (1) is the case that j satisfies $1 \le j < \ell$. We have $|V'[j]| = |V_i[j]| = 2n$, and we note that $\ell \ge 2$ in this case. Let $(C'[2j-1], C'[2j]) \xleftarrow{n} V'[j]$, and the decryption of $V'[j]$ is described as

$$\begin{cases} M'[2j-1] = \widetilde{\mathsf{E}}^{\mathsf{C},1,N',B'[2j],j-1}(C'[2j-1]) \oplus C'[2j]\,, & (5)\\ M'[2j] = \widetilde{\mathsf{E}}^{\mathsf{C},0,N',B'[2j-1],j-1}(M'[2j-1]) \oplus C'[2j-1]\,, & (6)\end{cases}$$

and $M'[2j]$ is used in the checksum S'.

We observe that $S' \oplus S_i$ contains $M'[2j] \oplus M_i[2j]$, and this is independent of other n-bit terms in $S' \oplus S_i$, since they are encrypted with $\widetilde{\mathsf{E}}$ using different tweaks. Therefore, we have

$$\Pr[S' = S_i] \le \mathrm{pp}(M'[2j] \oplus M_i[2j])\,. \qquad (7)$$

From (5), we observe that $\Pr\big[M'[2j-1] = M_i[2j-1]\big] \le 1/(2^n-1)$, and from (6), $\mathrm{pp}(M'[2j] \oplus M_i[2j] \mid M'[2j-1] \ne M_i[2j-1])$ is at most $1/(2^n-1)$. We obtain

$$\begin{aligned}\mathrm{pp}(M'[2j] \oplus M_i[2j]) &\le \mathrm{pp}(M'[2j] \oplus M_i[2j] \mid M'[2j-1] \ne M_i[2j-1])\\ &\quad + \Pr\big[M'[2j-1] = M_i[2j-1]\big]\\ &\le \frac{1}{2^n-1} + \frac{1}{2^n-1}\,. \qquad (8)\end{aligned}$$

From (4), (7), and (8), we have $p \le 3/(2^n-1)$.

Subcase (2) is the case $j = \ell$ and the difference is only in the last chunk. Let $U' = \mathsf{ozp}_n(C'[m]) \oplus Z'$ and $U_i = \mathsf{ozp}_n(C_i[m]) \oplus Z_i$. Note that $U' \oplus U_i$ is an independent term in $S' \oplus S_i$, thus $\Pr[S' = S_i]$ is at most $\mathrm{pp}(U' \oplus U_i)$. We have

$$\begin{aligned}\mathrm{pp}(U' \oplus U_i) &\le \mathrm{pp}(U' \oplus U_i \mid M'[m-1] \ne M_i[m-1])\\ &\quad + \Pr\big[M'[m-1] = M_i[m-1]\big]\,. \qquad (9)\end{aligned}$$

If $\mathsf{ozp}_n(C'[m]) \ne \mathsf{ozp}_n(C_i[m])$, then we have $\Pr\big[M'[m-1] = M_i[m-1]\big] \le 1/(2^n-1)$, and given $M'[m-1] \ne M_i[m-1]$, Z' is uniform over $\{0,1\}^n \setminus \{Z_i\}$ (note that $\mathsf{ozp}_n(C_i[m])$ and Z_i are dependent but $\mathsf{ozp}_n(C'[m])$ and Z' are independent). Therefore, $\mathrm{pp}(U' \oplus U_i \mid M'[m-1] \ne M_i[m-1])$ is at most $1/(2^n-1)$. Combining this and (4) and (9), we have $p \le 3/(2^n-1)$.

If $\mathsf{ozp}_n(C'[m]) = \mathsf{ozp}_n(C_i[m])$, we have $C'[m] = C_i[m]$, and thus $C'[m-1] \ne C_i[m-1]$, since $\nu' = \nu_i$ is assumed. Then $M'[m-1] \ne M_i[m-1]$ holds. From (9), we have $\mathrm{pp}(U' \oplus U_i) \le 1/(2^n-1)$. From (4), we have $p \le 2/(2^n-1)$.

Combining Subcases (1) and (2), we have $p \le 3/(2^n-1)$.

Case 2.3. We have $\nu' = \nu_i \in \{4,5\}$, implying $m' \bmod 2 = m_i \bmod 2 = 1$. We use the $2n$-bit parsing as in Case 2.2. There exists $1 \le j \le \ell$ for $\ell = \ell' = \ell_i$ such that $V'[j] \ne V_i[j]$. If $1 \le j < \ell$, from the same analysis as in Subcase (1) of Case 2.2, we have $p \le 3/(2^n-1)$. If $j = \ell$ and the difference is only in the last chunk, we necessary have $C'[m] \ne C_i[m]$, where $m = m' = m_i$. Now if we have $B'[j] \ne B[j]$ for some $1 \le j < m$, then we have a difference in tweaks used for two-block chunks, and by following a similar analysis to Subcase (1) of Case 2.2, we obtain $p \le 3/(2^n-1)$. If $B'[j] = B[j]$ for all $1 \le j < m$, then let $U' = \mathsf{ozp}_n(M'[m])$ and $U_i = \mathsf{ozp}_n(M_i[m])$. Observe that $S' \oplus S_i = U' \oplus U_i$. When $\nu' = \nu_i = 5$, we have

$$U' \oplus U_i = M'[m] \oplus M_i[m] = C'[m] \oplus C_i[m] \ne 0^n\,.$$

When $\nu' = \nu_i = 4$, we again obtain $U' \oplus U_i \ne 0^n$ from the property of ozp. Therefore, $S' \ne S_i$ holds for both cases, and we have $p \le 1/(2^n-1)$.

Case 3.1. The analysis is identical to that of Case 2.1, and we have $p = 1/2^n$.

Case 3.2. The analysis is almost the same as those of Cases 2.2 and 2.3. We let $1 \le j < m$, where $m = m' = m_i$, be an index such that $B'[j] \ne B_i[j]$ (note that $j = m$ is excluded as $B'[m] = B_i[m]$). Then, we have a difference in tweaks used for two-block chunks, and by following Subcase (1) of Case 2.2, we have $p \le 3/(2^n - 1)$.

The remaining cases, Cases 3.3–3.6, correspond to Cases 3.2–3.5 of the proof of Theorem 1, respectively, and the same proof works. This is because iZOCB and iZOTR share the same high-level structure and the same specification of iHash.

Therefore, we have $p \le 3/(2^n - 1) \le 6/2^n$ for all the cases, and this gives the upper bound of $\mathbf{Adv}^{\mathrm{auth}}_{\mathsf{iZOTR}[\mathrm{Perm}(\mathcal{G},n)]}(\mathcal{A})$ against an adversary $\mathcal{A}$ that makes single decryption query. By using [BGM04], we conclude the proof.

5 ZOCB

Overview. ZOCB is a TBC-based NAE scheme that is obtained from iZOCB by instantiating the TURP $\widetilde{\mathsf{E}} \in \mathrm{Perm}(\mathcal{G}, n)$ with a concrete TBC $\widehat{\mathsf{E}}$ that has the global tweak space $\mathcal{G}$. We start from a TBC $E : \mathcal{K} \times \mathcal{W} \times \{0,1\}^n \to \{0,1\}^n$, where $\mathcal{W} = \mathcal{I} \times \{0,1\}^t$, $\{0,1,2,3\} \subseteq \mathcal{I}$, and $t \ge 1$. This corresponds to a TBC as a primitive and for instance E could be SKINNY or Deoxys-BC. We introduce a *tweak extension scheme*, called XTX^*, which efficiently converts the ideally secure version of E, which we write E, into a TBC $\widehat{\mathsf{E}}$ that has the global tweak space $\mathcal{G}$, where $\mathcal{G}$ includes $\{0,1\}^n$ for a nonce, $\{0,1\}^t$ for an AD block, and $\mathbb{Z}_\rho$ for a block counter. XTX^* combines XTX [MI15] and XT [IMPS17], and uses the idea of XEX* [Rog04] (which is a blockcipher mode for TBCs) for efficiently combining TPRP-secure and TSPRP-secure TBCs. XTX^* is formalized in Appendix A. ZOCB is obtained by instantiating $\widetilde{\mathsf{E}}$ in iZOCB with $\widehat{\mathsf{E}}$, which itself is constructed from E.

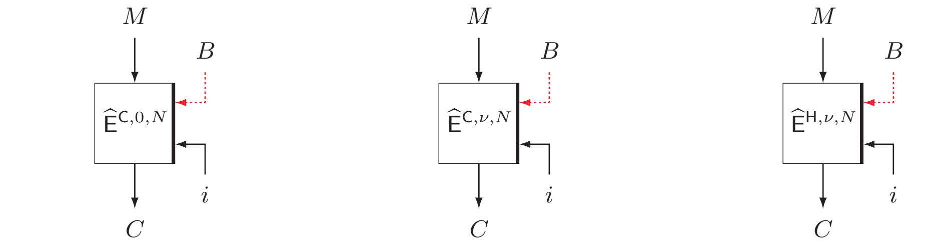
We summarize various TBCs that appear in this paper.

- $E : \mathcal{K} \times \mathcal{W} \times \{0,1\}^n \to \{0,1\}^n$ is the underlying TBC as a primitive. For instance this could be SKINNY or Deoxys-BC, and this corresponds to E_K in Fig. 1.
- $\mathsf{E} \in \mathrm{Perm}(\mathcal{W}, n)$ is a TURP that has the same tweak space and block length as E. This corresponds to the ideally secure version of E.
- $\widehat{\mathsf{E}} : \mathcal{K}_{\widehat{\mathsf{E}}} \times \mathcal{G} \times \{0,1\}^n \to \{0,1\}^n$ is a TBC that we construct by using E as a building block. The construction is based on XTX^*, and this has the global tweak space $\mathcal{G}$ as defined in (1). This corresponds to the dashed box in the rightmost figure of Fig. 1, where we use E instead of E_K.
- $\widetilde{\mathsf{E}} \in \mathrm{Perm}(\mathcal{G}, n)$ is a TURP that has the same tweak space and block length as $\widehat{\mathsf{E}}$, and this corresponds to the ideally secure version of $\widehat{\mathsf{E}}$. This was used to define iZOCB in Sect. 3.1.

See Fig. 13 for the input and output of $\widehat{\mathsf{E}}$, and Fig. 14 for the instantiation with E.

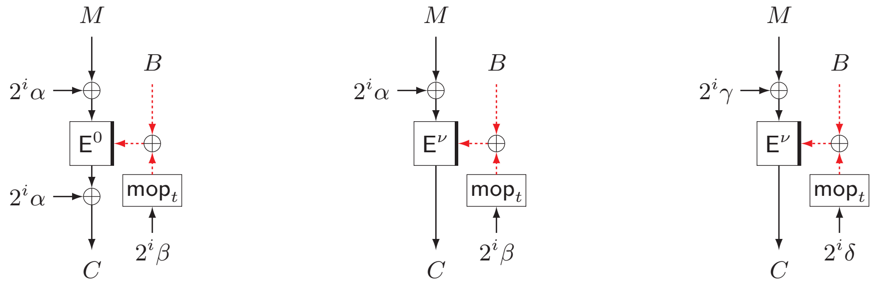
Specification of ZOCB. We present the specification of ZOCB. To use ZOCB, one has to specify a TBC $E : \mathcal{K} \times \mathcal{W} \times \{0,1\}^n \to \{0,1\}^n$ as a parameter, where $\mathcal{W} = \mathcal{I} \times \{0,1\}^t$, $\{0,1,2,3\} \subseteq \mathcal{I}$, and $t \ge 1$. We write $\mathrm{ZOCB}[E] = (\mathrm{ZOCB.Enc}, \mathrm{ZOCB.Dec})$ for ZOCB that uses E as the underlying TBC. We have $\mathcal{K}_{\mathrm{ZOCB}} = \mathcal{K}$, i.e., the key space is the key space of the underlying TBC, $\mathcal{N}_{\mathrm{ZOCB}} = \{0,1\}^n$, $\mathcal{A}_{\mathrm{ZOCB}} = \mathcal{M}_{\mathrm{ZOCB}} = \mathcal{C}_{\mathrm{ZOCB}} = \{0,1\}^{\le n\rho}$, and $\mathcal{T}_{\mathrm{ZOCB}} = \{0,1\}^n$. Recall that $\rho = 2^{(n+\min\{n,t\})/2} - 1$.

Let $\mathsf{E} \in \mathrm{Perm}(\mathcal{W}, n)$ be a TURP that has the same tweak space and block length as E. Now from $\mathsf{E} \in \mathrm{Perm}(\mathcal{W}, n)$, we define a TBC $\widehat{\mathsf{E}} : \mathcal{K}_{\widehat{\mathsf{E}}} \times \mathcal{G} \times \{0,1\}^n \to \{0,1\}^n$. The


M
B
$\widehat{\mathsf{E}}^{\mathsf{C},0,N}$
i
C
M
B
$\widehat{\mathsf{E}}^{\mathsf{C},\nu,N}$
i
C
M
B
$\widehat{\mathsf{E}}^{\mathsf{H},\nu,N}$
i
C

(a) $\widehat{\mathsf{E}}^{\mathsf{C},\nu,N,B,i}(M)$ for $\nu = 0$ **(b)** $\widehat{\mathsf{E}}^{\mathsf{C},\nu,N,B,i}(M)$ for $\nu \in \{1,2\}$ **(c)** $\widehat{\mathsf{E}}^{\mathsf{H},\nu,N,B,i}(M)$

Figure 13: (a) $C \leftarrow \widehat{\mathsf{E}}^{\mathsf{C},\nu,N,B,i}(M)$ for $\nu = 0$. The idealized version $\widetilde{\mathsf{E}}$ is used to process $B[1],\dots,B[m-1]$ and $M[1],\dots,M[m]$ in iCore.Enc of iZOCB. **(b)** $C \leftarrow \widehat{\mathsf{E}}^{\mathsf{C},\nu,N,B,i}(M)$ for $\nu \in \{1,2\}$. The idealized version is used to compute Y in iCore.Enc of iZOCB. **(c)** $C \leftarrow \widehat{\mathsf{E}}^{\mathsf{H},\nu,N,B,i}(M)$ for $\nu \in \{0,1,2\}$. The idealized version is used to compute $\widehat{Y}$ in iHash of iZOCB.



(a) $\widehat{\mathsf{E}}^{\mathsf{C},\nu,N,B,i}(M)$ for $\nu = 0$ **(b)** $\widehat{\mathsf{E}}^{\mathsf{C},\nu,N,B,i}(M)$ for $\nu \in \{1,2\}$ **(c)** $\widehat{\mathsf{E}}^{\mathsf{H},\nu,N,B,i}(M)$

Figure 14: (a) Instantiation of Fig. 13a with E. **(b)** Instantiation of Fig. 13b with E. **(c)** Instantiation of Fig. 13c with E. In these figures, $\alpha, \beta, \gamma, \delta$ are computed as (13).

key space is $\mathcal{K}_{\widehat{\mathsf{E}}} = \mathrm{Perm}(\mathcal{W}, n)$, i.e., it takes $\mathsf{E} \in \mathrm{Perm}(\mathcal{W}, n)$ as a key, the tweak space is $\mathcal{G}$, and for a global tweak $(\mathsf{b}, \nu, N, B, i) \in \mathcal{G}$, the encryption of $M \in \{0,1\}^n$ is defined as follows.

$$\begin{cases} \widehat{\mathsf{E}}^{\mathsf{C},\nu,N,B,i}_{\mathsf{E}}(M) = \mathsf{E}^{\nu,\mathsf{mop}_t(2^i\beta)\oplus B}(M \oplus 2^i\alpha) \oplus 2^i\alpha & \text{if } \nu = 0 \quad (10) \\ \widehat{\mathsf{E}}^{\mathsf{C},\nu,N,B,i}_{\mathsf{E}}(M) = \mathsf{E}^{\nu,\mathsf{mop}_t(2^i\beta)\oplus B}(M \oplus 2^i\alpha) & \text{if } \nu \in \{1,2\} \quad (11) \\ \widehat{\mathsf{E}}^{\mathsf{H},\nu,N,B,i}_{\mathsf{E}}(M) = \mathsf{E}^{\nu,\mathsf{mop}_t(2^i\delta)\oplus B}(M \oplus 2^i\gamma) & \quad (12) \end{cases}$$

Here, α, β, γ, and δ are defined as

$$\alpha = \mathsf{E}^{3,[0]_t}(N)\,,\ \beta = \mathsf{E}^{3,[1]_t}(N)\,,\ \gamma = \mathsf{E}^{3,[2]_t}(0^n)\,,\ \text{and } \delta = \mathsf{E}^{3,[3]_t}(0^n)\,. \tag{13}$$

Figures 14a, 14b, and 14c implement (10), (11), and (12), respectively. The decryption of $C \in \{0,1\}^n$, which we write $\widehat{\mathsf{D}}$, uses the decryption D of E and is defined in an obvious way as follows.

$$\begin{cases} \widehat{\mathsf{D}}^{\mathsf{C},\nu,N,B,i}_{\mathsf{E}}(C) = \mathsf{D}^{\nu,\mathsf{mop}_t(2^i\beta)\oplus B}(C \oplus 2^i\alpha) \oplus 2^i\alpha & \text{if } \nu = 0 \\ \widehat{\mathsf{D}}^{\mathsf{C},\nu,N,B,i}_{\mathsf{E}}(C) = \mathsf{D}^{\nu,\mathsf{mop}_t(2^i\beta)\oplus B}(C) \oplus 2^i\alpha & \text{if } \nu \in \{1,2\} \\ \widehat{\mathsf{D}}^{\mathsf{H},\nu,N,B,i}_{\mathsf{E}}(C) = \mathsf{D}^{\nu,\mathsf{mop}_t(2^i\delta)\oplus B}(C) \oplus 2^i\gamma & \end{cases}$$

Now we are ready to define ZOCB and present its specification. We observe that iZOCB$[\mathrm{Perm}(\mathcal{G}, n)]$ takes a TURP $\widetilde{\mathsf{E}} \in \mathrm{Perm}(\mathcal{G}, n)$ as the key, and this can be instantiated

```
Algorithm $\mathrm{ZOCB.Enc}_K(N, A, M)$

$m \leftarrow |M|_n$
if $|A| < mt$ then
    $B \leftarrow \mathsf{ozp}_{mt}(A)$
    $\widehat{Y} \leftarrow 0^n$
else
    $(B, \widehat{B}) \xleftarrow{mt,*} A$
    $\widehat{Y} \leftarrow \mathrm{Hash}_K(\widehat{B})$
$(C, Y) \leftarrow \mathrm{Core.Enc}_K(N, B, M)$
$T \leftarrow \widehat{Y} \oplus Y$
return $(C, T)$
```

```
Algorithm $\mathrm{Core.Enc}_K(N, B, M)$
// $M \in \{0,1\}^*$, $|M|_n = m$, and $|B| = mt$

$S \leftarrow 0^n$
$\alpha \leftarrow E_K^{3,[0]_t}(N)$, $\beta \leftarrow E_K^{3,[1]_t}(N)$
$(M[1], \ldots, M[m]) \xleftarrow{n} M$
$(B[1], \ldots, B[m]) \xleftarrow{t} B$
for $i = 1$ to $m - 1$ do    // $m \geq 2$
    $S \leftarrow S \oplus M[i]$
    $W[i] \leftarrow B[i] \oplus \mathsf{mop}_t(\beta)$
    $C[i] \leftarrow E_K^{0,W[i]}(M[i] \oplus \alpha) \oplus \alpha$
    $\alpha \leftarrow 2 \cdot \alpha$, $\beta \leftarrow 2 \cdot \beta$
$W[m] \leftarrow \mathsf{mop}_t(\beta)$
$Z \leftarrow E_K^{0,W[m]}(\alpha) \oplus \alpha$
$C[m] \leftarrow M[m] \oplus \mathsf{msb}_{|M[m]|}(Z)$
$S \leftarrow S \oplus \mathsf{ozp}_n(M[m])$
$W[m+1] \leftarrow B[m] \oplus \mathsf{mop}_t(\beta)$
if $|M[m]| \neq n$ then
    $Y \leftarrow E_K^{1,W[m+1]}(S \oplus \alpha)$
else
    $Y \leftarrow E_K^{2,W[m+1]}(S \oplus \alpha)$
$C \leftarrow (C[1], \ldots, C[m])$
return $(C, Y)$
```

```
Algorithm $\mathrm{ZOCB.Dec}_K(N, A, C, T)$

$m \leftarrow |C|_n$                                  ◁
if $|A| < mt$ then
    $B \leftarrow \mathsf{ozp}_{mt}(A)$
    $\widehat{Y} \leftarrow 0^n$
else
    $(B, \widehat{B}) \xleftarrow{mt,*} A$
    $\widehat{Y} \leftarrow \mathrm{Hash}_K(\widehat{B})$
$(M, Y) \leftarrow \mathrm{Core.Dec}_K(N, B, C)$        ◁
$T^* \leftarrow \widehat{Y} \oplus Y$                 ◁
if $T^* = T$ then return $M$                          ◁
else return $\bot$                                    ◁
```

```
Algorithm $\mathrm{Core.Dec}_K(N, B, C)$
// $C \in \{0,1\}^*$, $|C|_n = m$, and $|B| = mt$

$S \leftarrow 0^n$
$\alpha \leftarrow E_K^{3,[0]_t}(N)$, $\beta \leftarrow E_K^{3,[1]_t}(N)$
$(C[1], \ldots, C[m]) \xleftarrow{n} C$                ◁
$(B[1], \ldots, B[m]) \xleftarrow{t} B$
for $i = 1$ to $m - 1$ do    // $m \geq 2$
    $W[i] \leftarrow B[i] \oplus \mathsf{mop}_t(\beta)$     ◁
    $M[i] \leftarrow D_K^{0,W[i]}(C[i] \oplus \alpha) \oplus \alpha$ ◁
    $S \leftarrow S \oplus M[i]$                    ◁
    $\alpha \leftarrow 2 \cdot \alpha$, $\beta \leftarrow 2 \cdot \beta$
$W[m] \leftarrow \mathsf{mop}_t(\beta)$
$Z \leftarrow E_K^{0,W[m]}(\alpha) \oplus \alpha$
$M[m] \leftarrow C[m] \oplus \mathsf{msb}_{|C[m]|}(Z)$   ◁
$S \leftarrow S \oplus \mathsf{ozp}_n(M[m])$
$W[m+1] \leftarrow B[m] \oplus \mathsf{mop}_t(\beta)$
if $|C[m]| \neq n$ then                              ◁
    $Y \leftarrow E_K^{1,W[m+1]}(S \oplus \alpha)$
else
    $Y \leftarrow E_K^{2,W[m+1]}(S \oplus \alpha)$
$M \leftarrow (M[1], \ldots, M[m])$                   ◁
return $(M, Y)$                                       ◁
```

Figure 15: Definitions of $\mathrm{ZOCB.Enc}_K(N, A, M)$ and $\mathrm{ZOCB.Dec}_K(N, A, C, T)$. Recall that the blue lines with a blue triangle in $\mathrm{ZOCB.Dec}_K(N, A, C, T)$ and $\mathrm{Core.Dec}_K(N, B, C)$ indicate the difference from the corresponding encryption algorithms.

with any TBC in $\mathrm{Perm}(\mathcal{G}, n)$. In particular, $\widehat{\mathsf{E}}$ can be used as the underlying TBC, which itself takes $\mathsf{E} \in \mathrm{Perm}(\mathcal{W}, n)$ as a key. Let $\mathsf{iZOCB}[\widehat{\mathsf{E}}]$ be iZOCB that uses $\widehat{\mathsf{E}}$ as the underlying TBC and takes $\mathsf{E} \in \mathrm{Perm}(\mathcal{W}, n)$ as a key.

For $\mathsf{E} \in \mathrm{Perm}(\mathcal{W}, n)$, i.e., when we use a TURP $\mathsf{E} \in \mathrm{Perm}(\mathcal{W}, n)$, $\mathrm{ZOCB}[\mathrm{Perm}(\mathcal{W}, n)]$ is define as $\mathsf{iZOCB}[\widehat{\mathsf{E}}]$, where $\widehat{\mathsf{E}}$ is obtained from E as stated above.

For a TBC E, $\mathrm{ZOCB}[E]$ is defined as $\mathsf{iZOCB}[\widehat{\mathsf{E}}]$, where in $\widehat{\mathsf{E}}$, we use E_K for randomly chosen key $K \xleftarrow{\$} \mathcal{K}$ instead of E. See Figs. 13 and 14. We present the specification of $\mathrm{ZOCB}[E]$ in Figs. 15 and 16. The encryption algorithm is illustrated in Figs. 17, 18, and 19.

```
Algorithm Hash_K($\widehat{B}$)

$\widehat{Y} \leftarrow 0^n$
$\gamma \leftarrow E_K^{3,[2]_t}(0^n)$, $\delta \leftarrow E_K^{3,[3]_t}(0^n)$
$(\widehat{B}[1], \ldots, \widehat{B}[a]) \xleftarrow{n+t} \widehat{B}$
for $i = 1$ to $a - 1$ do  // $a \geq 2$
    $(P[i], Q[i]) \xleftarrow{n,t} \widehat{B}[i]$
    $W[i] \leftarrow Q[i] \oplus \mathsf{mop}_t(\delta)$
    $\widehat{Y} \leftarrow \widehat{Y} \oplus E_K^{0,W[i]}(P[i] \oplus \gamma)$
    $\gamma \leftarrow 2 \cdot \gamma$, $\delta \leftarrow 2 \cdot \delta$
$(P[a], Q[a]) \xleftarrow{n,t} \mathsf{ozp}_{n+t}(\widehat{B}[a])$
$W[a] \leftarrow Q[a] \oplus \mathsf{mop}_t(\delta)$
if $|\widehat{B}[a]| \neq n + t$ then
    $\widehat{Y} \leftarrow \widehat{Y} \oplus E_K^{1,W[a]}(P[a] \oplus \gamma)$
else
    $\widehat{Y} \leftarrow \widehat{Y} \oplus E_K^{2,W[a]}(P[a] \oplus \gamma)$
return $\widehat{Y}$
```

Figure 16: Definition of $\mathrm{Hash}_K(\widehat{B})$ of ZOCB

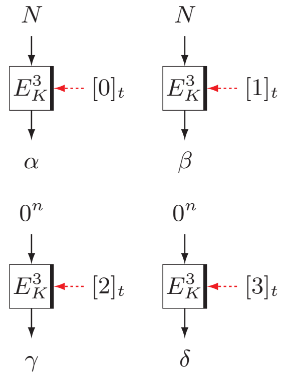


Figure 17: Generation of masks α, β, γ, and δ. Recall that the red dashed lines are t bits wide (also in other figures).

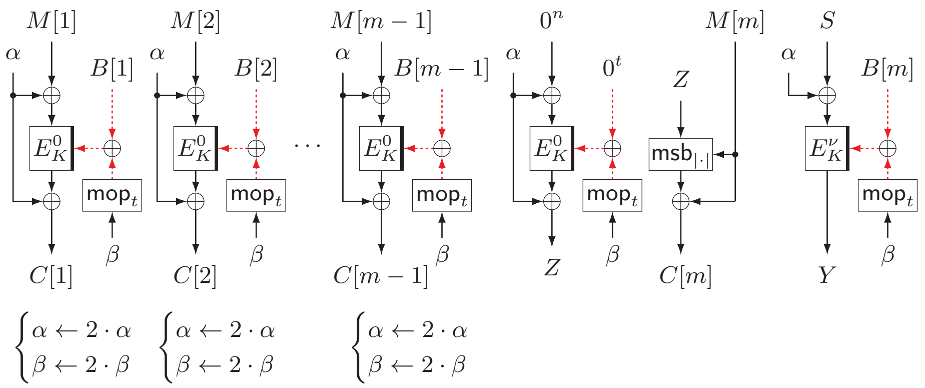


Figure 18: $\mathrm{Core.Enc}_K(N, B, M)$ of ZOCB, where $(B[1], \ldots, B[m]) \xleftarrow{t} B$. The value of ν in the last step is 1 if $|M[m]| \neq n$ and 2 otherwise. S is $M[1] \oplus \cdots \oplus M[m-1] \oplus \mathsf{ozp}_n(M[m])$.

6 ZOTR

Overview. ZOTR is obtained from iZOTR by instantiating the TURP $\widetilde{\mathsf{E}}$ with a concrete TBC E. We extend the tweak space of E to obtain $\widehat{\mathsf{E}}$ that has global tweak space $\mathcal{G}$ (defined in (2)) based on the tweak extension scheme XTX^*, so that each call of $\widehat{\mathsf{E}}$ takes a t-bit block of AD as a part of the tweak input. Since we need only the encryption routine of $\widehat{\mathsf{E}}$, the tweak extension is simpler than that of ZOCB in that we do not have the XOR of masks in the output of the TBC. ZOTR is defined as iZOTR by instantiating $\widetilde{\mathsf{E}}$ with $\widehat{\mathsf{E}}$.

Specification of ZOTR. As with ZOCB, to use ZOTR, one has to specify a TBC $E : \mathcal{K} \times \mathcal{W} \times \{0,1\}^n \to \{0,1\}^n$ as a parameter, where $\mathcal{W} = \mathcal{I} \times \{0,1\}^t$, $\{0, 1, \ldots, 6\} \subseteq \mathcal{I}$, and $t \geq 1$. We write $\mathrm{ZOTR}[E] = (\mathrm{ZOTR.Enc}, \mathrm{ZOTR.Dec})$ for ZOTR that uses E as the underlying TBC. We have $\mathcal{K}_{\mathrm{ZOTR}} = \mathcal{K}$, $\mathcal{N}_{\mathrm{ZOTR}} = \{0,1\}^n$, $\mathcal{A}_{\mathrm{ZOTR}} = \mathcal{M}_{\mathrm{ZOTR}} = \mathcal{C}_{\mathrm{ZOTR}} =$

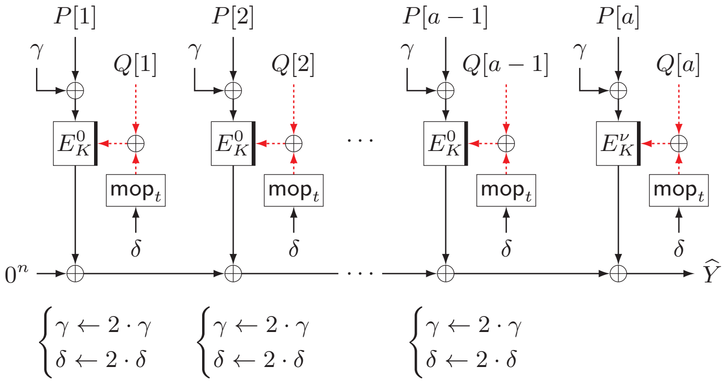
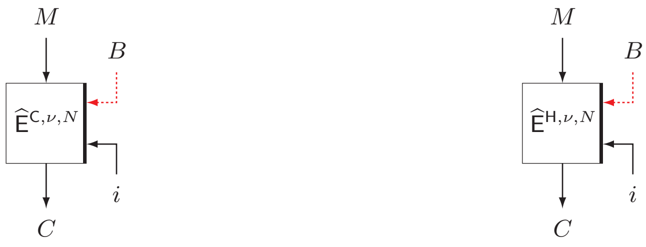


Figure 19: $\mathrm{Hash}_K(\widehat{B})$ of ZOCB, where $(\widehat{B}[1], \ldots, \widehat{B}[a]) \xleftarrow{n+t} \widehat{B}$ and $(P[i], Q[i]) \xleftarrow{n,t} \widehat{B}[i]$. ν in the last step is 1 if $|\widehat{B}[a]| \neq n + t$ and 2 otherwise.


M
B
$\widehat{\mathsf{E}}^{\mathsf{C},\nu,N}$
i
C
M
B
$\widehat{\mathsf{E}}^{\mathsf{H},\nu,N}$
i
C

(a) $\widehat{\mathsf{E}}^{\mathsf{C},\nu,N,B,i}(M)$ for $\nu \in \{0, 1, \ldots, 5\}$ **(b)** $\widehat{\mathsf{E}}^{\mathsf{H},\nu,N,B,i}(M)$ for $\nu \in \{0, 1, 2\}$

Figure 20: (a) $C \leftarrow \widehat{\mathsf{E}}^{\mathsf{C},\nu,N,B,i}(M)$ for $\nu \in \{0, 1, \ldots, 5\}$. The idealized version $\widetilde{\mathsf{E}}$ is used in iCore.Enc of iZOTR. **(b)** $C \leftarrow \widehat{\mathsf{E}}^{\mathsf{H},\nu,N,B,i}(M)$ for $\nu \in \{0, 1, 2\}$. The idealized version is used in iHash of iZOTR.

$\{0, 1\}^{\leq n\rho}$, and $\mathcal{T}_{\mathrm{ZOTR}} = \{0, 1\}^n$, where $\rho = 2^{(n+\min\{n,t\})/2} - 1$.

We define a TBC $\widehat{\mathsf{E}} : \mathcal{K}_{\widehat{\mathsf{E}}} \times \mathcal{G} \times \{0, 1\}^n \to \{0, 1\}^n$ from $\mathsf{E} \in \mathrm{Perm}(\mathcal{W}, n)$. Here, $\mathcal{K}_{\widehat{\mathsf{E}}} = \mathrm{Perm}(\mathcal{W}, n)$ and the tweak space is $\mathcal{G}$, which is defined in (2). For a global tweak $(\mathsf{b}, \nu, N, B, i) \in \mathcal{G}$, the encryption of $M \in \{0, 1\}^n$ is defined as

$$\begin{cases} \widehat{\mathsf{E}}_{\mathsf{E}}^{\mathsf{C},\nu,N,B,i}(M) = \mathsf{E}^{\nu,\mathsf{mop}_t(2^i\beta)\oplus B}(M \oplus 2^i\alpha)\,, & (14) \\ \widehat{\mathsf{E}}_{\mathsf{E}}^{\mathsf{H},\nu,N,B,i}(M) = \mathsf{E}^{\nu,\mathsf{mop}_t(2^i\delta)\oplus B}(M \oplus 2^i\gamma)\,, & (15) \end{cases}$$

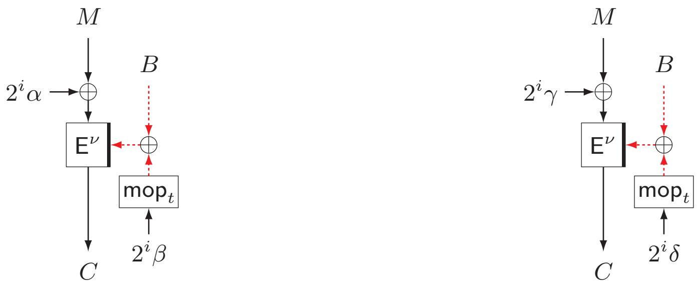
where

$$\alpha = \mathsf{E}^{6,[0]_t}(N)\,,\ \beta = \mathsf{E}^{6,[1]_t}(N)\,,\ \gamma = \mathsf{E}^{6,[2]_t}(0^n)\,,\ \text{and}\ \delta = \mathsf{E}^{6,[3]_t}(0^n)\,. \qquad (16)$$

The decryption is defined in an obvious way (and we omit this as this will not be used). We note that we can see it as XT introduced in [IMPS17]. See Fig. 20 for the input and output of $\widehat{\mathsf{E}}$, and Fig. 21 for the instantiation with E in (14) and (15).

We are now ready to define ZOTR. Let $\mathsf{iZOTR}[\widehat{\mathsf{E}}]$ be iZOTR that uses $\widehat{\mathsf{E}}$ as the underlying TBC and takes $\mathsf{E} \in \mathrm{Perm}(\mathcal{W}, n)$ as a key. For $\mathsf{E} \in \mathrm{Perm}(\mathcal{W}, n)$, $\mathrm{ZOTR}[\mathrm{Perm}(\mathcal{W}, n)]$ is defined as $\mathsf{iZOTR}[\widehat{\mathsf{E}}]$, where $\widehat{\mathsf{E}}$ is constructed from E as above. For a TBC E, $\mathrm{ZOTR}[E]$ is defined as $\mathsf{iZOTR}[\widehat{\mathsf{E}}]$, where $\widehat{\mathsf{E}}$ is obtained by using E_K instead of E.

The specification of $\mathrm{ZOTR}[E]$ is in Figs. 22, 23, 24, and 25. See Figs. 26, 27, and 28 for illustrations. We note that, as with the case of iZOTR, the pseudocode of $\mathrm{ZOTR.Enc}_K(N, A, M)$ and $\mathrm{ZOTR.Dec}_K(N, A, C, T)$ is the same as the corresponding



(a) $\widehat{\mathsf{E}}^{\mathsf{C},\nu,N,B,i}(M)$ for $\nu \in \{0,1,\ldots,5\}$

(b) $\widehat{\mathsf{E}}^{\mathsf{C},\nu,N,B,i}(M)$ for $\nu \in \{0,1,2\}$

Figure 21: **(a)** Instantiation of Fig. 20a with E. **(b)** Instantiation of Fig. 20b with E. $\alpha, \beta, \gamma, \delta$ are computed as (16).

```
Algorithm ZOTR.Enc_K(N, A, M)

m ← |M|_n
if |A| < mt then
    B ← ozp_mt(A)
    Ŷ ← 0^n
else
    (B, B̂) ←(mt,*)− A
    Ŷ ← Hash_K(B̂)
(C, Y) ← Core.Enc_K(N, B, M)
T ← Ŷ ⊕ Y
return (C, T)
```

```
Algorithm ZOTR.Dec_K(N, A, C, T)

m ← |C|_n                              ◁
if |A| < mt then
    B ← ozp_mt(A)
    Ŷ ← 0^n
else
    (B, B̂) ←(mt,*)− A
    Ŷ ← Hash_K(B̂)
(M, Y) ← Core.Dec_K(N, B, C)           ◁
T* ← Ŷ ⊕ Y                             ◁
if T* = T then return M                ◁
else return ⊥                          ◁
```

Figure 22: Definitions of ZOTR.Enc$_K(N, A, M)$ and ZOTR.Dec$_K(N, A, C, T)$

pseudocode of ZOCB, and Hash$_K(\widehat{B})$ is also the same except for the generation of γ and δ.

7 Security of ZOCB and ZOTR

7.1 Security of ZOCB

Privacy Theorem. Let $\mathcal{A}$ be a privacy adversary that makes q queries, and suppose that the queries are $(N_1, A_1, M_1), \ldots, (N_q, A_q, M_q)$. For $1 \le i \le q$, let $m_i = |M_i|_n$. If $|A_i| < m_i t$, then let $\sigma_i = m_i + 1$. Otherwise, we let $\sigma_i = m_i + 1 + |\widehat{B}_i|_{n+t}$, where $(B_i, \widehat{B}_i) \xleftarrow{m_i t,*} A_i$. Then we define the query complexity as $\sigma_{\text{priv}} = 2q + 2 + \sum_{1 \le i \le q} \sigma_i$, which corresponds to the maximum number of calls to the underlying TBC. We have the following information theoretic result for the privacy of ZOCB.

Theorem 3. *Let $\mathcal{A}$ be a privacy adversary against* ZOCB[Perm$(\mathcal{W}, n)$] *that makes at most q queries with the query complexity at most σ_{priv}. Then we have* $\mathbf{Adv}^{\text{priv}}_{\text{ZOCB[Perm}(\mathcal{W},n)]}(\mathcal{A}) \le 4\sigma^2_{\text{priv}}/2^{n+\min\{n,t\}}$.

A proof is presented in Sect. 7.2. Note that privacy adversaries are nonce-respecting. If we use a TBC E which is secure in the sense of the TPRP notion instead of Perm$(\mathcal{W}, n)$, then the corresponding complexity theoretic result can be shown by a standard argument. See e.g., [BKR00].

```
**Algorithm** Core.Enc$_K(N, B, M)$    // $M \in \{0,1\}^*$, $|M|_n = m$, and $|B| = mt$
$S \leftarrow 0^n$
$\alpha \leftarrow E_K^{6,[0]_t}(N)$, $\beta \leftarrow E_K^{6,[1]_t}(N)$
$(M[1], \dots, M[m]) \xleftarrow{n} M$
$\ell \leftarrow \lceil m/2 \rceil$
$(B[1], \dots, B[m]) \xleftarrow{t} B$
**for** $i = 1$ **to** $\ell - 1$ **do**    // $\ell \geq 2$
    $W[2i-1] \leftarrow B[2i-1] \oplus \mathsf{mop}_t(\beta)$
    $C[2i-1] \leftarrow E_K^{0,W[2i-1]}(M[2i-1] \oplus \alpha) \oplus M[2i]$
    $W[2i] \leftarrow B[2i] \oplus \mathsf{mop}_t(\beta)$
    $C[2i] \leftarrow E_K^{1,W[2i]}(C[2i-1] \oplus \alpha) \oplus M[2i-1]$
    $S \leftarrow S \oplus M[2i]$
    $\alpha \leftarrow 2 \cdot \alpha$, $\beta \leftarrow 2 \cdot \beta$
**if** $m \bmod 2 = 0$ **then**
    $W[m-1] \leftarrow B[m-1] \oplus \mathsf{mop}_t(\beta)$
    $Z \leftarrow E_K^{0,W[m-1]}(M[m-1] \oplus \alpha)$
    $C[m] \leftarrow \mathsf{msb}_{|M[m]|}(Z) \oplus M[m]$
    $W[m] \leftarrow \mathsf{mop}_t(\beta)$
    $C[m-1] \leftarrow E_K^{1,W[m]}(\mathsf{ozp}_n(C[m]) \oplus \alpha) \oplus M[m-1]$
    $S \leftarrow S \oplus \mathsf{ozp}_n(C[m]) \oplus Z$
**if** $m \bmod 2 = 1$ **then**
    $W[m] \leftarrow \mathsf{mop}_t(\beta)$
    $C[m] \leftarrow \mathsf{msb}_{|M[m]|}(E_K^{0,W[m]}(\alpha)) \oplus M[m]$
    $S \leftarrow S \oplus \mathsf{ozp}_n(M[m])$
$W[m+1] \leftarrow B[m] \oplus \mathsf{mop}_t(\beta)$
**if** $m \bmod 2 = 0$ **and** $|M[m]| \neq n$ **then** $Y \leftarrow E_K^{2,W[m+1]}(S \oplus \alpha)$
**if** $m \bmod 2 = 0$ **and** $|M[m]| = n$ **then** $Y \leftarrow E_K^{3,W[m+1]}(S \oplus \alpha)$
**if** $m \bmod 2 = 1$ **and** $|M[m]| \neq n$ **then** $Y \leftarrow E_K^{4,W[m+1]}(S \oplus \alpha)$
**if** $m \bmod 2 = 1$ **and** $|M[m]| = n$ **then** $Y \leftarrow E_K^{5,W[m+1]}(S \oplus \alpha)$
$C \leftarrow (C[1], \dots, C[m])$
**return** $(C, Y)$
```

Figure 23: Definition of Core.Enc$_K(N, B, M)$ of ZOTR

Authenticity Theorem. Let $\mathcal{A}$ be an authenticity adversary that makes q encryption queries and q' decryption queries. Let $(N_1, A_1, M_1), \dots, (N_q, A_q, M_q)$ be the encryption queries, and $(N'_1, A'_1, C'_1, T'_1), \dots, (N'_{q'}, A'_{q'}, C'_{q'}, T'_{q'})$ be the decryption queries. We define $\sigma_1, \dots, \sigma_q$ as in the privacy case. For $1 \leq j \leq q'$, let $m'_j = |C'_j|_n$. If $|A'_j| < m'_j t$, then let $\sigma'_j = m'_j + 1$. Otherwise, let $\sigma'_j = m'_j + 1 + |\widehat{B}'_j|_{n+t}$, where $(B'_j, \widehat{B}'_j) \xleftarrow{m'_j t,*} A'_j$. Then we define the query complexity as $\sigma_{\mathrm{auth}} = 2(q + q') + 2 + \sum_{1 \leq i \leq q} \sigma_i + \sum_{1 \leq j \leq q'} \sigma'_j$. We have the following information theoretic result for the authenticity of ZOCB.

Theorem 4. *Let $\mathcal{A}$ be an authenticity adversary against* ZOCB[Perm$(\mathcal{W}, n)$] *that makes at most q encryption queries and at most q' decryption queries, where the query complexity is at most σ_{auth}. Then we have* $\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{ZOCB[Perm}(\mathcal{W},n)]}(\mathcal{A}) \leq 4\sigma_{\mathrm{auth}}^2 / 2^{n+\min\{n,t\}} + 4q'/2^n$.

A proof is presented in Sect. 7.2. As in the privacy case, if we use a TBC E secure in the sense of the TSPRP notion, then we obtain the corresponding complexity theoretic result by following e.g., [BKR00].

```
Algorithm Core.Dec_K(N,B,C)        // $C \in \{0,1\}^*$, $|C|_n = m$, and $|B| = mt$

$S \leftarrow 0^n$
$\alpha \leftarrow E_K^{6,[0]_t}(N)$, $\beta \leftarrow E_K^{6,[1]_t}(N)$
$(C[1], \ldots, C[m]) \xleftarrow{n} C$                                  $\triangleleft$
$\ell \leftarrow \lceil m/2 \rceil$
$(B[1], \ldots, B[m]) \xleftarrow{t} B$
for $i = 1$ to $\ell - 1$ do                                            // $\ell \geq 2$
    $W[2i] \leftarrow B[2i] \oplus \mathsf{mop}_t(\beta)$                $\triangleleft$
    $M[2i-1] \leftarrow E_K^{1,W[2i]}(C[2i-1] \oplus \alpha) \oplus C[2i]$      $\triangleleft$
    $W[2i-1] \leftarrow B[2i-1] \oplus \mathsf{mop}_t(\beta)$            $\triangleleft$
    $M[2i] \leftarrow E_K^{0,W[2i-1]}(M[2i-1] \oplus \alpha) \oplus C[2i-1]$    $\triangleleft$
    $S \leftarrow S \oplus M[2i]$
    $\alpha \leftarrow 2 \cdot \alpha$, $\beta \leftarrow 2 \cdot \beta$
if $m \bmod 2 = 0$ then
    $W[m] \leftarrow \mathsf{mop}_t(\beta)$                              $\triangleleft$
    $M[m-1] \leftarrow E_K^{1,W[m]}(\mathsf{ozp}_n(C[m]) \oplus \alpha) \oplus C[m-1]$   $\triangleleft$
    $W[m-1] \leftarrow B[m-1] \oplus \mathsf{mop}_t(\beta)$              $\triangleleft$
    $Z \leftarrow E_K^{0,W[m-1]}(M[m-1] \oplus \alpha)$                  $\triangleleft$
    $M[m] \leftarrow \mathsf{msb}_{|C[m]|}(Z) \oplus C[m]$               $\triangleleft$
    $S \leftarrow S \oplus \mathsf{ozp}_n(C[m]) \oplus Z$
if $m \bmod 2 = 1$ then
    $W[m] \leftarrow \mathsf{mop}_t(\beta)$
    $M[m] \leftarrow \mathsf{msb}_{|C[m]|}(E_K^{0,W[m]}(\alpha)) \oplus C[m]$   $\triangleleft$
    $S \leftarrow S \oplus \mathsf{ozp}_n(M[m])$
$W[m+1] \leftarrow B[m] \oplus \mathsf{mop}_t(\beta)$
if $m \bmod 2 = 0$ and $|C[m]| \neq n$ then $Y \leftarrow E_K^{2,W[m+1]}(S \oplus \alpha)$   $\triangleleft$
if $m \bmod 2 = 0$ and $|C[m]| = n$ then $Y \leftarrow E_K^{3,W[m+1]}(S \oplus \alpha)$      $\triangleleft$
if $m \bmod 2 = 1$ and $|C[m]| \neq n$ then $Y \leftarrow E_K^{4,W[m+1]}(S \oplus \alpha)$   $\triangleleft$
if $m \bmod 2 = 1$ and $|C[m]| = n$ then $Y \leftarrow E_K^{5,W[m+1]}(S \oplus \alpha)$      $\triangleleft$
$M \leftarrow (M[1], \ldots, M[m])$                                      $\triangleleft$
return $(M, Y)$                                                          $\triangleleft$
```

Figure 24: Definition of Core.Dec$_K(N, B, C)$ of ZOTR

7.2 Proofs of Theorems 3 and 4

Overview. We consider $\widehat{\mathsf{E}}$ that is defined in Sect. 5, and we first formalize TPRP* and TSPRP* notions for it. These notions capture the indistinguishability between the real world (when the oracle implements $\widehat{\mathsf{E}}$) and the ideal world (when the oracle is $\widetilde{\mathsf{E}}$), and we show that $\widehat{\mathsf{E}}$ is secure in these notions (Proposition 1). Since we know that iZOCB is a secure NAE scheme (Theorem 1), we obtain Theorems 3 and 4 by noting that the adversary in the TPRP* and TSPRP* notions can simulate ZOCB (in the real world) or iZOCB (in the ideal world).

TPRP* and TSPRP* Notions. For the TBC $\widehat{\mathsf{E}} : \mathcal{K}_{\widehat{\mathsf{E}}} \times \mathcal{G} \times \{0,1\}^n \to \{0,1\}^n$ defined in Sect. 5, recall that $\widetilde{\mathsf{E}} \in \mathrm{Perm}(\mathcal{G}, n)$ is the corresponding TURP and $\widetilde{\mathsf{D}}$ is its decryption. We consider the security notions which we call TPRP* and TSPRP*. Let $\mathcal{B}$ be an adversary in the TSPRP* notion. The goal is to distinguish between $\widehat{\mathsf{E}}$ and $\mathrm{Perm}(\mathcal{G}, n)$, with the constraint that $\mathcal{B}$ does not have access to the decryption if it uses a tweak

```
Algorithm Hash_K($\widehat{B}$)

$\widehat{Y} \leftarrow 0^n$
$\gamma \leftarrow E_K^{6,[2]_t}(0^n)$, $\delta \leftarrow E_K^{6,[3]_t}(0^n)$
$(\widehat{B}[1], \dots, \widehat{B}[a]) \xleftarrow{n+t} \widehat{B}$
for $i = 1$ to $a - 1$ do   // $a \geq 2$
    $(P[i], Q[i]) \xleftarrow{n,t} \widehat{B}[i]$
    $W[i] \leftarrow Q[i] \oplus \mathsf{mop}_t(\delta)$
    $\widehat{Y} \leftarrow \widehat{Y} \oplus E_K^{0,W[i]}(P[i] \oplus \gamma)$
    $\gamma \leftarrow 2 \cdot \gamma$, $\delta \leftarrow 2 \cdot \delta$
$(P[a], Q[a]) \xleftarrow{n,t} \mathsf{ozp}_{n+t}(\widehat{B}[a])$
$W[a] \leftarrow Q[a] \oplus \mathsf{mop}_t(\delta)$
if $|\widehat{B}[a]| \neq n + t$ then
    $\widehat{Y} \leftarrow \widehat{Y} \oplus E_K^{1,W[a]}(P[a] \oplus \gamma)$
else
    $\widehat{Y} \leftarrow \widehat{Y} \oplus E_K^{2,W[a]}(P[a] \oplus \gamma)$
return $\widehat{Y}$
```

Figure 25: Definition of $\text{Hash}_K(\widehat{B})$ of ZOTR

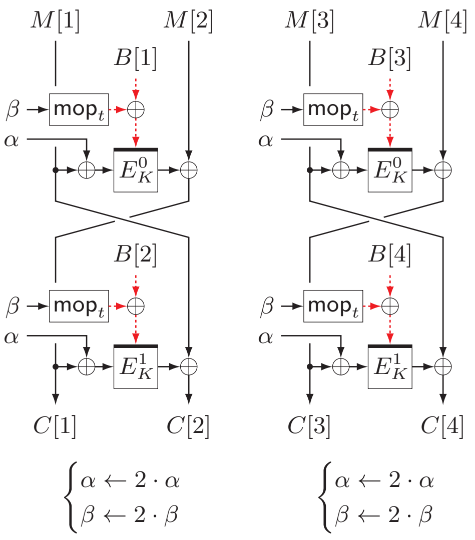


Figure 26: $\text{Core.Enc}_K(N, B, M)$ of ZOTR for the process of $M[1], \dots, M[4]$ and $B[1], \dots, B[4]$ when $|M|_n = m \geq 5$

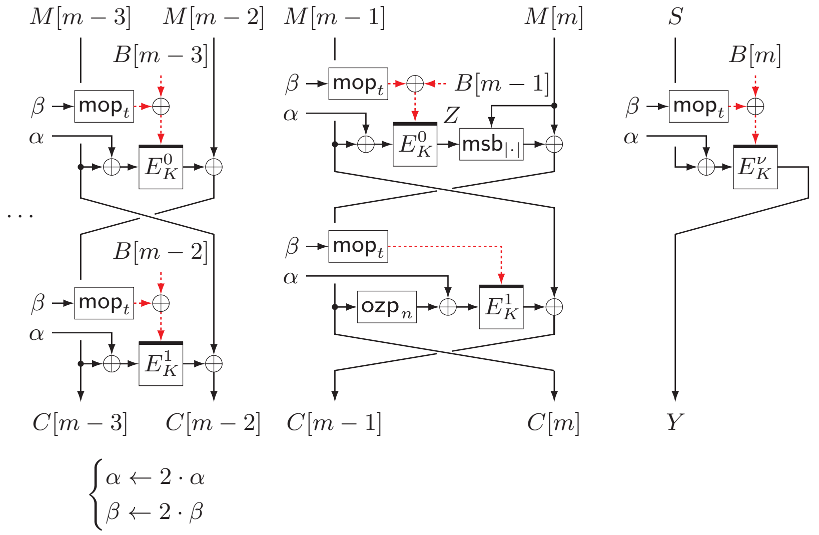


Figure 27: $\text{Core.Enc}_K(N, B, M)$ of ZOTR for the case $m \bmod 2 = 0$. This illustrates the process of $M[m-3], \dots, M[m]$ and $B[m-3], \dots, B[m]$, where $(B[1] \dots, B[m]) \xleftarrow{t} B$. ν in the last step is 2 if $|M[m]| \neq n$ and 3 otherwise. S is $M[2] \oplus M[4] \oplus \cdots \oplus M[m-2] \oplus \mathsf{ozp}_n(C[m]) \oplus Z$.

$(\mathsf{b}, \nu, N, B, i) \in \mathcal{G}$ with $\mathsf{b} = \mathsf{C}$ and $\nu \in \{1, 2\}$, nor $\mathsf{b} = \mathsf{H}$. The constraint can be described by seeing that $\mathcal{B}$ has access to three oracles, which are $(\widehat{\mathsf{E}}_{\mathsf{E}}^{\mathsf{C},\cdot,\cdot,\cdot,\cdot}(\cdot), \widehat{\mathsf{D}}_{\mathsf{E}}^{\mathsf{C},0,\cdot,\cdot,\cdot}(\cdot), \widehat{\mathsf{E}}_{\mathsf{E}}^{\mathsf{H},\cdot,0^n,\cdot,\cdot}(\cdot))$ for $\mathsf{E} \xleftarrow{\$} \text{Perm}(\mathcal{W}, n)$, or $(\widetilde{\mathsf{E}}^{\mathsf{C},\cdot,\cdot,\cdot,\cdot}(\cdot), \widetilde{\mathsf{D}}^{\mathsf{C},0,\cdot,\cdot,\cdot}(\cdot), \widetilde{\mathsf{E}}^{\mathsf{H},\cdot,0^n,\cdot,\cdot}(\cdot))$ for $\widetilde{\mathsf{E}} \xleftarrow{\$} \text{Perm}(\mathcal{G}, n)$. That is, $\nu = 0$ must be respected when $\mathsf{b} = \mathsf{C}$ for decryption, and $N = 0^n$ must be satisfied when

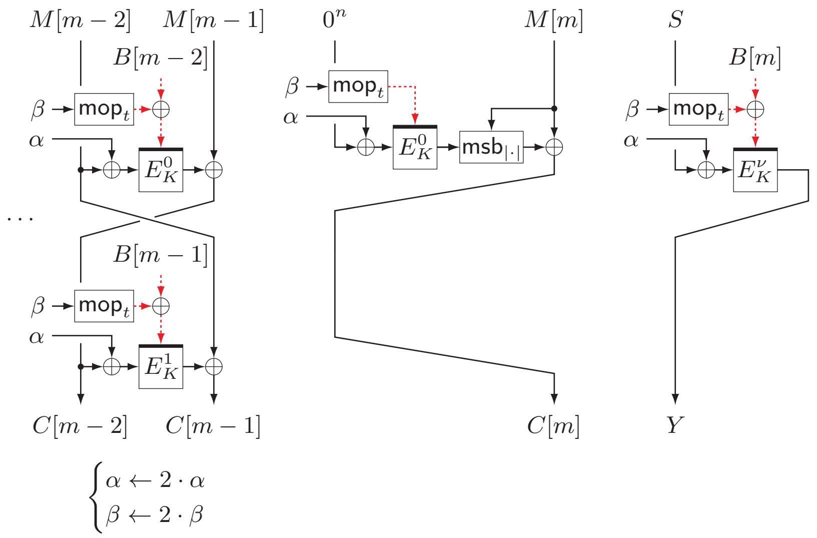


Figure 28: $\mathrm{Core.Enc}_K(N, B, M)$ of ZOTR for the case $m \bmod 2 = 1$. This processes $M[m-2], \ldots, M[m]$ and $B[m-2], \ldots, B[m]$, where $(B[1] \ldots, B[m]) \xleftarrow{t} B$. ν in the last step is 4 if $|M[m]| \neq n$ and 5 otherwise. S is $M[2] \oplus M[4] \oplus \cdots \oplus M[m-1] \oplus \mathsf{ozp}_n(M[m])$.

$\mathsf{b} = \mathsf{H}$ from the definition of $\mathcal{G}$. The TSPRP* notion is formalized as

$$\mathbf{Adv}^{\mathrm{tsprp}^*}_{\widehat{\mathsf{E}}}(\mathcal{B}) \stackrel{\mathrm{def}}{=} \Pr\left[\mathcal{B}^{\widehat{\mathsf{E}}_{\mathsf{E}}^{\mathsf{C},\cdot,\cdot,\cdot,\cdot}(\cdot),\widehat{\mathsf{D}}_{\mathsf{E}}^{\mathsf{C},0,\cdot,\cdot,\cdot}(\cdot),\widehat{\mathsf{E}}_{\mathsf{E}}^{\mathsf{H},\cdot,0^n,\cdot,\cdot}(\cdot)} \Rightarrow 1\right]$$
$$- \Pr\left[\mathcal{B}^{\widetilde{\mathsf{E}}^{\mathsf{C},\cdot,\cdot,\cdot,\cdot}(\cdot),\widetilde{\mathsf{D}}^{\mathsf{C},0,\cdot,\cdot,\cdot}(\cdot),\widetilde{\mathsf{E}}^{\mathsf{H},\cdot,0^n,\cdot,\cdot}(\cdot)} \Rightarrow 1\right],$$

where the first probability is over $\mathsf{E} \xleftarrow{\$} \mathrm{Perm}(\mathcal{W}, n)$ and $\mathcal{B}$, and the last is over $\widetilde{\mathsf{E}} \xleftarrow{\$} \mathrm{Perm}(\mathcal{G}, n)$ and $\mathcal{B}$.

The TPRP* notion captures the indistinguishability of $\widehat{\mathsf{E}}$ from $\mathrm{Perm}(\mathcal{G}, n)$ without decryption, and is formalized as

$$\mathbf{Adv}^{\mathrm{tprp}^*}_{\widehat{\mathsf{E}}}(\mathcal{B}) \stackrel{\mathrm{def}}{=} \Pr\left[\mathcal{B}^{\widehat{\mathsf{E}}_{\mathsf{E}}^{\mathsf{C},\cdot,\cdot,\cdot,\cdot}(\cdot),\widehat{\mathsf{E}}_{\mathsf{E}}^{\mathsf{H},\cdot,0^n,\cdot,\cdot}(\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{B}^{\widetilde{\mathsf{E}}^{\mathsf{C},\cdot,\cdot,\cdot,\cdot}(\cdot),\widetilde{\mathsf{E}}^{\mathsf{H},\cdot,0^n,\cdot,\cdot}(\cdot)} \Rightarrow 1\right], \quad (17)$$

where the probabilities are defined in an obvious way. We have the following result about the security of $\widehat{\mathsf{E}}$.

Proposition 1. *For any TSPRP* adversary $\mathcal{B}$ that makes q queries in total, we have $\mathbf{Adv}^{\mathrm{tsprp}^*}_{\widehat{\mathsf{E}}}(\mathcal{B}) \leq 4q^2/2^{n+\min\{n,t\}}$. Furthermore, for any TPRP* adversary $\mathcal{B}$ that makes q queries in total, we have $\mathbf{Adv}^{\mathrm{tprp}^*}_{\widehat{\mathsf{E}}}(\mathcal{B}) \leq 4q^2/2^{n+\min\{n,t\}}$.*

In Appendix A, we introduce a tweak extension scheme called XTX^*, and $\widehat{\mathsf{E}}$ can be seen as an instance of XTX^*. We present the analysis of XTX^*, from which Proposition 1 directly follows.

Completing the Proof. From the definitions of $\mathrm{ZOCB}[\mathrm{Perm}(\mathcal{W}, n)]$ and $\mathsf{iZOCB}[\widehat{\mathsf{E}}]$, it is easy to verify the following proposition.

Proposition 2. $\mathrm{ZOCB}[\mathrm{Perm}(\mathcal{W}, n)]$ *is equivalent to* $\mathsf{iZOCB}[\widehat{\mathsf{E}}]$.

Now, let $\mathcal{A}$ be a privacy adversary against $\mathrm{ZOCB}[\mathrm{Perm}(\mathcal{W}, n)]$. Given $\mathcal{A}$, we can construct a TPRP* adversary $\mathcal{B}$ against $\widehat{\mathsf{E}}$ by following the definition of $\mathsf{iZOCB}[\mathrm{Perm}(\mathcal{G}, n)]$ in Fig. 3 and by replacing the calls to $\widetilde{\mathsf{E}}$ by its corresponding oracle calls.

We see that when $\mathcal{B}$ has access to $\widehat{\mathsf{E}}$, then $\mathcal{A}$ is given access to $\mathsf{iZOCB}[\widehat{\mathsf{E}}]$, which is equivalent to $\mathrm{ZOCB}[\mathrm{Perm}(\mathcal{W},n)]$. When $\mathcal{B}$ has access to $\mathrm{Perm}(\mathcal{G},n)$, then $\mathcal{A}$ is given access to $\mathsf{iZOCB}[\mathrm{Perm}(\mathcal{G},n)]$. Since $\mathcal{B}$ makes at most σ_{priv} queries, we obtain

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{ZOCB}[\mathrm{Perm}(\mathcal{W},n)]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{tprp}^*}_{\widehat{\mathsf{E}}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{priv}}_{\mathsf{iZOCB}[\mathrm{Perm}(\mathcal{G},n)]}(\mathcal{A}) \le \frac{4\sigma^2_{\mathrm{priv}}}{2^{n+\min\{n,t\}}}$$

from Proposition 1 and Theorem 1.

For an authenticity adversary $\mathcal{A}$, we similarly have

$$\begin{aligned}\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{ZOCB}[\mathrm{Perm}(\mathcal{W},n)]}(\mathcal{A}) &\le \mathbf{Adv}^{\mathrm{tsprp}^*}_{\widehat{\mathsf{E}}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{auth}}_{\mathsf{iZOCB}[\mathrm{Perm}(\mathcal{G},n)]}(\mathcal{A})\\ &\le \frac{4\sigma^2_{\mathrm{auth}}}{2^{n+\min\{n,t\}}} + \frac{4q'}{2^n}\,,\end{aligned}$$

since the TSPRP* adversary $\mathcal{B}$ makes at most σ_{auth} queries. Note that the decryption is only needed for the case of (10), and we fix $N = 0^n$ for the case of (12). We see that $\mathcal{B}$ can indeed simulate the oracles of $\mathcal{A}$.

7.3 Security of ZOTR

The security bounds of ZOTR are similar to those of ZOCB. The main difference is that, in the computational setting, TPRP security is sufficient for authenticity of ZOTR, while we need TSPRP security for authenticity of ZOCB.

Privacy Theorem. For a privacy adversary $\mathcal{A}$ against ZOTR, we use the same definition of σ_{priv} of ZOCB. The information-theoretic bound is as follows. A proof is in Sect. 7.4.

Theorem 5. *Let $\mathcal{A}$ be a privacy adversary against* $\mathrm{ZOTR}[\mathrm{Perm}(\mathcal{W},n)]$ *that makes at most q queries with the query complexity at most σ_{priv}. Then we have* $\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{ZOTR}[\mathrm{Perm}(\mathcal{W},n)]}(\mathcal{A}) \le 4\sigma^2_{\mathrm{priv}}/2^{n+\min\{n,t\}}$.

Authenticity Theorem. For an authenticity adversary $\mathcal{A}$ against ZOTR, we also use the same definition of σ_{auth} of ZOCB. We have the following information theoretic result for the authenticity of ZOTR. A proof is in Sect. 7.4.

Theorem 6. *Let $\mathcal{A}$ be an authenticity adversary against* $\mathrm{ZOTR}[\mathrm{Perm}(\mathcal{W},n)]$ *that makes at most q encryption queries and at most q' decryption queries, where the query complexity is at most σ_{auth}. Then we have* $\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{ZOTR}[\mathrm{Perm}(\mathcal{W},n)]}(\mathcal{A}) \le 4\sigma^2_{\mathrm{auth}}/2^{n+\min\{n,t\}} + 6q'/2^n$.

For these results, we have the same remarks as in Sect. 7.1 for derivations of the computational counterparts, except that the underlying TBC is only required to be TPRP secure for both privacy and authenticity. See e.g., [BKR00] for the treatment of the computational setting.

7.4 Proofs of Theorems 5 and 6

Overview. The security proof of ZOTR has the same structure as that of ZOCB. For the TBC $\widehat{\mathsf{E}}$ with global tweak space $\mathcal{G}$ define from $\mathsf{E} \in \mathrm{Perm}(\mathcal{W},n)$ in Sect. 6, we show that it is secure in the TPRP* notion customized for ZOTR (Proposition 3). Theorem 2 shows that iZOTR is a secure NAE scheme, and we see that the adversary in the TPRP* notion can simulate ZOTR (in the real world) or iZOTR (in the ideal world), and Theorems 5 and 6 follow.

TPRP* Notion. Consider the TBC $\widehat{\mathsf{E}} : \mathcal{K}_{\widehat{\mathsf{E}}} \times \mathcal{G} \times \{0,1\}^n \to \{0,1\}^n$ defined in Sect. 6, and the corresponding TURP $\widetilde{\mathsf{E}} \in \mathrm{Perm}(\mathcal{G}, n)$. We formalize the TPRP* notion of $\widehat{\mathsf{E}}$ defined as

$$\mathbf{Adv}^{\mathrm{tprp}^*}_{\widehat{\mathsf{E}}}(\mathcal{B}) \overset{\mathrm{def}}{=} \Pr\left[\mathcal{B}^{\widehat{\mathsf{E}}_{\mathsf{E}}^{\mathsf{C},\cdot,\cdot,\cdot,\cdot,\cdot}(\cdot),\widehat{\mathsf{E}}_{\mathsf{E}}^{\mathsf{H},\cdot,\cdot,0^n,\cdot,\cdot}(\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{B}^{\widetilde{\mathsf{E}}^{\mathsf{C},\cdot,\cdot,\cdot,\cdot,\cdot}(\cdot),\widetilde{\mathsf{E}}^{\mathsf{H},\cdot,\cdot,0^n,\cdot,\cdot}(\cdot)} \Rightarrow 1\right],$$

where the probabilities are defined in an obvious way. Note that this is different from (17) in the definition of $\mathcal{G}$. We have the following result about the security of $\widehat{\mathsf{E}}$.

Proposition 3. *For any TPRP* adversary $\mathcal{B}$ that makes q queries in total, we have* $\mathbf{Adv}^{\mathrm{tprp}^*}_{\widehat{\mathsf{E}}}(\mathcal{B}) \le 4q^2/2^{n+\min\{n,t\}}$.

We present the analysis of a more general case of XTX^* in Appendix A, and Proposition 3 follows.

Completing the Proof. The following proposition immediately follows from the definitions.

Proposition 4. ZOTR[Perm($\mathcal{W}, n$)] *is equivalent to* $\mathsf{iZOTR}[\widehat{\mathsf{E}}]$.

Given a privacy or authenticity adversary $\mathcal{A}$ against ZOTR[Perm($\mathcal{W}, n$)], we can construct a TPRP* adversary $\mathcal{B}$ against $\widehat{\mathsf{E}}$ by following the definition of iZOTR[Perm($\mathcal{G}, n$)] and by replacing the calls to $\widetilde{\mathsf{E}}$ by its corresponding oracle calls. We see that TPRP* adversary $\mathcal{B}$ against $\widehat{\mathsf{E}}$ can simulate $\mathsf{iZOTR}[\widehat{\mathsf{E}}]$, which is equivalent to ZOTR[Perm($\mathcal{W}, n$)], or iZOTR[Perm($\mathcal{G}, n$)].

For a privacy adversary $\mathcal{A}$, since $\mathcal{B}$ makes at most σ_{priv} queries, we obtain

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{ZOTR}[\mathrm{Perm}(\mathcal{W},n)]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{tprp}^*}_{\widehat{\mathsf{E}}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{priv}}_{\mathsf{iZOTR}[\mathrm{Perm}(\mathcal{G},n)]}(\mathcal{A}) \le \frac{4\sigma^2_{\mathrm{priv}}}{2^{n+\min\{n,t\}}}$$

from Proposition 3 and Theorem 2.

For an authenticity adversary $\mathcal{A}$, $\mathcal{B}$ makes at most σ_{auth} queries, and

$$\begin{aligned}\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{ZOTR}[\mathrm{Perm}(\mathcal{W},n)]}(\mathcal{A}) &\le \mathbf{Adv}^{\mathrm{tprp}^*}_{\widehat{\mathsf{E}}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{auth}}_{\mathsf{iZOTR}[\mathrm{Perm}(\mathcal{G},n)]}(\mathcal{A}) \\ &\le \frac{4\sigma^2_{\mathrm{auth}}}{2^{n+\min\{n,t\}}} + \frac{6q'}{2^n}\end{aligned}$$

holds from Proposition 3 and Theorem 2.

8 Instantiation and Implementation

8.1 Instantiation

Instantiation with TAES. AES is naturally the first cipher we would like to instantiate under ZOCB and ZOTR. However, it is a plain blockcipher that does not take tweaks as input. We observe AES-256 has a key length of 256 bits, while 128-bit security is sufficient for most of real-world applications. Hence we define Tweakable AES (TAES for short), by inputting the tweak as the second half of the master key. We avoid using the first half as the tweak because the first half is used as pre-whitening key of AES-256, which leaves the first round of AES-256 offering no security when the tweak is known. TAES immediately results in a TBC of 128-bit keys and 128-bit tweaks, whose software performance enjoys the fast AES New Instructions (AES-NI) and security inherits directly from the intensive cryptanalysis against AES in the last 20 years. To instantiate ZOCB and ZOTR with TAES, the first byte (8 bits) of TAES's tweak is reserved to encode the elements of $\mathcal{I}$, and the remaining 120 bits are for the effective tweak inputs of ZOCB and ZOTR. Hence, both TAES-ZOCB and TAES-ZOTR are of key and effective tweak lengths of 128 and 120 bits.

We claim 128-bit security of TAES under the single-key and chosen-tweak setting, i.e., both value and difference of the tweak can be chosen freely by attackers while the key is kept the same and secret. One might have the concern that the security of AES-256 is violable to the related-key attack due to Biryukov and Khovratovich [BK09]. Note the related-key attack only works with differences in both halves of the 256-bit key, and the attackers choose differences in round-keys, rather than that in the master key. This attack might still work under the setting of related key-and-tweak (i.e., non-zero differences in both key and tweak) for TAES. However, the proof of ZOCB and ZOTR only concerns the security under single-key. Under this constraint, the attackers can only choose differences in the second half of the master key where the tweak lies, rather than round keys. This requirement contradicts the attack settings in the Biryukov-Khovratovich attack, hence this attack as in the current form is not applicable to TAES. Yet, dedicated cryptanalysis of TAES is required to gain confidence in its security, which we leave as an open problem.

Instantiation with SKINNY-128-256. SKINNY is a family of TBCs proposed by Beierle et al. at CRYPTO 2016 [BJK$^+$16]. The family consists of a set of versions for different block sizes and tweakey sizes (SKINNY does not differentiate the tweak and key, and calls the combined input tweakey). It fits exactly the usecase of ZOCB and ZOTR as the tweakey space is large. To demonstrate the effectiveness of our design, we choose SKINNY-128-256 with block size 128 bits and tweakey size 256 bits to be instantiated under ZOCB and ZOTR.

Instantiation with Other TBCs. ZOCB and ZOTR are modes of operation for TBCs, and any reasonable TBCs can be used to instantiate our schemes. For instance KIASU-BC [JNP14c] or Deoxys-BC [JNP14a] can be used, and we chose TAES and SKINNY-128-256 as the tweak length of KIASU-BC does not best suit for our schemes, TAES can fully benefit from the efficiency of AES-NI, and the security of SKINNY-128-256 is extensively analyzed [SKI]. Given that Deoxys-II was selected as the final portfolio of CAESAR [CAE], Deoxys-BC is certainly another reasonable option to instantiate our schemes. See e.g. [CHP$^+$17, MMS18, CHP$^+$18, Sas18] for cryptanalytic results on Deoxys-BC.

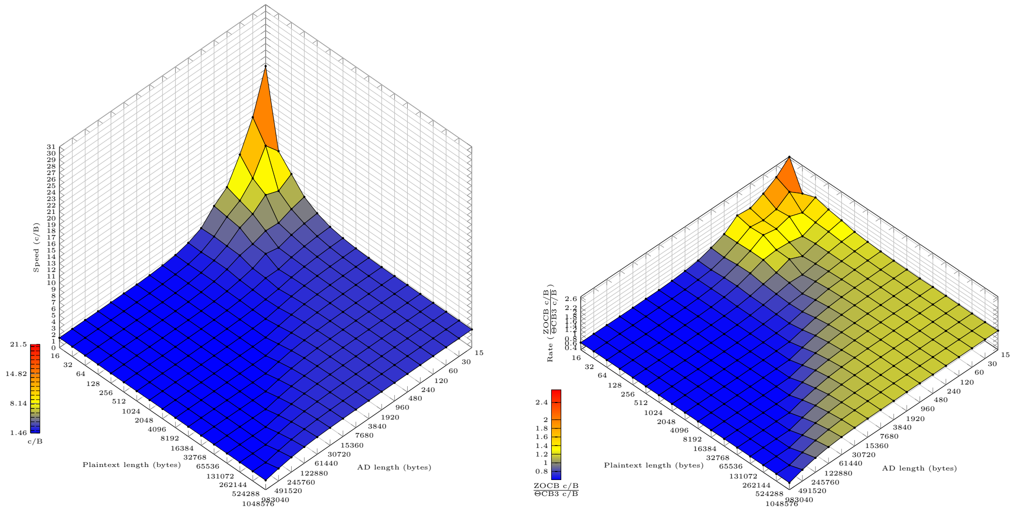
8.2 Implementation Results

Implementation with TAES. To benefit from the efficiency of AES-NI, the speed test of TAES implementation is carried out on an Intel Core i5-6500 CPU clocked at 3.20 GHz (Skylake family). Although the encryption runs as fast as 0.65 cycles per byte (c/B) for long plaintexts when the key schedule is not counted (i.e., all round keys are precomputed and stored somewhere), it is noted that the key schedule can be slower than encryption itself. Gueron et al. [GLNP15] developed a dedicated key schedule implementation using SSE instructions which is faster than the standard implementation based on `aeskeygenassist` instruction of AES-NI. In order to take advantage of this fast key schedule and not to waste the limited number of registers storing all the round keys, we interleave the key schedule with the `aesenc` instruction, which performs AES round function. As `aesenc` takes 4 cycles, we set the parallelism to 4 to achieve the best possible performance. Our implementation also confirms that 4 is the best choice for achieving the fastest speed.

To compare the efficiency to process plaintexts and AD of various lengths, we implemented ΘCB3 instantiated with TAES, where 8 bits of the tweak is to encode $\mathcal{I}$ (ΘCB3 needs a few bits for $\mathcal{I}$), 56 bits for the block counter, and 64 bits for the nonce[8].

[8] This is because the tweak space of the TBC inside ΘCB3 exceeds n bits if the nonce length is n bits,

Implementation with SKINNY-128-256. For ease of notation, we will use SKINNY for short here to represent the specific instance SKINNY-128-256. The best performance of SKINNY is achieved by bit-sliced implementation using AVX2 registers and parallelism set to be 64 [Köl17], and we are to follow the same when it is instantiated under ZOCB and ZOTR. There are two 128-bit chunks of the 256-bit tweakey named TK1 and TK2. Since SKINNY does not differentiate the tweak and key, we have the choice of using either TK1 or TK2 as the secret key when instantiated under ZOCB and ZOTR. We observe the round key contribution from TK2 follows a cycle of length 16, i.e., round-key contribution from TK2 is the same for round $16 \cdot i + j$, for $i = 0, 1, 2$ and any fixed j in the range $0 \leq j < 16$. For better performance, we choose TK2 to be the secret key, precompute its contribution in the first 16 rounds, store them in registers, and re-use for the subsequent 32 rounds. This is consistent with the recommendation in [BJK+16], where using TK1 for processing tweaks is recommended. Hence, the key schedule for processing the secret key is almost free in computation for long data. However, because the bit-sliced implementation only supports processing 64 blocks of data in parallel and there is no optimized implementation available for processing short data, in our implementation, short inputs are processed using the encryption function for a single block, which follows the reference code of SKINNY. Due to this differentiation between data of different lengths, a significant gap appears between the performance with data of less than 64 (2×64 for ZOTR) blocks and that with data of more than 64 (2×64 for ZOTR) blocks (hundreds of cycles per byte vs. several cycles per byte). As a result, for SKINNY-ZOCB/ZOTR/ΘCB3, the graphs of c/B as a function of input lengths lose lots of information. Thus, we merely provide the raw data of the experimental results in Tables 4–5 in Appendix F for comparison.



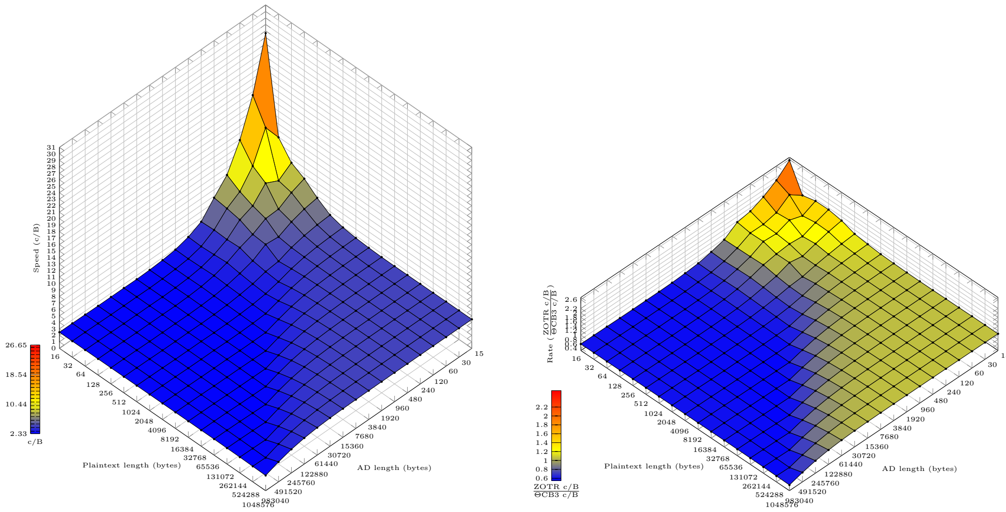
(a) Absolute

(b) Relative

Figure 29: Speed of TAES-ZOCB on Skylake. **(a)** Absolute speed in c/B. **(b)** Relative to TAES-ΘCB3. Corresponding graphs for TAES-ZOTR on Skylake are very similar.

Timing Method. We timed ZOCB/ZOTR/ΘCB3 instantiated with TAES and SKINNY on CPUs of two families. One is an Intel(R) Core(R) i5-6500 CPU clocked at 3.20 GHz (Skylake family), with 32 (or 256, 6144) KB L1 (resp. L2, L3) cache, with Ubuntu 16.04.3

where $n = 128$ in our case. We could use n-bit nonces by using (a simple variant of) XTX^* applied to TAES instead of raw TAES, but this slightly increases the computation and memory. Hence, our setting only gains the performance of TAES-ΘCB3.



(a) Absolute **(b)** Relative

Figure 30: Speed of TAES-ZOTR on Haswell. **(a)** Absolute speed in c/B. **(b)** Relative to TAES-ΘCB3. Corresponding graphs for TAES-ZOCB on Haswell are very similar.

LTS OS, g++ (GNU/gcc) compiler of version 5.4.0, and icpc (Intel/icc) compiler of version 19.0.1. Another is an Intel(R) Xeon(R) E5-2603 v3 CPU clocked at 1.60 GHz (Haswell family), with 32 (or 256, 15360) KB L1 (resp. L2, L3) cache, with Ubuntu 14.04.5 LTS OS, g++ (GNU/gcc) compiler of version 4.8.5, and icpc (Intel/icc) compiler of version 19.0.1. We used `-O3` option to compile the programs. During the timing, we turned off hyper-threading and disabled Turbo Boost. As the timing method, we used Brain Gladman's program for timing the implementations of AES[9]. The instruction used is `rdtsc`. We used the function `e_cycles()` which uses 100 loops to evaluate the entire time of eight repetitions of data processing of particular lengths. Concretely, the data consists of plaintexts (M) and associated data (AD), where plaintexts are of length $|\mathrm{M}| \in \{2^4, 2^5, \ldots, 2^{20}\}$ bytes and AD are of length $|\mathrm{AD}| \in (15/16) \times \{2^4, 2^5, \ldots, 2^{20}\}$ bytes. Note here one block of a plaintext is of 128 bits, that of AD is of 120 bits, and we only time the processing of data of length divisible by the block sizes.

Observations on Performance Evolves with the Length of the Inputs. Figures 29a and 30a are graphs of absolute values for c/B as functions of input lengths. Figures 29b and 30b are graphs of relative values for c/B (i.e., the c/B of TAES-ZOCB and TAES-ZOTR divided by c/B of TAES-ΘCB3). The raw data generating those figures are reported in Tables 2–3 in Appendix F. From these figures and the raw data, we have the following observations.

- For short input data such that $|\mathrm{AD}| \lesssim 480$ bytes or $|\mathrm{AD}|/|\mathrm{M}| \lesssim 0.12$, TAES-ZOCB and TAES-ZOTR are not (always) as fast as TAES-ΘCB3.
- For long input data such that $|\mathrm{AD}| \gtrsim 480$ bytes and $|\mathrm{AD}|/|\mathrm{M}| \gtrsim 0.12$, TAES-ZOCB and TAES-ZOTR perform better than TAES-ΘCB3.
- Asymptotically with long data, for the case where $|\mathrm{AD}|/|\mathrm{M}| \gtrsim 0.12$, the performance gain of TAES-ZOCB/ZOTR is about 40% (i.e., $\frac{\mathrm{c/B(TAES\text{-}\Theta CB3)} - \mathrm{c/B(TAES\text{-}ZOCB)}}{\mathrm{c/B(TAES\text{-}\Theta CB3)}} \approx$ 40%), they are about 1.7× faster than TAES-ΘCB3.

[9] Available via http://github.com/BrianGladman/AES.

- Similar observations on the performances of SKINNY-ZOCB/ZOTR/ΘCB3 can be derived from raw data presented in Tables 4–5 in Appendix F. The difference lies in the boundary between the portions of the two input spaces where ΘCB3 or ZOCB/ZOTR perform better. The boundary is closer to the small values of data length for SKINNY instances than that for TAES instances.

We expect the performance gains over long data remain roughly the same for instantiations under other TBCs as long as the block size of the underlying TBC remains unchanged.

We note that the above observations are made from the timing data of particular lengths, and behavior of other lengths can vary.

Source code, raw data, and the graphs are available at https://github.com/zocbzotr.

9 Discussions and Conclusions

In this paper, we specified ZOCB and ZOTR, and showed their provable security. These schemes simultaneously achieve full absorption and full parallelizability. We also presented experimental results showing the advantages of our schemes and clarifying the cost for it. Here, we present discussions related to our designs.

upBB Security of Components. We first point out that upBB security is enough for masks. That is, the masks α and β are generated by TBC calls taking a nonce as input (as a block), and this mask generation provides $\Theta(q^2/2^n)$-distinguishing advantage from uniform with q queries, seemingly violating our security claim. However, this is irrelevant because these masks are only required to satisfy certain independence conditions with $O(1/2^n)$ bias for any distinct *pair* of nonces.

We also remark that upBB security is enough for Hash_K used in ZOCB and ZOTR. The output length of the hash function is n bits, and this may seem to allow an attack with birthday complexity. However, it is only required to be a (computational) universal hash function of n-bit output, as with the case of ΘCB3 that uses a PMAC-style hash function to process AD.

Tags Can Be Truncated. We next note that depending on the application, n-bit tags of ZOCB and ZOTR can be safely truncated. If the tag is truncated to $1 \leq \tau \leq n$ bits, the resulting authenticity term with respect to q' will be multiplied by $2^{n-\tau}$ since the adversary just needs to guess the truncated τ bits of a random value. That is, $4q'/2^n$ in Theorem 4 is increased to $4q'/2^\tau$, and $6q'/2^n$ in Theorem 6 is increased to $6q'/2^\tau$, and they can be proved by modifying the corresponding proofs in a straightforward way. However, we note that the tag length has to be fixed during the lifetime of the key.

Using iZOCB and iZOTR. If one has a secure TBC $\widehat{\mathsf{E}} : \mathcal{K}_{\widehat{\mathsf{E}}} \times \mathcal{G} \times \{0,1\}^n \to \{0,1\}^n$ as a primitive with a long tweak input, where $\mathcal{G}$ is defined as in (1) or (2), then it is tempting to practically use iZOCB and iZOTR. They both have perfect security in privacy and n-bit security in authenticity, just as like ΘCB3 and $\mathbb{OTR}$. However, even with such $\widehat{\mathsf{E}}$, using it as E in ZOCB and ZOTR improves the efficiency as they can process longer AD per one primitive call, and hence iZOCB and iZOTR are conceptual schemes to highlight the feasibility of the block-by-block processing of AD and the AD-independent tag approach, and for provable security rather than schemes for real applications.

Tightness of the Security Bounds. It is easy to see that the security bounds of iZOCB and iZOTR (Theorems 1 and 2) are tight, i.e., there is an attack that matches the corresponding security bound. We also see that the security bounds of $\widehat{\mathsf{E}}$ in Propositions 1 and 3 are tight, since a simultaneous collision of the input block and input tweak of E

reveals the masks. However, we do not know the tightness of the security bounds of ZOCB and ZOTR (Theorems 3, 4, 5, and 6).

Switching-Off the Doubling. We next discuss a possible way to extend our schemes. In the current specifications of ZOCB and ZOTR, the doubling operations for α and β are always performed m times for m-block plaintexts no matter how short AD is. While the computational cost of doubling is not large in general, this may still cause slight speed degradation when AD is short and the plaintext is long, and the TBC is ultimately fast. The case is visible from the experimental results in Sect. 8. In order to mitigate the issue, as a possible direction to further improve the efficiency, one could derive an extension of our schemes that stops doubling operations for α and β immediately after the end of AD. That is, the doubling operations for α and β are performed exactly a times for at-bit AD with $a < m$. For this extension, one needs to add one more element to $\mathcal{I}$ to indicate the presence of an AD block in the encryption/decryption core, and add a plain block counter in the space that was used for the empty AD blocks. This extension adds complexity to the specification and the proof, and we leave it as a future work to specify the details of this approach and the analysis of it.

Further Future Work. While we used TAES and SKINNY-128-256 in our experiments, ZOCB and ZOTR benefit if the underlying TBC has large tweak space, and if the tweak can be updated efficiently. A TBC with the same characteristic also improves ZMAC and ZAE. A dedicated design of such TBCs remains to be explored. We emphasize that detailed security analysis of TAES remains, and we recommend its use only after a sufficient amount of analysis confirms it security.

In this paper, we use the t-bit tweak input space of the TBC to process a block of AD, while we use the n-bit input space to process a block of plaintext. Our schemes achieve full absorption and allow fully parallel computation of the underlying TBC, and they have the unique design feature in the tag generation, where they demonstrate that AD-independent tags can provide authenticity. We believe that our approach adds design space to various TBC modes, and has wide applications related to TBC-based constructions. For instance it could be used to design efficient tweakable enciphering schemes [HR03, HR04] with large tweak space from a TBC, it could be used to obtain efficient robust AE schemes [HKR15], or it could be used to improve the efficiency of online AE schemes [AFF⁺14].

A Proofs of Propositions 1 and 3

To prove Propositions 1 and 3, we introduce a general tweak space extender called XTX^* as a generalization of XTX [MI15] and XEX* [Rog04]. We start with the definition of partition-respecting TSPRP adversaries.

Definition 1. Let $\mathcal{G}$ be finite tweak space, and let $E : \mathcal{K}_E \times \mathcal{G} \times \{0,1\}^n \to \{0,1\}^n$ be a TBC, where $\mathcal{K}_E$ is the key space. Suppose that there is a partition of $\mathcal{G}$ into two sets, the enc-dec set $\mathcal{G}_{\mathsf{ed}}$ and enc-only set $\mathcal{G}_{\mathsf{eo}}$. If an adversary $\mathcal{A}$ against E has access to both E_K and D_K with any tweak in $\mathcal{G}_{\mathsf{ed}}$ but has access to only E_K with any tweak in $\mathcal{G}_{\mathsf{eo}}$, then we say that $\mathcal{A}$ is a partition-respecting TSPRP adversary with respect to $\mathcal{G}_{\mathsf{ed}}$ and $\mathcal{G}_{\mathsf{eo}}$. If the partition is clear from the context, we simply say that $\mathcal{A}$ is partition-respecting.

We next introduce the tweak space extender XTX^*. Fix the tweak space $\mathcal{G}$, where $\mathcal{G}$ can be partitioned into enc-dec set $\mathcal{G}_{\mathsf{ed}}$ and enc-only set $\mathcal{G}_{\mathsf{eo}}$. Let $\mathsf{E} \in \mathrm{Perm}(\mathcal{W}, n)$ be a TURP for some $\mathcal{W}$, and $H : \mathcal{L} \times \mathcal{G} \to \{0,1\}^n \times \mathcal{W}$ be a keyed hash function, where $\mathcal{L}$ is the key space of H and $L \in \mathcal{L}$ is the key.

$\mathsf{XTX}^*[\mathrm{Perm}(\mathcal{W}, n), H]$ is a TBC that is parametrized by $\mathrm{Perm}(\mathcal{W}, n)$ and H, and has encryption algorithm $\mathsf{XTX.Enc}^* : \mathcal{K}_{\mathsf{XTX}^*} \times \mathcal{G} \times \{0,1\}^n \to \{0,1\}^n$ and decryption algorithm

$\mathsf{XTX.Dec}^* : \mathcal{K}_{\mathsf{XTX}^*} \times \mathcal{G} \times \{0,1\}^n \to \{0,1\}^n$. The key space is $\mathcal{K}_{\mathsf{XTX}^*} = \mathrm{Perm}(\mathcal{W}, n) \times \mathcal{L}$ and they take $\mathsf{E} \in \mathrm{Perm}(\mathcal{W}, n)$ and $L \in \mathcal{L}$ as the key. The encryption of $M \in \{0,1\}^n$ under the tweak $G \in \mathcal{G}$ and key $(\mathsf{E}, L) \in \mathcal{K}_{\mathsf{XTX}^*}$ is defined as

$$\begin{cases} C = \mathsf{XTX.Enc}^*{}_{\mathsf{E},L}(G, M) = \mathsf{E}(U, M \oplus V) \oplus V & \text{if } G \in \mathcal{G}_{\mathsf{ed}}\,, \\ C = \mathsf{XTX.Enc}^*{}_{\mathsf{E},L}(G, M) = \mathsf{E}(U, M \oplus V) & \text{if } G \in \mathcal{G}_{\mathsf{eo}}\,, \end{cases}$$

where $H_L(G) = (V, U) \in \{0,1\}^n \times \mathcal{W}$. The decryption of $C \in \{0,1\}^n$ is defined by using the decryption D of E as

$$\begin{cases} M = \mathsf{XTX.Dec}^*{}_{\mathsf{E},L}(G, C) = \mathsf{D}(U, C \oplus V) \oplus V & \text{if } G \in \mathcal{G}_{\mathsf{ed}}\,, \\ M = \mathsf{XTX.Dec}^*{}_{\mathsf{E},L}(G, C) = \mathsf{D}(U, C) \oplus V & \text{if } G \in \mathcal{G}_{\mathsf{eo}}\,. \end{cases}$$

A tweak of $\mathsf{XTX}^*[\mathrm{Perm}(\mathcal{W}, n), H]$ is called a global tweak, and a tweak of E may be called an internal tweak. We next define two properties needed for H.

Definition 2. Let $H : \mathcal{L} \times \mathcal{G} \to \{0,1\}^n \times \mathcal{W}$ be a keyed hash function. If

$$\max_{c \in \{0,1\}^n} \Pr[L \xleftarrow{\$} \mathcal{L} : V \oplus V' = c, U = U'] \le \epsilon \tag{18}$$

holds for any distinct $G, G' \in \mathcal{G}$, where $H_L(G) = (V, U)$ and $H_L(G') = (V', U')$, then H is $(n, \mathcal{W}, \epsilon)$-partial almost XOR universal, or $(n, \mathcal{W}, \epsilon)$-pAXU. If

$$\max_{c \in \{0,1\}^n} \Pr[L \xleftarrow{\$} \mathcal{L} : V = c, U = U'] \le \epsilon \tag{19}$$

holds for any distinct $G, G' \in \mathcal{G}$, then H is $(n, \mathcal{W}, \epsilon)$-partial uniform, or $(n, \mathcal{W}, \epsilon)$-pU.

We remark that $(n, \mathcal{W}, \epsilon)$-pAXU was previously defined for XTX [MI15].

The following lemma shows that XTX^* is an information-theoretic TSPRP-secure TBC for any partition-respecting TSPRP adversary. A proof is in Appendix B.

Lemma 1. *Consider* $\mathsf{XTX}^*[\mathrm{Perm}(\mathcal{W}, n), H]$*, and let* $\mathcal{A}$ *be a partition-respecting TSPRP adversary that makes* q *queries in total. If* H *is* $(n, \mathcal{W}, \epsilon)$*-pAXU and* $(n, \mathcal{W}, \epsilon)$*-pU, then we have* $\mathbf{Adv}^{\mathrm{tsprp}}_{\mathsf{XTX}^*[\mathrm{Perm}(\mathcal{W},n),H]}(\mathcal{A}) \le \epsilon q^2$.

We next observe that $\widehat{\mathsf{E}}$ defined in Sect. 7.2 for ZOCB is an instance of XTX^*. Specifically, for $\mathcal{G}$ defined in (1), let $H : \mathcal{L} \times \mathcal{G} \to \{0,1\}^n \times \mathcal{W}$ be defined as $H_L(\mathsf{b}, \nu, N, B, i) = (V, U)$, where

$$V = \begin{cases} 2^i \alpha & \text{if } \mathsf{b} = \mathsf{C}\,, \\ 2^i \gamma & \text{if } \mathsf{b} = \mathsf{H}\,, \end{cases} \tag{20}$$

$$U = \begin{cases} (\nu, \mathsf{mop}_t(2^i \beta) \oplus B) & \text{if } \mathsf{b} = \mathsf{C}\,, \\ (\nu, \mathsf{mop}_t(2^i \delta) \oplus B) & \text{if } \mathsf{b} = \mathsf{H}\,. \end{cases} \tag{21}$$

The key space $\mathcal{L}$ consists of all functions of $\mathsf{E}^{3,\cdot}(\cdot)$, and α, β, γ, and δ are defined as in (13). The tweak partition is defined as $\mathcal{G}_{\mathsf{ed}} = \{(\mathsf{b}, \nu, N, B, i) \in \mathcal{G} : \mathsf{b} = \mathsf{C}, \nu = 0\}$ for the enc-dec set, and $\mathcal{G}_{\mathsf{eo}} = \mathcal{G} \setminus \mathcal{G}_{\mathsf{ed}}$ for the enc-only set.

Similarly, we see that $\widehat{\mathsf{E}}$ defined in Sect. 7.4 for ZOTR is an instance of XTX^*. For $\mathcal{G}$ in (2), we define $H : \mathcal{L} \times \mathcal{G} \to \{0,1\}^n \times \mathcal{W}$ as (20) and (21), where the key space $\mathcal{L}$ consists of all functions of $\mathsf{E}^{6,\cdot}(\cdot)$, and α, β, γ, and δ are defined as in (16). The tweak partition is defined as $\mathcal{G}_{\mathsf{ed}} = \emptyset$ for the enc-dec set, and $\mathcal{G}_{\mathsf{eo}} = \mathcal{G}$ for the enc-only set.

It remains to show that the instantiation of H in (20) and (21) is $(n, \mathcal{W}, \epsilon)$-pAXU and $(n, \mathcal{W}, \epsilon)$-pU for small ϵ.

Proposition 5. *For both* ZOCB *and* ZOTR*, the instantiation of H in* (20) *and* (21) *is* $(n, \mathcal{W}, \epsilon)$*-pAXU and* $(n, \mathcal{W}, \epsilon)$*-pU for* $\epsilon = 4/2^{n+\min\{n,t\}}$.

A proof is an elementary case analysis and is presented in Appendix C. Finally, Propositions 1 and 3 follow from the fact that $\widehat{\mathsf{E}}$ is an instantiation of XTX^*, Lemma 1, and Proposition 5.

B Proof of Lemma 1

The proof is based on Patarin's H-coefficient technique [Pat08]. We follow the standard terminology and techniques commonly used. For details we refer to [CS14] by Chen and Steinberger which provides a compact exposition of the H-coefficient technique.

In the real world, the adversary $\mathcal{A}$ makes queries to $\mathsf{XTX}^*[\mathrm{Perm}(\mathcal{W}, n), H]$, which we abbreviate as XTX^* throughout the proof. In the ideal world, $\mathcal{A}$ makes queries to a TURP $\widetilde{\mathsf{E}} \in \mathrm{Perm}(\mathcal{G}, n)$. Here, $\mathcal{A}$ is a partition-respecting TSPRP adversary, and we assume that it makes q_{ed} queries that have tweaks in $\mathcal{G}_{\mathsf{ed}}$, and q_{eo} queries that have tweaks in $\mathcal{G}_{\mathsf{eo}}$. Thus $q = q_{\mathsf{ed}} + q_{\mathsf{eo}}$. We remark that q is a parameter of $\mathcal{A}$ but q_{ed} and q_{eo} are not static and possibly adaptively determined by $\mathcal{A}$.

We index the queries (and all internal variables) from 1 to q with no distinction on the tweaks. Let $G_i \in \mathcal{G}$ be the (global) tweak used in the i-th query and let $(M_i, C_i) \in \{0,1\}^n \times \{0,1\}^n$ be the plaintext and ciphertext blocks. We do not distinguish the direction of queries, hence if the i-th query is an encryption query, (G_i, M_i) is queried and C_i is a response, and if the i-th query is a decryption query, (G_i, C_i) is queried and M_i is a response. As explained, when $G_i \in \mathcal{G}_{\mathsf{eo}}$, it must be an encryption query. We define $X_i = V_i \oplus M_i$ and $Y_i = \widehat{V}_i \oplus C_i$, where $\widehat{V}_i = V_i$ when $G_i \in \mathcal{G}_{\mathsf{ed}}$, and $\widehat{V}_i = 0^n$ when $G_i \in \mathcal{G}_{\mathsf{eo}}$.

We assume that $\widetilde{\mathsf{E}}$ in the ideal world also internally invokes $H_L(G_i)$ to derive V_i and U_i, so that X_i and Y_i are determined. Note that they are dummy variables to simplify the analysis and do not affect the outputs.

Without loss of generality, we assume that $\mathcal{A}$ is deterministic and does not make repeated queries nor reversed queries (i.e. encryption query (G, M) to obtain C then decryption query (G, C) to obtain M, and vice versa). From the property of a TBC, this guarantees

$$\begin{cases} (G_i, M_i) \neq (G_j, M_j) \text{ for any } 1 \leq i < j \leq q\,, \text{ and} \\ (G_i, C_i) \neq (G_j, C_j) \text{ for any } 1 \leq i < j \leq q\,. \end{cases} \tag{22}$$

It is a popular technique for proofs based on the H-coefficient technique that we relax the game such that the key L of H is given to $\mathcal{A}$ after $\mathcal{A}$ made all the queries, which allows $\mathcal{A}$ to determine all $X_1, \ldots, X_q$ and $Y_1, \ldots, Y_q$ from the transcript. This only benefits $\mathcal{A}$.

Therefore, we write the transcript as

$$\tau = ((G_1, M_1, C_1), \ldots, (G_q, M_q, C_q), L)\,.$$

Here, q_{ed} and q_{eo} are uniquely determined from τ.

Good and Bad Transcripts. As $\mathcal{A}$ is assumed to be deterministic, the probability space of the transcript (for both real and ideal worlds) solely depends on the internal TBC used in the world. We write the probability space under the real and ideal worlds as $\Pr_{\mathrm{re}}[\cdot]$ and $\Pr_{\mathrm{id}}[\cdot]$, and use $\Theta = \tau$ to denote the event that the transcript is τ. We say τ is *attainable* if $\Pr_{\mathrm{id}}[\Theta = \tau] > 0$, which is equivalent to that (22) holds true, and we only consider attainable transcripts.

We say τ is *bad* if $(X_i, U_i) = (X_j, U_j)$ or $(Y_i, U_i) = (Y_j, U_j)$ for some $1 \leq i < j \leq q$, and τ is *good* if it is not bad. Let BadT and GoodT denote the set of all bad and good transcripts. Notice that $\mathsf{BadT} \cup \mathsf{GoodT}$ is the set of all attainable transcripts.

We use the following fundamental lemma of the H-coefficient technique. See e.g. [CS14] for the proof.

Lemma 2. *If there exists ϵ_1 and ϵ_2 such that for any $\tau \in \mathsf{GoodT}$,*

$$\frac{\Pr_{\mathrm{re}}[\Theta = \tau]}{\Pr_{\mathrm{id}}[\Theta = \tau]} \geq 1 - \epsilon_1 \textit{ and } \Pr_{\mathrm{id}}[\Theta \in \mathsf{BadT}] \leq \epsilon_2 \,,$$

then $\mathcal{A}$'s advantage is bounded by $\epsilon_1 + \epsilon_2$.

We first evaluate $\Pr_{\mathrm{id}}[\Theta \in \mathsf{BadT}]$ in the following lemma.

Lemma 3. *If H is $(n, \mathcal{W}, \epsilon)$-pAXU and $(n, \mathcal{W}, \epsilon)$-pU, then* $\Pr_{\mathrm{id}}[\Theta \in \mathsf{BadT}] \leq \epsilon q^2$.

Proof. We have

$$\begin{aligned}\Pr_{\mathrm{id}}[\Theta \in \mathsf{BadT}] \leq & \Pr_{\mathrm{id}}[(X_i, U_i) = (X_j, U_j) \text{ for some } 1 \leq i < j \leq q] \\ & + \Pr_{\mathrm{id}}[(Y_i, U_i) = (Y_j, U_j) \text{ for some } 1 \leq i < j \leq q] \,. \end{aligned} \tag{23}$$

For $i \neq j$, the event $(X_i, U_i) = (X_j, U_j)$ never happens when $G_i = G_j$, since $G_i = G_j$ implies $(V_i, U_i) = (V_j, U_j)$, and $M_i \neq M_j$ (as it must be attainable) ensures $(X_i, U_i) \neq (X_j, U_j)$. If $G_i \neq G_j$, then we have

$$\begin{aligned}& \Pr_{\mathrm{id}}[(X_i, U_i) = (X_j, U_j)] \\ & = \sum_{m_i, m_j} \Pr_{\mathrm{id}}[(X_i, U_i) = (X_j, U_j) \mid M_i = m_i, M_j = m_j] \cdot \Pr_{\mathrm{id}}[M_i = m_i, M_j = m_j] \\ & \leq \max_{m_i, m_j} \Pr_{\mathrm{id}}[(X_i, U_i) = (X_j, U_j) \mid M_i = m_i, M_j = m_j] \\ & \leq \max_{m_i, m_j} \Pr_{\mathrm{id}}[V_i \oplus V_j = m_i \oplus m_j, U_i = U_j \mid M_i = m_i, M_j = m_j] \\ & \leq \max_{c} \Pr_{\mathrm{id}}[V_i \oplus V_j = c, U_i = U_j] \\ & \leq \epsilon \,,\end{aligned}$$

where the second last inequality follows from the fact that H_L does not affect the responses in the ideal world, and the last one follows from the assumption on H.

We next evaluate $\Pr_{\mathrm{id}}[(Y_i, U_i) = (Y_j, U_j)]$ in two cases. First, assuming $G_i \neq G_j$ and $G_i, G_j \in \mathcal{G}_{\mathsf{ed}}$, or $G_i \neq G_j$, $G_i \in \mathcal{G}_{\mathsf{ed}}$, and $G_j \in \mathcal{G}_{\mathsf{eo}}$, we have

$$\begin{aligned}& \Pr_{\mathrm{id}}[(Y_i, U_i) = (Y_j, U_j)] \\ & = \sum_{c_i, c_j} \Pr_{\mathrm{id}}[(Y_i, U_i) = (Y_j, U_j) \mid C_i = c_i, C_j = c_j] \cdot \Pr_{\mathrm{id}}[C_i = c_i, C_j = c_j] \\ & \leq \max_{c_i, c_j} \Pr_{\mathrm{id}}[(Y_i, U_i) = (Y_j, U_j) \mid C_i = c_i, C_j = c_j] \\ & = \max_{c_i, c_j} \Pr_{\mathrm{id}}[\widehat{V}_i \oplus \widehat{V}_j = c_i \oplus c_j, U_i = U_j \mid C_i = c_i, C_j = c_j] \\ & \leq \max_{c_i, c_j} \Pr_{\mathrm{id}}[\widehat{V}_i \oplus \widehat{V}_j = c_i \oplus c_j, U_i = U_j] \\ & \leq \epsilon \,,\end{aligned}$$

where the last inequality follows from the assumption on H.

Finally, when $G_i \neq G_j$ and $G_i, G_j \in \mathcal{G}_{\mathsf{eo}}$, $(Y_i, U_i) = (Y_j, U_j)$ is equivalent to $(C_i, U_i) = (C_j, U_j)$. Therefore, we have

$$\Pr_{\mathrm{id}}[(Y_i, U_i) = (Y_j, U_j)] = \Pr_{\mathrm{id}}[(C_i, U_i) = (C_j, U_j)]$$

$$\begin{aligned}
&\leq \Pr_{\mathrm{id}}[C_i = C_j \mid U_i = U_j] \cdot \sum_{c} \Pr_{\mathrm{id}}[V_i \oplus V_j = c, U_i = U_j] \\
&\leq \frac{1}{2^n} \cdot 2^n \epsilon \\
&= \epsilon \,,
\end{aligned}$$

and therefore, the right hand side of (23) is at most $2\epsilon\binom{q}{2} \leq \epsilon q^2$, which proves Lemma 3. □

Finally, it remains to evaluate the ratio between $\Pr_{\mathrm{re}}[\Theta = \tau]$ and $\Pr_{\mathrm{id}}[\Theta = \tau]$.

Lemma 4. *For any* $\tau \in \mathsf{GoodT}$,

$$\frac{\Pr_{\mathrm{re}}[\Theta = \tau]}{\Pr_{\mathrm{id}}[\Theta = \tau]} \geq 1 \,.$$

Proof. Let $G_1, \ldots, G_q$ denote the q tweaks in the transcript. Let d_{ed} denote the number of distinct tweaks in $\mathcal{G}_{\mathsf{ed}}$ and let $\{G^{\mathsf{ed}}_1, \ldots, G^{\mathsf{ed}}_{d_{\mathsf{ed}}}\}$ be the set of all unique tweaks among $\{G_i : 1 \leq i \leq q, G_i \in \mathcal{G}_{\mathsf{ed}}\}$. Similarly, the number of distinct tweaks in $\mathcal{G}_{\mathsf{eo}}$ is denoted by d_{eo}, and the set of distinct tweaks in $\mathcal{G}_{\mathsf{eo}}$ is denoted by $\{G^{\mathsf{eo}}_1, \ldots, G^{\mathsf{eo}}_{d_{\mathsf{eo}}}\}$. We have $d_{\mathsf{ed}} \leq q_{\mathsf{ed}}$ and $d_{\mathsf{eo}} \leq q_{\mathsf{eo}}$.

For $1 \leq i \leq d_{\mathsf{ed}}$, let $I_{\mathsf{ed}}(i) = \{j : 1 \leq j \leq q, G_j = G^{\mathsf{ed}}_i\}$ and $c_{\mathsf{ed}}(i) = |I_{\mathsf{ed}}(i)|$, i.e., $c_{\mathsf{ed}}(i)$ is the number of queries with $G = G^{\mathsf{ed}}_i$. Similarly we define $I_{\mathsf{eo}}(i)$ and $c_{\mathsf{eo}}(i)$ for $1 \leq i \leq d_{\mathsf{eo}}$.

Here, for $\mathsf{x} \in \{\mathsf{ed}, \mathsf{eo}\}$, $\{I_{\mathsf{x}}(i)\}_{i=1,\ldots,d_{\mathsf{x}}}$ is a partition of $\{i : G_i \in \mathcal{G}_{\mathsf{x}}\}$, that is, $I_{\mathsf{x}}(i) \cap I_{\mathsf{x}}(j) = \emptyset$ for any $i \neq j$, and $\bigcup_{i=1,\ldots,q_{\mathsf{x}}} I_{\mathsf{x}}(i) = \{i : G_i \in \mathcal{G}_{\mathsf{x}}\}$.

Similarly, for $U_1, \ldots, U_q$, let d'_{ed} denote the number of unique U_i's with $G_i \in \mathcal{G}_{\mathsf{ed}}$ (which is uniquely determined as the transcript contains L), and let $\{U^{\mathsf{ed}}_1, \ldots, U^{\mathsf{ed}}_{d'_{\mathsf{ed}}}\}$ be the set of unique U_i's with $G_i \in \mathcal{G}_{\mathsf{ed}}$. Also, for $1 \leq i \leq d'_{\mathsf{ed}}$, let $I'_{\mathsf{ed}}(i) = \{j : 1 \leq j \leq q, U_j = U^{\mathsf{ed}}_i\}$ and $c'_{\mathsf{ed}}(i) = |I'_{\mathsf{ed}}(i)|$. In the same manner, we define d'_{eo}, $\{U^{\mathsf{eo}}_1, \ldots, U^{\mathsf{eo}}_{d'_{\mathsf{eo}}}\}$, and $I'_{\mathsf{eo}}(i)$ and $c'_{\mathsf{eo}}(i)$ for $1 \leq i \leq d'_{\mathsf{eo}}$.

For $i \geq 0$, let $(2^n)_i$ be $(2^n) \cdot (2^n - 1) \cdots (2^n - i + 1)$, with the convention that $(2^n)_0 = 1$. In the ideal world, the probability of any good (in fact, any attainable) τ is completely determined by $\{c_{\mathsf{ed}}(i)\}_{i=1,\ldots,d_{\mathsf{ed}}}$ and $\{c_{\mathsf{eo}}(i)\}_{i=1,\ldots,d_{\mathsf{eo}}}$ from the definition of TURP $\widetilde{\mathsf{E}}$ (which takes the global tweak as its tweak input). For any attainable transcript τ, we have

$$\Pr_{\mathrm{id}}[\Theta = \tau] = \prod_{i=1}^{d_{\mathsf{ed}}} \frac{1}{(2^n)_{c_{\mathsf{ed}}(i)}} \cdot \prod_{j=1}^{d_{\mathsf{eo}}} \frac{1}{(2^n)_{c_{\mathsf{eo}}(j)}} \cdot \frac{1}{|\mathcal{L}|} \,, \tag{24}$$

where the last multiplication by $1/|\mathcal{L}|$ comes from the uniform distribution of L for the keyed hash function.

In the real world, the probability of any good transcript τ is determined by $\{c'_{\mathsf{ed}}(i)\}_{i=1,\ldots,d'_{\mathsf{ed}}}$ and $\{c'_{\mathsf{eo}}(i)\}_{i=1,\ldots,d'_{\mathsf{eo}}}$, since E takes tweak U_i as its input, and if $U_i = U_j$, we must have distinct input blocks and distinct output blocks for E, irrespective of the direction of each query. The probability distribution of a response (which is C_i for an encryption query and M_i for a decryption query) is uniquely determined by the probability distribution of X_i and Y_i as $X_i = M_i \oplus V_i$ and $Y_i = C_i \oplus \widehat{V}_i$, where V_i (and whether $\widehat{V}_i = V_i$ or 0^n) is determined by L and G_i given in the transcript. Therefore, we have

$$\Pr_{\mathrm{re}}[\Theta = \tau] = \prod_{i=1}^{d'_{\mathsf{ed}}} \frac{1}{(2^n)_{c'_{\mathsf{ed}}(i)}} \cdot \prod_{j=1}^{d'_{\mathsf{eo}}} \frac{1}{(2^n)_{c'_{\mathsf{eo}}(j)}} \cdot \frac{1}{|\mathcal{L}|} \,. \tag{25}$$

We observe that $G_i = G_j$ implies $U_i = U_j$, but the other direction is not necessarily true. Thus, for $\mathsf{x} \in \{\mathsf{ed}, \mathsf{eo}\}$ and for all $1 \leq i \leq q_{\mathsf{x}}$, $I_{\mathsf{x}}(i)$ is either equal to $I'_{\mathsf{x}}(j)$ for some j (which occurs iff ${}^{\forall}h \notin I_{\mathsf{x}}(i)$, $U_h \neq U^{\mathsf{x}}_j$), or a union of $\{I'_{\mathsf{x}}(j)\}_{j \in \mathcal{J}}$ for some $\mathcal{J} \subseteq \{1, \ldots, q_{\mathsf{x}}\}$, but it is not possible that, for any $j \neq j'$, $I'_{\mathsf{x}}(i)$ contains both elements of $I_{\mathsf{x}}(j)$ and $I_{\mathsf{x}}(j')$. In other words, $\{I'_{\mathsf{x}}(1), \ldots, I'_{\mathsf{x}}(d'_{\mathsf{x}})\}$ is obtained by applying some join operations to $\{I_{\mathsf{x}}(1), \ldots, I_{\mathsf{x}}(d_{\mathsf{x}})\}$.

This implies that $\prod_{i=1,\ldots,d'_{\mathsf{x}}}(2^n)_{c'_{\mathsf{x}}(i)} \leq \prod_{j=1,\ldots,d_{\mathsf{x}}}(2^n)_{c_{\mathsf{x}}(j)}$ holds for any $\mathsf{x} \in \{\mathsf{ed}, \mathsf{eo}\}$, since $(2^n)_{a+b} \leq (2^n)_a \cdot (2^n)_b$ holds for any non-negative a and b. From this observation, (24), and (25), we obtain Lemma 4. □

Finally, we obtain Lemma 1 from Lemmas 2, 3, and 4.

C Proof of Proposition 5

Let G and G' be two distinct (global) tweaks in $\mathcal{G}$, where $G = (\mathsf{b}, \nu, N, B, i)$ and $G' = (\mathsf{b}', \nu', N', B', i')$. We write $W = \mathsf{mop}_t(2^i\beta) \oplus B$ when $\mathsf{b} = \mathsf{C}$ and $W = \mathsf{mop}_t(2^i\delta) \oplus B$ when $\mathsf{b} = \mathsf{H}$. For G', all the internal variables are written analogously using a prime symbol.

First, we assume $\nu = \nu'$, as otherwise $U \neq U'$ holds with probability one. In what follows, we evaluate $p_{\text{pAXU}} = \max_c \Pr[V \oplus V' = c, W = W']$ and $p_{\text{pU}} = \max_c \Pr[V = c, W = W']$.

We observe that, for the case of ZOCB, V and V' are generated by $\mathsf{E}^{3,[0]_t}$ and $\mathsf{E}^{3,[2]_t}$, and W and W' are generated by $\mathsf{E}^{3,[1]_t}$ and $\mathsf{E}^{3,[3]_t}$. For the case of ZOTR, they are generated by $\mathsf{E}^{6,[0]_t}$, $\mathsf{E}^{6,[2]_t}$, $\mathsf{E}^{6,[1]_t}$, and $\mathsf{E}^{6,[3]_t}$. This implies

$$\begin{cases} \Pr[V \oplus V' = c, W = W'] = \Pr[V \oplus V' = c] \cdot \Pr[W = W']\,, \\ \Pr[V = c, W = W'] = \Pr[V = c] \cdot \Pr[W = W']\,. \end{cases}$$

We also see that V and V' are always uniform (but not necessarily independent), hence $\max_c \Pr[V = c] = \max_c \Pr[V' = c] = 1/2^n$ holds for any case. Therefore, it is sufficient to evaluate $\max_c \Pr[V \oplus V' = c]$ and $\Pr[W = W']$, which is presented in the following seven cases.

Case 1: $\mathsf{b} = \mathsf{C}$ and $\mathsf{b}' = \mathsf{H}$, or $\mathsf{b} = \mathsf{H}$ and $\mathsf{b}' = \mathsf{C}$. We assume $\mathsf{b} = \mathsf{C}$ and $\mathsf{b}' = \mathsf{H}$ as the other case is analyzed similarly. We have $V = 2^i\alpha$, $W = \mathsf{mop}_t(2^i\beta) \oplus B$, $V' = 2^{i'}\gamma$, and $W' = \mathsf{mop}_t(2^{i'}\delta) \oplus B'$. We see that $V \oplus V'$ is uniform, and the first $\min\{n, t\}$ bits of $\mathsf{mop}_t(2^i\beta) \oplus \mathsf{mop}_t(2^{i'}\delta) = \mathsf{mop}_t(2^i\beta \oplus 2^{i'}\delta)$ is uniform. Hence $p_{\text{pAXU}} = p_{\text{pU}} = 1/2^n \cdot 1/2^{\min\{n,t\}}$.

Case 2: $\mathsf{b} = \mathsf{b}' = \mathsf{C}$ and $N \neq N'$. We have $V = 2^i\alpha$, $V' = 2^{i'}\alpha'$, $W = \mathsf{mop}_t(2^i\beta) \oplus B$, and $W' = \mathsf{mop}_t(2^{i'}\beta') \oplus B'$. Because α and α' are the output blocks of a random permutation that takes distinct input blocks (i.e., N and N'), $\Pr[2^i\alpha \oplus 2^{i'}\alpha' = c]$, which is bounded by $\max_x \Pr[2^i\alpha = 2^{i'}x \oplus c \mid \alpha' = x]$, is at most $1/(2^n - 1)$ for any $c \in \{0, 1\}^n$. Similarly, β and β' are the output blocks of a random permutation, and hence $\Pr[\mathsf{mop}_t(2^i\beta) \oplus \mathsf{mop}_t(2^{i'}\beta') = B \oplus B']$ is bounded by $2^{n-t}/(2^n - 1)$ if $t \leq n$ and by $1/(2^n - 1)$ if $t > n$. Therefore, the unified bound is $2/2^{\min\{n,t\}}$. Hence $p_{\text{pAXU}} = 2/2^n \cdot 2/2^{\min\{n,t\}}$ and $p_{\text{pU}} = 1/2^n \cdot 2/2^{\min\{n,t\}}$.

Case 3: $\mathsf{b} = \mathsf{b}' = \mathsf{H}$ and $N \neq N'$. This case is impossible from the definition of $\mathcal{G}$ that has a constraint that $N = 0^n$ must hold when $\mathsf{b} = \mathsf{H}$.

Case 4: $\mathsf{b} = \mathsf{b}' = \mathsf{C}$, $N = N'$, and $i \neq i'$. We have $V = 2^i\alpha$, $V' = 2^{i'}\alpha$, $W = \mathsf{mop}_t(2^i\beta) \oplus B$, and $W' = \mathsf{mop}_t(2^{i'}\beta) \oplus B'$. Then $V \oplus V' = (2^i + 2^{i'}) \cdot \alpha$ is uniform over $\{0, 1\}^n$, and the first $\min\{n, t\}$ bits of $W \oplus W' = \mathsf{mop}_t((2^i + 2^{i'}) \cdot \beta) \oplus B \oplus B'$ is also uniform. Note here that $i, i' \in \mathbb{Z}_\rho = \{0, \ldots, \rho - 1\}$ and $\rho = 2^{(n+\min\{n,t\})/2} - 1$, implying that $2^i + 2^{i'} \neq 0$ for both cases of $\min\{n, t\} = t$ and $\min\{n, t\} = n$.

Therefore, we have $p_{\text{pAXU}} = 1/2^n \cdot 1/2^{\min\{n,t\}}$ and $p_{\text{pU}} = 1/2^n \cdot 1/2^{\min\{n,t\}}$.

Case 5: $\mathsf{b} = \mathsf{b}' = \mathsf{H}$, $N = N'$, and $i \neq i'$. Due to the symmetry to Case 4, we have $p_{\mathrm{pAXU}} = 1/2^n \cdot 1/2^{\min\{n,t\}}$ and $p_{\mathrm{pU}} = 1/2^n \cdot 1/2^{\min\{n,t\}}$.

Case 6: $\mathsf{b} = \mathsf{b}' = \mathsf{C}$, $N = N'$, $i = i'$, and $B \neq B'$. We have $V = 2^i\alpha$, $V' = 2^i\alpha$, $W = \mathsf{mop}_t(2^i\beta) \oplus B$, and $W' = \mathsf{mop}_t(2^i\beta) \oplus B'$. Then $V \oplus V' = 0^n$ and $W \oplus W' = B \oplus B' \neq 0^t$, thus $p_{\mathrm{pAXU}} = p_{\mathrm{pU}} = 0$.

Case 7: $\mathsf{b} = \mathsf{b}' = \mathsf{H}$, $N = N'$, $i = i'$, and $B \neq B'$. The analysis is the same as in Case 6, and we have $p_{\mathrm{pAXU}} = p_{\mathrm{pU}} = 0$.

Finally, we can take p_{pAXU} and p_{pU} as $4/2^{n+\min\{n,t\}}$ to cover all the cases.

Acknowledgments

The authors thank Elmar Tischhauser and Andrey Bogdanov for sharing the code to draw graphs in Figs. 29 and 30. This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Strategic Capability Research Centres Funding Initiative, Nanyang Technological University under research grant M4082123, and Singapore's Ministry of Education under grant M4012049.

References

[ADL17] Tomer Ashur, Orr Dunkelman, and Atul Luykx. Boosting Authenticated Encryption Robustness with Minimal Modifications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 3–33. Springer, 2017.

[AFF$^+$14] Farzaneh Abed, Scott R. Fluhrer, Christian Forler, Eik List, Stefan Lucks, David A. McGrew, and Jakob Wenzel. Pipelineable On-line Encryption. In Carlos Cid and Christian Rechberger, editors, *FSE 2014*, volume 8540 of *LNCS*, pages 205–223. Springer, 2014.

[Ava17] Roberto Avanzi. The QARMA Block Cipher Family. Almost MDS Matrices Over Rings With Zero Divisors, Nearly Symmetric Even-Mansour Constructions With Non-Involutory Central Rounds, and Search Heuristics for Low-Latency S-Boxes. *IACR Trans. Symmetric Cryptol.*, 2017(1):4–44, 2017.

[AY13] Kazumaro Aoki and Kan Yasuda. The Security of the OCB Mode of Operation without the SPRP Assumption. In Willy Susilo and Reza Reyhanitabar, editors, *ProvSec 2013*, volume 8209 of *LNCS*, pages 202–220. Springer, 2013.

[BDPA08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the Indifferentiability of the Sponge Construction. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 181–197. Springer, 2008.

[BDPA11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. In Ali Miri and Serge Vaudenay, editors, *SAC 2011*, volume 7118 of *LNCS*, pages 320–337. Springer, 2011.

[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.

[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 2016, Part II*, volume 9815 of *LNCS*, pages 123–153. Springer, 2016.

[BK09] Alex Biryukov and Dmitry Khovratovich. Related-Key Cryptanalysis of the Full AES-192 and AES-256. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 1–18. Springer, 2009.

[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.

[BLMR19] Christof Beierle, Gregor Leander, Amir Moradi, and Shahram Rasoolzadeh. CRAFT: Lightweight Tweakable Block Cipher with Efficient Protection Against DFA Attacks. *IACR Trans. Symmetric Cryptol.*, 2019(1):5–45, 2019.

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In Tatsuaki Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 531–545. Springer, 2000.

[BN08] Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. *J. Cryptology*, 21(4):469–491, 2008.

[BN17] Ritam Bhaumik and Mridul Nandi. Improved Security for OCB3. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 638–666. Springer, 2017.

[CAE] CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness. https://competitions.cr.yp.to/caesar.html.

[CHP+17] Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. A Security Analysis of Deoxys and its Internal Tweakable Block Ciphers. *IACR Trans. Symmetric Cryptol.*, 2017(3):73–107, 2017.

[CHP+18] Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. Boomerang Connectivity Table: A New Cryptanalysis Tool. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 683–714. Springer, 2018.

[CS14] Shan Chen and John P. Steinberger. Tight Security Bounds for Key-Alternating Ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 327–350. Springer, 2014.

[FLS+10] Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. The Skein Hash Function Family. SHA3 submission to NIST (Round 3), 2010.

[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, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 263–293. Springer, 2016.

[GLNP15] Shay Gueron, Yehuda Lindell, Ariel Nof, and Benny Pinkas. Fast Garbling of Circuits Under Standard Assumptions. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 2015*, pages 567–578. ACM, 2015.

[GLS$^+$14] Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, Kerem Varici, François Durvaux, Lubos Gaspar, and Stéphanie Kerckhof. SCREAM and iSCREAM. CAESAR submission, 2014.

[Hag09] Silvia Hagen. *IPv6 Essentials*. O'Reilly Media, 2009.

[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 2015, Part I*, volume 9056 of *LNCS*, pages 15–44. Springer, 2015.

[HR03] Shai Halevi and Phillip Rogaway. A Tweakable Enciphering Mode. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 482–499. Springer, 2003.

[HR04] Shai Halevi and Phillip Rogaway. A Parallelizable Enciphering Mode. In Tatsuaki Okamoto, editor, *CT-RSA 2004*, volume 2964 of *LNCS*, pages 292–304. Springer, 2004.

[IIMP19] Akiko Inoue, Tetsu Iwata, Kazuhiko Minematsu, and Bertram Poettering. Cryptanalysis of OCB2: Attacks on Authenticity and Confidentiality. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019*, *LNCS*. Springer, 2019. To appear.

[IM18] Akiko Inoue and Kazuhiko Minematsu. Cryptanalysis of OCB2. *IACR Cryptology ePrint Archive*, 2018:1040, 2018.

[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 2017, Part III*, volume 10403 of *LNCS*, pages 34–65. Springer, 2017.

[ISO09] Information technology – Security techniques – Authenticated encryption. ISO/IEC 19772:2009, 2009.

[Iwa18] Tetsu Iwata. Plaintext Recovery Attack of OCB2. *IACR Cryptology ePrint Archive*, 2018:1090, 2018.

[JNP14a] Jérémy Jean, Ivica Nikolić, and Thomas Peyrin. Deoxys v1. CAESAR submission, 2014.

[JNP14b] Jérémy Jean, Ivica Nikolić, and Thomas Peyrin. Joltik v1. CAESAR submission, 2014.

[JNP14c] Jérémy Jean, Ivica Nikolić, and Thomas Peyrin. KIASU v1. CAESAR submission, 2014.

[JNP14d] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 274–288. Springer, 2014.

[Jut01] Charanjit S. Jutla. Encryption Modes with Almost Free Message Integrity. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 529–544. Springer, 2001.

[Jut08] Charanjit S. Jutla. Encryption Modes with Almost Free Message Integrity. *J. Cryptology*, 21(4):547–578, 2008.

[Köl17] Stefan Kölbl. AVX implementation of the Skinny block cipher. https://github.com/kste/skinny_avx, 2017.

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

[KR14] Ted Krovetz and Phillip Rogaway. The OCB Authenticated-Encryption Algorithm. RFC 7253, 2014. https://tools.ietf.org/html/rfc7253.

[KY00] Jonathan Katz and Moti Yung. Unforgeable Encryption and Chosen Ciphertext Secure Modes of Operation. In Bruce Schneier, editor, *FSE 2000*, volume 1978 of *LNCS*, pages 284–299. Springer, 2000.

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

[LRW11] Moses Liskov, Ronald L. Rivest, and David A. Wagner. Tweakable Block Ciphers. *J. Cryptology*, 24(3):588–613, 2011.

[MI15] Kazuhiko Minematsu and Tetsu Iwata. Tweak-Length Extension for Tweakable Blockciphers. In Jens Groth, editor, *IMACC 2015*, volume 9496 of *LNCS*, pages 77–93. Springer, 2015.

[Min14] Kazuhiko Minematsu. Parallelizable Rate-1 Authenticated Encryption from Pseudorandom Functions. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 275–292. Springer, 2014.

[MMS18] Alireza Mehrdad, Farokhlagha Moazami, and Hadi Soleimany. Impossible Differential Cryptanalysis on Deoxys-BC-256. *IACR Cryptology ePrint Archive*, 2018:48, 2018.

[MRV15] Bart Mennink, Reza Reyhanitabar, and Damian Vizár. Security of Full-State Keyed Sponge and Duplex: Applications to Authenticated Encryption. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 465–489. Springer, 2015.

[MV04] David A. McGrew and John Viega. The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In Anne Canteaut and Kapalee Viswanathan, editors, *INDOCRYPT 2004*, volume 3348 of *LNCS*, pages 343–355. Springer, 2004.

[Nai17] Yusuke Naito. Tweakable Blockciphers for Efficient Authenticated Encryptions with Beyond the Birthday-Bound Security. *IACR Trans. Symmetric Cryptol.*, 2017(2):1–26, 2017.

[NIS] NIST Lightweight Cryptography. https://csrc.nist.gov/projects/lightweight-cryptography.

[NIS07] Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D, 2007. National Institute of Standards and Technology.

[NRS14] Chanathip Namprempre, Phillip Rogaway, and Thomas Shrimpton. Reconsidering Generic Composition. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 257–274. Springer, 2014.

[Pat08] Jacques Patarin. The “Coefficients H” Technique. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *SAC 2008*, volume 5381 of *LNCS*, pages 328–345. Springer, 2008.

[Poe18] Bertram Poettering. Breaking the Confidentiality of OCB2. *IACR Cryptology ePrint Archive*, 2018:1087, 2018.

[PS16] Thomas Peyrin and Yannick Seurin. Counter-in-Tweak: Authenticated Encryption Modes for Tweakable Block Ciphers. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 33–63. Springer, 2016.

[RBB03] Phillip Rogaway, Mihir Bellare, and John Black. OCB: A block-cipher mode of operation for efficient authenticated encryption. *ACM Trans. Inf. Syst. Secur.*, 6(3):365–403, 2003.

[RBBK01] Phillip Rogaway, Mihir Bellare, John Black, and Ted Krovetz. OCB: a block-cipher mode of operation for efficient authenticated encryption. In Michael K. Reiter and Pierangela Samarati, editors, *ACM CCS 2001*, pages 196–205. ACM, 2001.

[Rog02] Phillip Rogaway. Authenticated-Encryption with Associated-Data. In Vijayalakshmi Atluri, editor, *ACM CCS 2002*, pages 98–107. ACM, 2002.

[Rog04] Phillip Rogaway. Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 16–31. Springer, 2004.

[RVV15] Reza Reyhanitabar, Serge Vaudenay, and Damian Vizár. Boosting OMD for Almost Free Authentication of Associated Data. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 411–427. Springer, 2015.

[Sas18] Yu Sasaki. Improved Related-Tweakey Boomerang Attacks on Deoxys-BC. In Antoine Joux, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *AFRICACRYPT 2018*, volume 10831 of *LNCS*, pages 87–106. Springer, 2018.

[Sch98] Richard Schroeppel. The Hasty Pudding Cipher. AES submission to NIST, 1998.

[SKI] SKINNY Family of Block Ciphers, Cryptanalysis Competition. `https://sites.google.com/site/skinnycipher/home`.

[SY15] Yu Sasaki and Kan Yasuda. How to Incorporate Associated Data in Sponge-Based Authenticated Encryption. In Kaisa Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 353–370. Springer, 2015.

[ZWH16] Ping Zhang, Peng Wang, and Honggang Hu. The INT-RUP Security of OCB with Intermediate (Parity) Checksum. *IACR Cryptology ePrint Archive*, 2016:1059, 2016.

D Informal Introduction to iZOCB and ZOCB

In this section, we informally introduce iZOCB and ZOCB to see the overview of our schemes, neglecting various corner cases and precise mathematical definitions. To use iZOCB, the idealized version of ZOCB, we assume that there is a TBC $\widetilde{\mathsf{E}}$ that takes an n-bit block X as input, and has the tweak space $\{\mathsf{C},\mathsf{H}\} \times \{0,1,2\} \times \{0,1\}^n \times \{0,1\}^t \times \{0,1,\dots,2^{(n+\min\{n,t\})/2}-2\}$. For simplicity, we assume that $n=t$ and hence the tweak space is $\{\mathsf{C},\mathsf{H}\} \times \{0,1,2\} \times \{0,1\}^n \times \{0,1\}^n \times \{0,1,\dots,2^n-2\}$. The first two arguments of the tweak space are for domain separation, the third argument $\{0,1\}^n$ is for a nonce, the fourth argument $\{0,1\}^n$ is for an AD block, and the last argument $\{0,1,\dots,2^n-2\}$ is for a block counter.

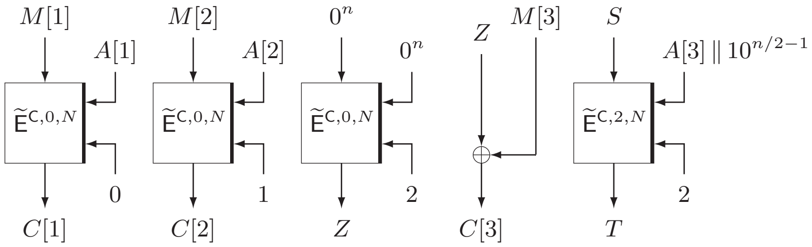
Suppose that we have a plaintext $M=(M[1],M[2],M[3]) \in \{0,1\}^{3n}$ and AD $A=(A[1],A[2],A[3]) \in \{0,1\}^{2.5n}$, where $|A[1]|=|A[2]|=n$ and $|A[3]|=0.5n$. Given (N,A,M), the encryption process to compute the ciphertext $C=(C[1],C[2],C[3]) \in \{0,1\}^{3n}$ and the tag $T \in \{0,1\}^n$ is as in Fig. 31. The decryption, given (N,A,C,T) with $C=(C[1],C[2],C[3]) \in \{0,1\}^{3n}$ and $A=(A[1],A[2],A[3]) \in \{0,1\}^{2.5n}$, is to first obtain $M[1]$ from $(A[1],C[1])$, then $M[2]$ from $(A[2],C[2])$, $M[3]$ from Z and $C[3]$, and we check the validity of T by comparing it with T^*, which is the encryption of the checksum $S=M[1]\oplus M[2]\oplus M[3]$ and $A[3]$.

With respect to the security, the privacy is proved from the fact that all the TBC calls use distinct tweak values, and hence the adversary that has access to the encryption oracle obtains the output blocks of the TBC with distinct tweak values. The authenticity is non-obvious. We see that the tag T is independent of $A[1]$ and $A[2]$ in encryption, and iZOCB nevertheless achieves provable authenticity. The intuition is that, given (N,A,M,C,T) obtained from the encryption oracle, if the adversary manipulates $A[1]$ or $A[2]$ to forge in decryption, this will result in a random checksum S as this will randomize $M[1]$ or $M[2]$. More precisely, let us assume that the adversary makes one encryption query and has (N,A,M,C,T), and now makes a single forgery attempt (N',A',C',T'), where $|A|=|A'|=2.5n$ and $|C|=|C'|=3n$.

- If $N' \neq N$, then (N,A,M,C,T) obtained from the encryption oracle is of no use since the tag T^* for (N',A',C') is computed by using a tweak $(\mathsf{C},2,N',A'[3]\,\|\,10^{n/2-1},2)$, which is unique due to the domain separation and N'.
- If $N'=N$ and $C' \neq C$, then we have $C'[1] \neq C[1]$, $C'[2] \neq C[2]$, or $C'[3] \neq C[3]$. If $C'[1] \neq C[1]$, then $M'[1]$ is uniformly random over $\{0,1\}^n$ (in case $A'[1] \neq A[1]$) or $\{0,1\}^n \setminus \{M[1]\}$ (in case $A'[1]=A[1]$). In either case, we see that the checksum S' for (N',A',C') is sufficiently randomized, and the forgery cannot succeed. The same analysis applies for the case $C'[2] \neq C[2]$. If $C'[1]=C[1]$, $C'[2]=C[2]$, and $C'[3] \neq C[3]$, then we must have $S' \neq S$ (in case $A'[1]=A[1]$ and $A'[2]=A[2]$), or S' is uniformly random over $\{0,1\}^n$ (in case $A'[1] \neq A[1]$ or $A'[2] \neq A[2]$). Therefore, the success probability of the forgery is low for all the cases.
- If $N'=N$, $C'=C$, and $A' \neq A$, then we have $A'[1] \neq A[1]$, $A'[2] \neq A[2]$, or $A'[3] \neq A[3]$. In the former two cases, S' is uniformly random over $\{0,1\}^n$, and in the last case, T^* is uniformly random over $\{0,1\}^n$, and hence the forgery will be unsuccessful with a high probability.

Next, we consider ZOCB. Let E_K be a TBC as a primitive. This could be SKINNY or Deoxys-BC. It takes an n-bit block X as input, and has the tweak space $\{0,1,2,3\} \times \{0,1\}^t$. We assume that $n=t$, and we consider the following instantiation of $\widetilde{\mathsf{E}}$.

$$\begin{cases} \widetilde{\mathsf{E}}^{\mathsf{C},0,N,A,i}(X) = E_K^{0,2^i\beta\oplus A}(X \oplus 2^i\alpha) \oplus 2^i\alpha & (26) \\ \widetilde{\mathsf{E}}^{\mathsf{C},2,N,A,i}(X) = E_K^{2,2^i\beta\oplus A}(X \oplus 2^i\alpha) & (27) \end{cases}$$



$M[1]$
$A[1]$
$\widetilde{\mathsf{E}}^{\mathsf{C},0,N}$
0
$C[1]$
$M[2]$
$A[2]$
$\widetilde{\mathsf{E}}^{\mathsf{C},0,N}$
1
$C[2]$
0^n
0^n
$\widetilde{\mathsf{E}}^{\mathsf{C},0,N}$
2
Z
Z
$M[3]$
$C[3]$
S
$A[3] \,\|\, 10^{n/2-1}$
$\widetilde{\mathsf{E}}^{\mathsf{C},2,N}$
2
T

Figure 31: The encryption process of a plaintext $M = (M[1], M[2], M[3]) \in \{0,1\}^{3n}$ and AD $A = (A[1], A[2], A[3]) \in \{0,1\}^{2.5n}$ with iZOCB to produce the ciphertext $C = (C[1], C[2], C[3])$ and the tag T. S is $M[1] \oplus M[2] \oplus M[3]$.

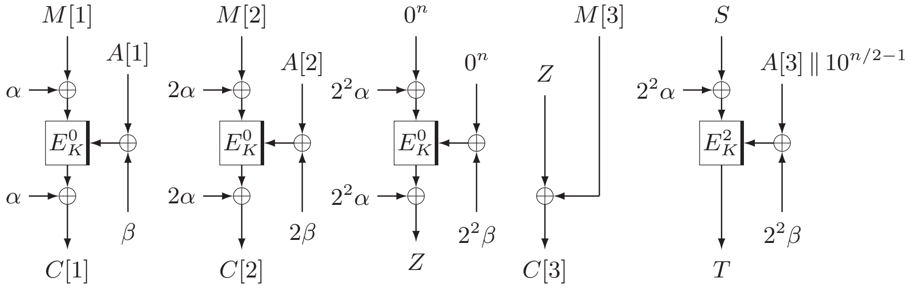


Figure 32: The encryption process of $M = (M[1], M[2], M[3]) \in \{0,1\}^{3n}$ and $A = (A[1], A[2], A[3]) \in \{0,1\}^{2.5n}$ with ZOCB to produce $C = (C[1], C[2], C[3])$ and T. $\alpha = E_K^{3,0^n}(N), \beta = E_K^{3,0^{n-1}1}(N)$, and $S = M[1] \oplus M[2] \oplus M[3]$.

Here, $\alpha = E_K^{3,0^n}(N)$ and $\beta = E_K^{3,0^{n-1}1}(N)$. With this instantiation, we obtain Fig. 32, which shows the encryption process of ZOCB.

It can be shown that, under certain restrictions where the adversary has the encryption and decryption oracles of (26) but only the encryption oracle of (27), the adversary cannot distinguish the ideal $\widetilde{\mathsf{E}}$ from $\widetilde{\mathsf{E}}$ that is instantiated as (26) and (27). The adversary with the restrictions can still simulate iZOCB (in the case of ideal $\widetilde{\mathsf{E}}$) or ZOCB (in the case of $\widetilde{\mathsf{E}}$ that is instantiated as (26) and (27)), and hence the provable security of ZOCB follows from the security of iZOCB.

E Informal Introduction to iZOTR and ZOTR

As the introduction of iZOCB and ZOCB in Appendix D, we informally introduce iZOTR and ZOTR. To use iZOTR, we fix a TBC $\widetilde{\mathsf{E}}$ that takes an n-bit block X as input, and has the tweak space $\{\mathsf{C}, \mathsf{H}\} \times \{0, 1, \ldots, 5\} \times \{0,1\}^n \times \{0,1\}^t \times \{0, 1, \ldots, 2^{(n+\min\{n,t\})/2} - 2\}$. We assume $n = t$, in which case the tweak space is $\{\mathsf{C}, \mathsf{H}\} \times \{0, 1, \ldots, 5\} \times \{0,1\}^n \times \{0,1\}^n \times \{0, 1, \ldots, 2^n - 2\}$. The tweak space has the same semantics as iZOCB, i.e., the first two arguments are for domain separation, the third argument $\{0,1\}^n$ is for a nonce, the fourth argument $\{0,1\}^n$ is for an AD block, and the last argument $\{0, 1, \ldots, 2^n - 2\}$ is for a block counter.

To encrypt (N, A, M), where $M = (M[1], M[2], M[3]) \in \{0,1\}^{3n}$ is a plaintext and $A = (A[1], A[2], A[3]) \in \{0,1\}^{2.5n}$ is AD, the ciphertext $C = (C[1], C[2], C[3]) \in \{0,1\}^{3n}$ and the tag $T \in \{0,1\}^n$ are computed as in Fig. 33. To decrypt (N, A, C, T) with $C = (C[1], C[2], C[3]) \in \{0,1\}^{3n}$ and $A = (A[1], A[2], A[3]) \in \{0,1\}^{2.5n}$, we first compute $M[1]$ from $(A[2], C[1], C[2])$, then $M[2]$ from $(A[1], M[1], C[1])$, and $M[3]$ from $C[3]$. We then check if $T = T^*$ holds, where T^* is obtained from $S = M[2] \oplus M[3]$ and $A[3]$.

The privacy is perfect from the uniqueness of the tweak in all the TBC calls. The authenticity needs a case analysis, and as in Appendix D, we assume that the adversary makes one encryption query to obtain (N, A, M, C, T), and now makes a single forgery attempt (N', A', C', T'), where $|A| = |A'| = 2.5n$ and $|C| = |C'| = 3n$.

- If $N' \neq N$, then the tag T^* for (N', A', C') is computed by using a unique tweak value $(\mathsf{C}, 5, N', A'[3] \,\|\, 10^{n/2-1}, 1)$, and hence the success probability is $1/2^n$.
- If $N' = N$ and $C' \neq C$, then we have $C'[1] \neq C[1]$, $C'[2] \neq C[2]$, or $C'[3] \neq C[3]$. If $C'[1] \neq C[1]$, then $M'[1]$ is randomized from the randomness of $\widetilde{\mathsf{E}}^{\mathsf{C},1,N',A[2],0}(C'[1])$, and the probability of $M[1] = M'[1]$ is low. Under the condition that $M[1] \neq M'[1]$, $M'[2]$ is sufficiently randomized from the randomness of $\widetilde{\mathsf{E}}^{\mathsf{C},0,N',A[1],0}(M'[1])$. It follows that S' is sufficiently randomized from $M'[2]$, and hence the forgery will not succeed.

  If $C'[1] = C[1]$ and $C'[2] \neq C[2]$, then we must have $M[1] \neq M'[1]$ (in case $A'[2] = A[2]$), or $M[1] \neq M'[1]$ holds with an overwhelming probability (in case $A'[2] \neq A[2]$), and hence the same reasoning as above works.

  If $C'[1] = C[1], C'[2] = C[2]$, and $C'[3] \neq C[3]$, then $S' \neq S$ holds from $M'[3] \neq M[3]$ (in case $A'[1] = A[1]$ and $A'[2] = A[2]$), or S' is sufficiently randomized (in case $A'[1] \neq A[1]$ or $A'[2] \neq A[2]$). Therefore, the success probability of the forgery is low.
- If $N' = N$, $C' = C$, and $A' \neq A$, then we have $A'[1] \neq A[1]$, $A'[2] \neq A[2]$, or $A'[3] \neq A[3]$. We see that $M'[2]$ is sufficiently random over $\{0,1\}^n$ in the first two cases, implying the randomness of S'. The last case directly implies that T^* is uniformly random, and the forgery will not succeed.

Next, we present ZOTR. Let E_K be a TBC as a primitive. It takes an n-bit block X as input, and has the tweak space $\{0, 1, \ldots, 6\} \times \{0,1\}^t$. We assume that $n = t$, and we instantiate $\widetilde{\mathsf{E}}$ as follows.

$$\begin{cases} \widetilde{\mathsf{E}}^{\mathsf{C},0,N,A,i}(X) = E_K^{0,2^i\beta \oplus A}(X \oplus 2^i\alpha) & (28) \\ \widetilde{\mathsf{E}}^{\mathsf{C},1,N,A,i}(X) = E_K^{1,2^i\beta \oplus A}(X \oplus 2^i\alpha) & (29) \\ \widetilde{\mathsf{E}}^{\mathsf{C},5,N,A,i}(X) = E_K^{5,2^i\beta \oplus A}(X \oplus 2^i\alpha) & (30) \end{cases}$$

Here, $\alpha = E_K^{6,0^n}(N)$ and $\beta = E_K^{6,0^{n-1}1}(N)$. With this instantiation, we obtain Fig. 32, which shows the encryption process of ZOTR.

We can show that the adversary that has the encryption oracles of (28)–(30) cannot distinguish the ideal $\widetilde{\mathsf{E}}$ from $\widetilde{\mathsf{E}}$ that is instantiated as (28)–(30). In case of ideal $\widetilde{\mathsf{E}}$, we obtain iZOTR, and in the case of $\widetilde{\mathsf{E}}$ that is instantiated as (28)–(30), we obtain ZOTR. Therefore, the provable security of ZOTR follows from the security of iZOTR.

F Raw Timing Data

Here, we report raw timing data of TAES-ZOCB/TAES-ZOTR/TAES-ΘCB3 and SKINNY-ZOCB/SKINNY-ZOTR/SKINNY-ΘCB3.

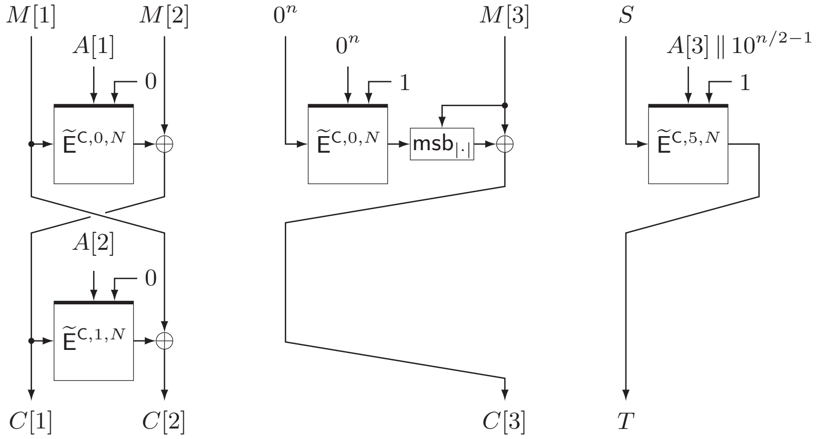

$M[1]$
$M[2]$
$A[1]$
0
$\widetilde{\mathsf{E}}^{\mathsf{C},0,N}$
$A[2]$
0
$\widetilde{\mathsf{E}}^{\mathsf{C},1,N}$
$C[1]$
$C[2]$
0^n
0^n
1
$\widetilde{\mathsf{E}}^{\mathsf{C},0,N}$
$\mathsf{msb}_{|\cdot|}$
$M[3]$
$C[3]$
S
$A[3] \,\|\, 10^{n/2-1}$
1
$\widetilde{\mathsf{E}}^{\mathsf{C},5,N}$
T

Figure 33: The encryption process of a plaintext $M = (M[1], M[2], M[3]) \in \{0,1\}^{3n}$ and AD $A = (A[1], A[2], A[3]) \in \{0,1\}^{2.5n}$ with iZOTR. The output is a ciphertext $C = (C[1], C[2], C[3])$ and a tag T. S is $M[2] \oplus M[3]$.

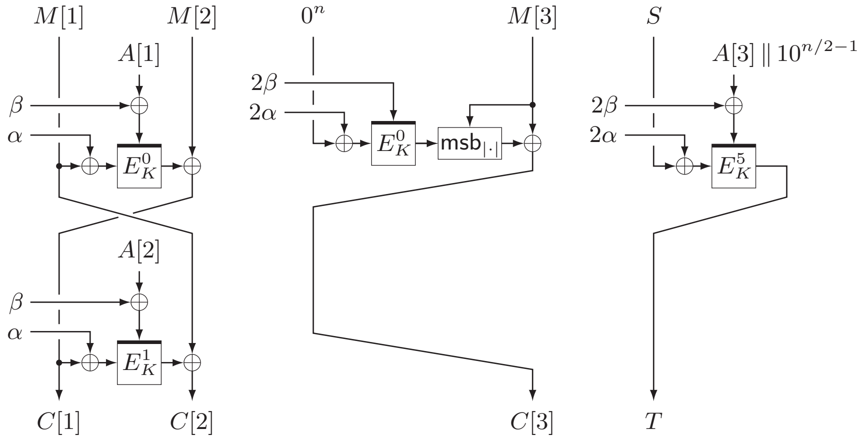


Figure 34: The encryption process of $M = (M[1], M[2], M[3]) \in \{0,1\}^{3n}$ and $A = (A[1], A[2], A[3]) \in \{0,1\}^{2.5n}$ with ZOTR to produce $C = (C[1], C[2], C[3])$ and T. $\alpha = E_K^{6,0^n}(N), \beta = E_K^{6,0^{n-1}1}(N)$, and $S = M[2] \oplus M[3]$.

Table 2: Software performance in c/B of TAES-ZOCB/ZOTR/ΘCB3 on Skylake compiled using icc, with various input length (in bytes)

TAES-ZOCB																		
M\AD	0	15	30	60	120	240	480	960	1920	3840	7680	15360	30720	61440	122880	245760	491520	983040
16	25.5	23.24	16.51	11.65	7.66	5.73	3.49	2.53	1.99	1.77	1.63	1.56	1.56	1.54	1.54	1.55	1.56	1.56
32	15.2	10.24	12.86	9.08	6.85	5.35	3.54	2.46	1.95	1.76	1.63	1.57	1.56	1.54	1.54	1.55	1.56	1.56
64	10.0	8.15	6.85	7.71	6.07	5.09	3.52	2.57	2.03	1.76	1.62	1.55	1.55	1.53	1.53	1.55	1.56	1.56
128	6.52	5.81	5.27	4.43	4.65	4.02	3.46	2.60	2.05	1.77	1.63	1.56	1.55	1.53	1.53	1.55	1.56	1.56
256	4.85	4.56	4.32	3.94	3.30	3.74	2.88	2.34	1.94	1.71	1.61	1.57	1.56	1.54	1.54	1.55	1.56	1.56
512	3.92	3.76	3.69	3.51	3.16	2.66	2.64	2.21	1.91	1.70	1.59	1.55	1.54	1.53	1.53	1.55	1.55	1.55
1024	3.43	3.36	3.32	3.22	3.07	2.79	2.32	2.09	1.85	1.69	1.59	1.55	1.54	1.53	1.53	1.55	1.55	1.55
2048	3.17	3.15	3.13	3.09	3.00	2.86	2.56	2.16	1.81	1.67	1.60	1.55	1.54	1.53	1.53	1.55	1.55	1.55
4096	3.05	3.03	3.01	3.00	2.97	2.88	2.73	2.46	2.07	1.68	1.61	1.57	1.55	1.53	1.53	1.55	1.55	1.55
8192	2.99	2.99	2.99	2.98	2.96	2.92	2.82	2.68	2.42	2.04	1.61	1.58	1.56	1.54	1.54	1.55	1.55	1.55
16384	2.96	2.96	2.96	2.95	2.94	2.94	2.87	2.80	2.65	2.41	2.02	1.59	1.56	1.54	1.54	1.55	1.56	1.55
32768	2.96	2.98	2.97	2.95	2.96	2.95	2.93	2.88	2.78	2.66	2.44	2.06	1.58	1.55	1.55	1.56	1.56	1.56
65536	2.99	2.97	2.98	2.98	2.99	2.98	2.95	2.94	2.90	2.82	2.68	2.40	2.04	1.57	1.57	1.56	1.56	1.56
131072	2.96	2.97	2.97	2.99	2.97	2.98	2.97	2.97	2.93	2.89	2.81	2.68	2.42	2.03	1.57	1.57	1.57	1.56
262144	2.98	2.95	2.96	2.96	2.96	2.95	2.95	2.96	2.93	2.91	2.87	2.79	2.65	2.41	2.02	1.58	1.57	1.56
524288	2.97	2.96	2.96	2.97	2.97	2.97	2.97	2.96	2.97	2.95	2.93	2.88	2.82	2.67	2.42	2.04	1.57	1.57
1048576	2.98	2.99	2.99	2.98	2.98	2.98	2.97	2.98	2.98	2.96	2.95	2.94	2.90	2.82	2.68	2.43	2.06	1.56

TAES-ZOTR																		
M\AD	0	15	30	60	120	240	480	960	1920	3840	7680	15360	30720	61440	122880	245760	491520	983040
16	25.44	23.06	16.23	11.57	7.59	5.69	3.46	2.53	2.00	1.77	1.63	1.56	1.56	1.54	1.54	1.55	1.56	1.56
32	15.02	10.19	12.63	8.98	6.78	5.26	3.51	2.46	1.96	1.76	1.63	1.57	1.56	1.54	1.54	1.55	1.56	1.56
64	9.95	8.01	6.73	7.58	5.97	5.03	3.46	2.56	2.04	1.76	1.62	1.55	1.55	1.53	1.53	1.55	1.56	1.56
128	7.36	6.56	5.93	5.00	5.03	4.27	3.56	2.67	2.09	1.80	1.64	1.56	1.55	1.53	1.53	1.55	1.56	1.56
256	5.24	4.92	4.68	4.22	3.54	3.94	3.00	2.42	2.00	1.74	1.63	1.57	1.56	1.54	1.54	1.55	1.56	1.56
512	3.98	3.87	3.77	3.57	3.24	2.73	2.71	2.26	1.95	1.72	1.60	1.55	1.55	1.53	1.53	1.55	1.55	1.55
1024	3.38	3.34	3.29	3.20	3.03	2.74	2.30	2.07	1.85	1.69	1.59	1.54	1.54	1.53	1.53	1.55	1.55	1.55
2048	3.09	3.06	3.05	3.00	2.92	2.77	2.51	2.11	1.75	1.64	1.58	1.54	1.54	1.53	1.53	1.55	1.55	1.55
4096	2.92	2.91	2.90	2.88	2.84	2.77	2.62	2.37	1.99	1.60	1.56	1.54	1.54	1.53	1.53	1.55	1.55	1.55
8192	2.86	2.85	2.85	2.83	2.81	2.77	2.70	2.56	2.31	1.95	1.52	1.52	1.52	1.52	1.52	1.54	1.55	1.55
16384	2.81	2.81	2.81	2.80	2.79	2.77	2.74	2.66	2.52	2.29	1.93	1.49	1.50	1.51	1.52	1.54	1.55	1.55
32768	2.79	2.80	2.81	2.79	2.79	2.79	2.76	2.72	2.65	2.51	2.28	1.92	1.48	1.49	1.51	1.53	1.55	1.55
65536	2.79	2.79	2.79	2.79	2.79	2.79	2.77	2.75	2.71	2.64	2.51	2.27	1.91	1.47	1.49	1.52	1.54	1.55
131072	2.78	2.78	2.79	2.79	2.78	2.78	2.77	2.77	2.75	2.71	2.63	2.50	2.26	1.91	1.47	1.50	1.52	1.54
262144	2.79	2.79	2.79	2.78	2.79	2.79	2.78	2.78	2.77	2.75	2.71	2.64	2.50	2.27	1.92	1.47	1.50	1.52
524288	2.79	2.79	2.79	2.79	2.79	2.79	2.79	2.78	2.78	2.77	2.75	2.71	2.64	2.51	2.27	1.92	1.47	1.50
1048576	2.79	2.79	2.79	2.79	2.80	2.79	2.79	2.79	2.78	2.78	2.77	2.75	2.72	2.64	2.50	2.28	1.92	1.46

TAES-ΘCB3																		
M\AD	0	15	30	60	120	240	480	960	1920	3840	7680	15360	30720	61440	122880	245760	491520	983040
16	11.59	8.63	7.62	6.41	4.93	3.53	2.97	2.66	2.58	2.54	2.52	2.51	2.50	2.50	2.50	2.49	2.50	2.50
32	7.92	7.18	6.77	6.02	4.87	3.58	3.00	2.69	2.60	2.55	2.53	2.51	2.50	2.50	2.50	2.49	2.50	2.50
64	4.55	4.95	5.05	4.90	4.30	3.33	2.92	2.63	2.57	2.54	2.51	2.50	2.50	2.50	2.50	2.50	2.50	2.50
128	3.55	3.88	4.03	4.10	3.87	3.21	2.89	2.63	2.56	2.53	2.52	2.50	2.50	2.50	2.50	2.50	2.50	2.50
256	3.05	3.25	3.36	3.46	3.40	3.04	2.83	2.63	2.57	2.54	2.52	2.51	2.50	2.50	2.50	2.50	2.50	2.50
512	2.82	2.94	3.01	3.07	3.09	2.88	2.77	2.62	2.57	2.54	2.52	2.51	2.51	2.50	2.50	2.50	2.50	2.50
1024	2.69	2.75	2.79	2.83	2.85	2.76	2.70	2.60	2.57	2.54	2.52	2.51	2.51	2.50	2.50	2.50	2.50	2.50
2048	2.63	2.66	2.68	2.70	2.71	2.67	2.64	2.59	2.57	2.55	2.53	2.51	2.51	2.50	2.50	2.50	2.50	2.50
4096	2.59	2.61	2.62	2.64	2.64	2.62	2.61	2.58	2.57	2.55	2.53	2.52	2.51	2.51	2.50	2.50	2.50	2.50
8192	2.58	2.59	2.60	2.60	2.61	2.60	2.59	2.57	2.57	2.56	2.54	2.53	2.52	2.51	2.50	2.50	2.50	2.50
16384	2.57	2.58	2.58	2.58	2.59	2.58	2.58	2.57	2.57	2.56	2.55	2.54	2.53	2.51	2.51	2.50	2.50	2.50
32768	2.57	2.57	2.57	2.58	2.58	2.58	2.57	2.57	2.57	2.57	2.56	2.55	2.54	2.52	2.51	2.51	2.50	2.50
65536	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.56	2.56	2.56	2.55	2.54	2.52	2.51	2.51	2.50
131072	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.56	2.56	2.56	2.56	2.55	2.53	2.52	2.51	2.51
262144	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.56	2.56	2.56	2.56	2.55	2.54	2.53	2.52	2.51
524288	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.56	2.56	2.56	2.56	2.55	2.54	2.53	2.52
1048576	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.57	2.56	2.56	2.55	2.54	2.53

Table 3: Software performance in c/B of TAES-ZOCB/ZOTR/ΘCB3 on Haswell compiled using icc, with various input length (in bytes)

TAES-ZOCB																	
M\AD	0	15	30	60	120	240	480	960	1920	3840	7680	15360	30720	61440	122880	245760	491520	983040
16	30.16	26.92	19.20	13.48	9.25	6.79	4.42	3.45	2.90	2.63	2.49	2.41	2.44	2.42	2.42	2.46	2.47	2.48
32	17.81	12.17	14.85	10.42	7.87	6.42	4.46	3.37	2.85	2.62	2.48	2.42	2.44	2.42	2.42	2.46	2.47	2.47
64	12.24	9.91	8.36	9.16	7.12	6.18	4.45	3.50	2.93	2.63	2.49	2.43	2.44	2.42	2.42	2.46	2.47	2.47
128	8.33	7.46	6.76	5.67	5.70	4.89	4.29	3.47	2.93	2.63	2.49	2.42	2.45	2.42	2.42	2.46	2.47	2.48
256	6.57	6.23	5.89	5.33	4.48	4.73	3.72	3.21	2.80	2.56	2.45	2.40	2.42	2.40	2.41	2.45	2.46	2.47
512	5.55	5.39	5.25	4.96	4.48	3.77	3.52	3.04	2.74	2.54	2.44	2.40	2.42	2.40	2.41	2.45	2.46	2.47
1024	5.02	4.95	4.88	4.74	4.51	4.07	3.42	2.92	2.65	2.51	2.43	2.40	2.42	2.39	2.40	2.45	2.46	2.47
2048	4.76	4.72	4.69	4.62	4.50	4.26	3.86	3.24	2.62	2.48	2.42	2.41	2.41	2.39	2.40	2.45	2.46	2.47
4096	4.64	4.62	4.60	4.57	4.50	4.39	4.15	3.76	3.16	2.48	2.42	2.42	2.42	2.39	2.40	2.45	2.46	2.47
8192	4.57	4.56	4.55	4.54	4.51	4.44	4.32	4.09	3.71	3.12	2.40	2.40	2.41	2.39	2.41	2.45	2.46	2.47
16384	4.55	4.55	4.54	4.53	4.52	4.48	4.42	4.30	4.08	3.70	3.12	2.41	2.40	2.40	2.42	2.45	2.46	2.47
32768	4.54	4.54	4.54	4.53	4.53	4.51	4.47	4.41	4.29	4.07	3.69	3.11	2.38	2.38	2.41	2.45	2.46	2.47
65536	4.53	4.53	4.53	4.53	4.53	4.52	4.50	4.47	4.41	4.29	4.06	3.68	3.10	2.37	2.40	2.44	2.45	2.46
131072	4.54	4.54	4.54	4.54	4.53	4.53	4.52	4.50	4.47	4.41	4.29	4.07	3.69	3.12	2.39	2.42	2.44	2.46
262144	4.56	4.56	4.56	4.56	4.56	4.56	4.55	4.55	4.53	4.50	4.44	4.32	4.10	3.72	3.14	2.40	2.43	2.45
524288	4.57	4.57	4.57	4.57	4.57	4.57	4.56	4.56	4.55	4.54	4.50	4.44	4.32	4.10	3.72	3.14	2.40	2.43
1048576	4.57	4.57	4.57	4.57	4.57	4.57	4.57	4.57	4.56	4.56	4.54	4.51	4.44	4.32	4.10	3.72	3.14	2.40

TAES-ZOTR																	
M\AD	0	15	30	60	120	240	480	960	1920	3840	7680	15360	30720	61440	122880	245760	491520	983040
16	29.88	26.89	19.12	13.43	9.24	6.80	4.39	3.44	2.87	2.63	2.48	2.42	2.42	2.40	2.41	2.44	2.46	2.47
32	17.61	12.12	14.96	10.37	7.89	6.40	4.47	3.35	2.82	2.61	2.47	2.42	2.41	2.40	2.41	2.44	2.46	2.47
64	11.91	9.68	8.06	9.05	7.01	6.09	4.40	3.46	2.89	2.61	2.48	2.42	2.42	2.40	2.41	2.44	2.46	2.47
128	9.06	8.13	7.35	6.15	6.14	5.16	4.44	3.54	2.96	2.65	2.49	2.43	2.42	2.40	2.41	2.44	2.46	2.47
256	7.19	6.79	6.43	5.82	4.90	5.03	3.93	3.33	2.87	2.59	2.47	2.42	2.41	2.39	2.41	2.44	2.46	2.47
512	5.62	5.44	5.30	5.01	4.54	3.82	3.54	3.06	2.75	2.54	2.45	2.41	2.41	2.39	2.41	2.44	2.46	2.47
1024	5.01	4.94	4.87	4.74	4.51	4.07	3.41	2.92	2.65	2.50	2.43	2.40	2.40	2.39	2.40	2.44	2.46	2.47
2048	4.65	4.62	4.59	4.52	4.40	4.16	3.77	3.16	2.56	2.44	2.40	2.39	2.40	2.38	2.40	2.44	2.46	2.47
4096	4.49	4.48	4.46	4.43	4.37	4.25	4.02	3.65	3.06	2.40	2.36	2.37	2.39	2.38	2.40	2.44	2.46	2.47
8192	4.41	4.40	4.39	4.38	4.35	4.29	4.17	3.95	3.57	3.00	2.32	2.33	2.37	2.37	2.39	2.44	2.46	2.46
16384	4.38	4.37	4.37	4.36	4.35	4.32	4.26	4.14	3.92	3.56	3.00	2.31	2.33	2.35	2.38	2.44	2.45	2.46
32768	4.37	4.37	4.37	4.36	4.36	4.34	4.31	4.25	4.13	3.92	3.55	2.99	2.29	2.32	2.36	2.42	2.44	2.46
65536	4.37	4.37	4.36	4.36	4.36	4.35	4.33	4.30	4.24	4.13	3.91	3.55	2.99	2.28	2.33	2.40	2.43	2.45
131072	4.37	4.37	4.37	4.37	4.36	4.36	4.35	4.34	4.31	4.25	4.13	3.92	3.55	3.00	2.30	2.36	2.40	2.43
262144	4.39	4.39	4.39	4.39	4.39	4.39	4.38	4.38	4.36	4.33	4.27	4.16	3.94	3.58	3.02	2.31	2.37	2.41
524288	4.40	4.40	4.40	4.40	4.40	4.40	4.39	4.39	4.38	4.37	4.34	4.28	4.16	3.95	3.58	3.02	2.31	2.36
1048576	4.40	4.40	4.40	4.40	4.40	4.40	4.40	4.40	4.39	4.39	4.37	4.34	4.28	4.16	3.95	3.58	3.02	2.31

TAES-ΘCB3																	
M\AD	0	15	30	60	120	240	480	960	1920	3840	7680	15360	30720	61440	122880	245760	491520	983040
16	14.03	10.47	9.30	7.89	6.38	5.04	4.58	4.31	4.24	4.19	4.17	4.16	4.16	4.16	4.16	4.16	4.16	4.16
32	9.86	8.83	8.33	7.48	6.30	5.10	4.61	4.33	4.25	4.20	4.18	4.17	4.16	4.16	4.16	4.16	4.16	4.16
64	6.06	6.38	6.48	6.28	5.71	4.81	4.49	4.27	4.21	4.19	4.17	4.16	4.16	4.16	4.16	4.16	4.16	4.16
128	5.10	5.34	5.52	5.57	5.30	4.69	4.44	4.27	4.21	4.19	4.17	4.16	4.16	4.16	4.16	4.16	4.16	4.16
256	4.62	4.77	4.90	5.00	4.91	4.55	4.39	4.25	4.22	4.19	4.17	4.16	4.16	4.16	4.16	4.16	4.16	4.16
512	4.38	4.47	4.56	4.62	4.59	4.41	4.32	4.23	4.20	4.18	4.17	4.16	4.16	4.16	4.16	4.16	4.16	4.16
1024	4.26	4.31	4.35	4.39	4.38	4.30	4.27	4.21	4.19	4.18	4.17	4.16	4.16	4.16	4.16	4.16	4.16	4.16
2048	4.21	4.23	4.25	4.28	4.28	4.24	4.22	4.19	4.19	4.17	4.17	4.16	4.16	4.16	4.16	4.16	4.16	4.16
4096	4.18	4.19	4.20	4.21	4.22	4.20	4.19	4.17	4.18	4.17	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16
8192	4.16	4.17	4.17	4.18	4.18	4.17	4.17	4.17	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16
16384	4.16	4.16	4.17	4.17	4.17	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16
32768	4.15	4.16	4.16	4.16	4.16	4.16	4.16	4.15	4.15	4.15	4.15	4.16	4.15	4.15	4.15	4.16	4.16	4.16
65536	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.16	4.16	4.16	4.16
131072	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.16	4.16	4.16	4.16	4.16
262144	4.15	4.15	4.16	4.16	4.16	4.15	4.15	4.16	4.16	4.15	4.16	4.16	4.15	4.16	4.16	4.16	4.16	4.16
524288	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16
1048576	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16	4.16

Table 4: Software performance in c/B of SKINNY-ZOCB/ZOTR/ΘCB3 on Skylake compiled using gcc, with various input length (in bytes)

SKINNY-ZOCB																		
M\AD	0	15	30	60	120	240	480	960	1920	3840	7680	15360	30720	61440	122880	245760	491520	983040
16	816.38	738.94	498.88	343.99	239.44	178.17	137.04	122.22	112.91	58.34	29.26	14.42	7.20	6.76	4.93	4.02	3.56	3.33
32	511.38	348.24	422.51	285.24	214.65	167.98	139.26	120.33	112.05	58.13	29.21	14.62	7.20	6.77	4.93	4.02	3.57	3.33
64	358.85	290.84	244.33	264.80	196.11	161.65	137.48	123.07	111.96	58.50	29.52	14.80	7.40	6.80	4.95	4.03	3.57	3.33
128	282.74	253.18	228.94	192.19	185.94	151.65	134.35	122.21	113.43	59.29	30.19	15.21	7.63	6.95	5.00	4.06	3.59	3.34
256	244.68	230.89	218.82	198.19	166.52	146.82	129.31	120.41	113.02	61.62	31.44	15.97	8.05	7.19	5.15	4.12	3.61	3.36
512	225.20	218.65	212.66	201.56	182.39	153.25	126.69	117.80	112.19	64.22	34.13	17.41	8.82	7.58	5.35	4.23	3.66	3.38
1024	215.50	212.39	209.41	203.55	192.82	174.65	146.76	116.50	111.06	68.62	38.37	20.37	16.65	8.38	5.76	4.44	3.78	3.64
2048	114.04	113.23	112.42	110.79	107.74	102.05	92.40	77.67	61.44	74.56	45.84	25.59	13.59	10.03	6.62	4.88	4.00	3.55
4096	60.04	59.77	59.59	59.13	58.29	56.69	53.71	48.62	40.88	32.25	38.91	23.55	13.15	6.91	6.61	4.89	4.00	3.55
8192	33.02	32.96	32.90	32.79	32.55	32.09	31.21	29.60	26.80	22.56	17.74	20.55	12.44	6.89	5.11	4.92	4.02	3.57
16384	19.56	19.53	19.52	19.49	19.41	19.27	19.01	18.48	17.51	15.85	13.33	10.44	11.34	6.80	5.17	4.21	4.06	3.59
32768	12.81	12.82	12.82	12.83	12.81	12.76	12.67	12.49	12.15	11.51	10.43	8.77	6.82	6.71	5.28	4.32	3.75	3.63
65536	9.44	9.44	9.44	9.43	9.42	9.40	9.37	9.30	9.17	8.92	8.45	7.66	6.46	5.00	5.45	4.53	3.90	3.52
131072	7.76	7.76	7.76	7.76	7.75	7.75	7.73	7.70	7.65	7.55	7.34	6.95	6.30	5.31	4.08	4.31	3.83	3.51
262144	6.93	6.93	6.92	6.93	6.92	6.92	6.91	6.90	6.88	6.83	6.73	6.54	6.21	5.62	4.74	3.63	3.73	3.48
524288	6.50	6.50	6.51	6.50	6.50	6.50	6.50	6.49	6.48	6.46	6.41	6.32	6.15	5.83	5.28	4.45	3.39	3.44
1048576	6.29	6.29	6.29	6.29	6.29	6.28	6.28	6.29	6.28	6.27	6.25	6.20	6.11	5.94	5.64	5.11	4.30	3.28

SKINNY-ZOTR																		
M\AD	0	15	30	60	120	240	480	960	1920	3840	7680	15360	30720	61440	122880	245760	491520	983040
16	805.56	733.71	495.66	342.07	238.29	177.52	136.85	122.05	112.84	58.35	29.27	14.42	7.19	6.76	4.93	4.02	3.56	3.34
32	512.06	348.70	423.65	286.10	214.90	168.27	139.50	120.44	112.12	58.25	29.29	14.67	7.22	6.77	4.94	4.03	3.57	3.34
64	356.92	289.39	243.44	264.09	195.46	161.18	137.13	122.88	111.95	58.54	29.54	14.81	7.40	6.80	4.95	4.04	3.58	3.34
128	279.78	250.34	226.44	190.33	184.09	150.17	133.41	121.70	113.17	59.25	30.12	15.17	7.60	6.96	5.01	4.07	3.60	3.35
256	240.77	227.45	215.58	194.92	164.02	144.60	127.67	119.46	112.44	61.32	31.28	15.88	7.98	7.16	5.13	4.12	3.62	3.36
512	221.20	214.85	208.98	198.01	179.20	150.61	124.35	116.33	111.27	63.66	33.85	17.25	8.75	7.54	5.33	4.22	3.66	3.38
1024	211.42	208.30	205.40	199.69	189.25	171.25	143.94	114.20	109.54	67.65	37.83	20.07	16.50	8.31	5.72	4.42	3.77	3.63
2048	206.46	204.95	203.47	200.59	195.01	184.80	167.25	140.58	109.11	106.69	65.27	36.46	19.35	12.98	8.12	5.65	4.38	3.74
4096	107.99	107.60	107.22	106.44	104.94	102.03	96.70	87.51	73.57	57.03	55.58	33.64	18.77	9.90	8.15	5.68	4.41	3.76
8192	57.31	57.21	57.11	56.90	56.50	55.69	54.16	51.32	46.44	39.03	30.23	28.96	17.51	9.72	6.62	5.70	4.42	3.77
16384	31.95	31.93	31.91	31.85	31.73	31.50	31.05	30.20	28.66	25.93	21.80	16.86	15.67	9.40	6.63	4.98	4.46	3.79
32768	19.28	19.33	19.32	19.31	19.27	19.20	19.06	18.79	18.27	17.32	15.68	13.18	10.17	8.95	6.65	5.09	4.16	3.84
65536	13.00	13.00	13.00	12.99	12.98	12.96	12.91	12.82	12.64	12.28	11.65	10.54	8.87	6.82	6.69	5.27	4.32	3.75
131072	9.84	9.84	9.84	9.83	9.83	9.82	9.80	9.77	9.70	9.56	9.30	8.81	7.98	6.72	5.15	5.03	4.26	3.75
262144	8.27	8.27	8.27	8.27	8.27	8.26	8.26	8.24	8.21	8.15	8.03	7.82	7.41	6.71	5.65	4.32	4.17	3.75
524288	7.48	7.48	7.48	7.48	7.48	7.48	7.48	7.47	7.45	7.43	7.37	7.27	7.07	6.70	6.07	5.11	3.90	3.76
1048576	7.09	7.09	7.09	7.09	7.09	7.09	7.08	7.08	7.07	7.06	7.04	6.99	6.89	6.70	6.35	5.75	4.84	3.69

SKINNY-ΘCB3																		
M\AD	0	15	30	60	120	240	480	960	1920	3840	7680	15360	30720	61440	122880	245760	491520	983040
16	406.81	314.13	283.27	257.98	240.84	218.10	211.90	208.74	104.71	48.84	20.70	6.55	6.12	5.90	5.79	5.74	5.71	5.70
32	305.98	276.83	262.88	248.70	237.09	217.26	211.75	208.76	105.62	49.51	21.16	6.82	6.27	5.99	5.84	5.76	5.73	5.70
64	255.18	247.52	243.13	237.27	231.44	215.91	211.29	208.57	107.09	50.76	21.84	7.17	6.43	6.05	5.86	5.78	5.73	5.71
128	229.99	228.50	227.63	226.19	224.54	214.01	210.69	208.44	110.17	53.32	23.42	8.04	6.89	6.29	5.99	5.84	5.76	5.72
256	217.36	217.12	217.34	217.54	217.75	211.58	209.61	207.96	115.71	57.97	26.26	9.60	7.66	6.67	6.18	5.93	5.81	5.75
512	211.00	211.07	211.36	211.77	212.45	209.24	208.37	207.44	125.16	66.68	31.86	12.82	9.33	7.53	6.62	6.14	5.91	5.80
1024	14.08	17.03	19.94	25.47	35.66	50.51	75.24	106.86	71.97	40.60	19.35	6.72	6.23	5.97	5.84	5.77	5.73	5.71
2048	9.94	11.48	12.97	15.94	21.56	30.49	47.14	72.47	54.88	34.54	17.91	6.66	6.22	5.97	5.84	5.77	5.73	5.71
4096	7.81	8.61	9.38	10.89	13.87	18.80	28.59	45.44	38.15	27.09	15.80	6.56	6.19	5.97	5.84	5.77	5.74	5.72
8192	6.71	7.13	7.51	8.28	9.80	12.39	17.74	27.62	24.98	19.80	13.18	6.41	6.14	5.95	5.84	5.77	5.74	5.72
16384	6.16	6.37	6.57	6.96	7.73	9.05	11.85	17.23	16.33	14.07	10.62	6.22	6.06	5.92	5.83	5.77	5.74	5.72
32768	5.87	5.98	6.07	6.27	6.65	7.33	8.75	11.57	11.29	10.30	8.60	6.02	5.95	5.87	5.81	5.77	5.73	5.72
65536	5.73	5.78	5.82	5.92	6.12	6.45	7.17	8.62	8.55	8.09	7.27	5.87	5.85	5.81	5.78	5.75	5.73	5.71
131072	5.65	5.68	5.70	5.75	5.85	6.02	6.38	7.12	7.11	6.90	6.50	5.78	5.78	5.77	5.74	5.73	5.72	5.71
262144	5.63	5.65	5.66	5.68	5.73	5.82	6.00	6.37	6.38	6.28	6.08	5.70	5.70	5.70	5.71	5.70	5.70	5.70
524288	5.62	5.63	5.63	5.64	5.67	5.71	5.80	5.99	6.00	5.95	5.86	5.67	5.67	5.67	5.67	5.68	5.69	5.69
1048576	5.61	5.62	5.62	5.63	5.64	5.66	5.71	5.80	5.82	5.79	5.74	5.64	5.64	5.65	5.65	5.66	5.66	5.67

Table 5: Software performance in c/B of SKINNY-ZOCB/ZOTR/ΘCB3 on Haswell compiled using gcc, with various input length (in bytes)

SKINNY-ZOCB																	
M\AD	0	15	30	60	120	240	480	960	1920	3840	7680	15360	30720	61440	122880	245760	491520	983040
16	864.06	777.94	525.11	362.91	253.10	189.11	145.94	130.38	120.53	62.43	31.73	16.11	8.50	8.07	6.20	5.25	4.77	4.53
32	539.06	367.06	444.30	299.83	226.54	177.92	148.00	128.28	119.53	62.18	31.66	16.32	8.49	8.06	6.20	5.25	4.78	4.53
64	376.55	305.03	256.37	277.38	205.73	170.50	145.61	130.96	119.33	62.55	31.99	16.51	8.71	8.12	6.22	5.27	4.78	4.54
128	295.23	264.28	239.19	201.01	193.91	158.67	141.50	129.50	120.61	63.27	32.61	16.89	8.91	8.27	6.27	5.29	4.80	4.54
256	255.12	240.87	228.26	206.58	173.62	153.03	135.45	127.08	119.80	65.47	33.80	17.62	9.31	8.48	6.40	5.34	4.82	4.55
512	234.82	228.12	221.65	210.05	190.11	159.74	131.88	123.62	118.52	67.94	36.53	19.07	10.12	8.89	6.62	5.47	4.88	4.58
1024	224.44	221.18	218.03	212.00	200.88	181.81	152.80	121.17	116.52	72.06	40.67	22.03	18.32	9.69	7.03	5.68	4.99	4.85
2048	119.25	118.38	117.52	115.85	112.65	106.74	96.62	81.21	64.21	78.75	48.61	27.57	15.05	11.43	7.93	6.14	5.22	4.76
4096	63.78	63.55	63.31	62.87	61.97	60.24	57.09	51.68	43.42	34.26	41.47	25.46	14.60	8.11	7.92	6.15	5.23	4.76
8192	36.05	35.98	35.92	35.78	35.53	35.02	34.06	32.27	29.21	24.55	19.29	22.34	13.87	8.10	6.33	6.17	5.25	4.77
16384	22.17	22.16	22.12	22.09	22.01	21.86	21.55	20.95	19.86	17.97	15.11	11.80	12.77	8.04	6.40	5.43	5.29	4.80
32768	15.25	15.23	15.23	15.21	15.19	15.13	15.02	14.82	14.40	13.65	12.36	10.39	8.06	7.97	6.53	5.55	4.96	4.84
65536	11.80	11.80	11.80	11.80	11.79	11.76	11.72	11.64	11.47	11.16	10.57	9.58	8.06	6.21	6.72	5.77	5.12	4.73
131072	10.09	10.09	10.09	10.08	10.09	10.08	10.05	10.02	9.95	9.81	9.54	9.04	8.20	6.90	5.30	5.54	5.05	4.71
262144	9.26	9.26	9.26	9.25	9.25	9.25	9.24	9.20	9.19	9.11	8.97	8.73	8.30	7.52	6.33	4.84	4.94	4.68
524288	8.80	8.80	8.81	8.80	8.80	8.80	8.79	8.79	8.77	8.74	8.67	8.55	8.32	7.89	7.15	6.02	4.59	4.64
1048576	8.58	8.58	8.58	8.58	8.58	8.58	8.58	8.57	8.57	8.55	8.52	8.46	8.34	8.11	7.69	6.97	5.87	4.47

SKINNY-ZOTR																		
M\AD	0	15	30	60	120	240	480	960	1920	3840	7680	15360	30720	61440	122880	245760	491520	983040
16	863.31	777.47	524.84	362.80	253.03	189.12	145.80	130.29	120.51	62.39	31.73	16.11	8.50	8.06	6.17	5.26	4.77	4.53
32	535.52	364.67	443.19	298.91	225.99	177.61	147.87	128.22	119.50	62.14	31.67	16.32	8.49	8.06	6.17	5.26	4.77	4.53
64	376.15	304.85	256.09	277.48	205.30	170.18	145.53	130.93	119.33	62.52	31.99	16.51	8.71	8.11	6.20	5.27	4.78	4.54
128	295.47	264.50	239.31	201.23	194.17	158.78	141.60	129.57	120.65	63.25	32.61	16.89	8.91	8.27	6.25	5.30	4.79	4.54
256	254.98	240.90	228.17	206.51	173.60	152.85	135.39	127.05	119.79	65.39	33.78	17.61	9.31	8.47	6.38	5.35	4.82	4.55
512	234.50	227.79	221.46	209.88	189.93	159.61	131.70	123.52	118.43	67.86	36.51	19.06	10.11	8.88	6.59	5.47	4.87	4.58
1024	224.19	220.92	217.75	211.71	200.63	181.58	152.60	121.02	116.45	72.00	40.65	22.01	18.31	9.68	7.01	5.68	4.98	4.85
2048	218.98	217.40	215.83	212.75	206.86	196.01	177.41	149.09	115.72	113.43	69.64	39.31	21.29	14.64	9.54	6.97	5.63	4.96
4096	115.34	114.90	114.48	113.65	112.05	108.96	103.23	93.44	78.53	60.86	59.41	36.38	20.71	11.37	9.57	7.00	5.66	4.98
8192	62.07	61.95	61.84	61.61	61.17	60.31	58.64	55.57	50.29	42.27	32.70	31.45	19.38	11.17	7.94	7.02	5.68	4.99
16384	35.43	35.40	35.36	35.30	35.17	34.92	34.43	33.48	31.73	28.72	24.13	18.64	17.40	10.85	7.96	6.26	5.72	5.01
32768	22.14	22.14	22.14	22.12	22.08	21.99	21.85	21.54	20.95	19.84	17.97	15.11	11.63	10.36	7.97	6.36	5.39	5.06
65536	15.48	15.48	15.47	15.47	15.47	15.45	15.40	15.28	15.07	14.66	13.89	12.58	10.58	8.11	8.01	6.55	5.55	4.96
131072	12.19	12.19	12.20	12.20	12.20	12.19	12.16	12.12	12.03	11.86	11.53	10.94	9.91	8.34	6.39	6.28	5.49	4.96
262144	10.56	10.57	10.57	10.56	10.55	10.55	10.54	10.53	10.49	10.42	10.27	9.98	9.47	8.59	7.23	5.52	5.39	4.96
524288	9.73	9.73	9.72	9.72	9.72	9.73	9.72	9.71	9.69	9.65	9.59	9.44	9.19	8.71	7.89	6.64	5.06	4.95
1048576	9.30	9.30	9.31	9.31	9.31	9.31	9.31	9.31	9.30	9.26	9.23	9.16	9.03	8.78	8.34	7.56	6.37	4.86

SKINNY-ΘCB3																		
M\AD	0	15	30	60	120	240	480	960	1920	3840	7680	15360	30720	61440	122880	245760	491520	983040
16	432.22	334.48	298.55	268.91	248.97	223.54	216.94	213.35	107.39	51.07	22.70	8.44	8.04	7.84	7.75	7.71	7.69	7.68
32	323.59	293.84	277.02	259.59	245.43	223.05	216.88	213.38	108.28	51.77	23.10	8.66	8.15	7.90	7.78	7.72	7.70	7.68
64	269.36	261.94	255.90	248.06	240.11	222.25	216.77	213.43	110.03	53.10	23.89	9.09	8.37	8.01	7.84	7.75	7.71	7.69
128	242.15	240.92	239.33	236.79	233.65	221.01	216.60	213.53	113.31	55.72	25.47	9.94	8.80	8.22	7.95	7.80	7.73	7.70
256	228.59	228.68	228.44	228.01	227.31	219.51	216.34	213.70	119.36	60.74	28.52	11.62	9.65	8.65	8.16	7.91	7.79	7.73
512	221.92	222.10	222.18	222.27	222.46	218.02	216.01	213.93	129.44	69.84	34.36	14.90	11.34	9.51	8.59	8.13	7.90	7.79
1024	16.46	19.54	22.56	28.17	38.45	53.30	78.31	110.12	74.62	42.91	21.40	8.64	8.16	7.91	7.79	7.73	7.70	7.69
2048	12.16	13.74	15.32	18.26	24.00	32.97	49.82	75.36	57.37	36.80	19.98	8.59	8.15	7.91	7.79	7.74	7.70	7.69
4096	9.95	10.79	11.59	13.13	16.14	21.10	31.02	48.00	40.50	29.31	17.86	8.50	8.13	7.90	7.79	7.73	7.70	7.68
8192	8.84	9.27	9.66	10.44	11.99	14.59	20.00	29.97	27.22	21.96	15.25	8.38	8.10	7.90	7.80	7.73	7.70	7.69
16384	8.27	8.49	8.68	9.07	9.86	11.20	14.03	19.46	18.50	16.20	12.69	8.21	8.03	7.88	7.79	7.73	7.70	7.69
32768	7.98	8.09	8.19	8.39	8.79	9.46	10.91	13.75	13.41	12.41	10.68	8.04	7.95	7.87	7.78	7.74	7.71	7.69
65536	7.84	7.90	7.95	8.05	8.25	8.58	9.32	10.79	10.68	10.22	9.37	7.92	7.89	7.83	7.78	7.74	7.71	7.69
131072	7.78	7.81	7.84	7.89	7.99	8.16	8.52	9.27	9.24	9.03	8.62	7.85	7.84	7.81	7.78	7.75	7.72	7.70
262144	7.76	7.77	7.79	7.81	7.86	7.95	8.13	8.51	8.51	8.40	8.20	7.81	7.80	7.79	7.77	7.75	7.73	7.71
524288	7.75	7.75	7.76	7.77	7.80	7.84	7.93	8.13	8.14	8.08	7.98	7.78	7.78	7.78	7.76	7.75	7.73	7.71
1048576	7.75	7.75	7.75	7.76	7.77	7.79	7.84	7.93	7.94	7.92	7.87	7.76	7.77	7.76	7.76	7.75	7.74	7.72