Parallelizable Authenticated Encryption with Small State Size

Akiko Inoue and Kazuhiko Minematsu

NEC Corporation, Kawasaki, Japan
a_inoue@nec.com, k-minematsu@nec.com

Abstract. Authenticated encryption (AE) is a symmetric-key encryption function that provides confidentiality and authenticity of a message. One of the evaluation criteria for AE is state size, which is memory size needed for encryption. State size is especially important when cryptosystem is implemented in constrained devices, while trivial reduction by using a small primitive is not generally acceptable as it leads to a degraded security.

In these days, the state size of AE has been very actively studied and a number of small-state AE schemes have been proposed, but they are inherently serial. It would be a natural question if we come up with a parallelizable AE with a smaller state size than the state-of-the-art.

In this paper, we study the seminal OCB mode for parallelizable AE and propose a method to reduce its state size without losing the bit security of it. More precisely, while (the most small-state variant of) OCB has $3n$-bit state, by carefully treating the checksum that is halved, we can achieve $2.5n$-bit state, while keeping the $n/2$-bit security as original. We also propose an inverse-free variant of it based on OTR. While the original OTR has $4n$-bit state, ours has $3.5n$-bit state. To our knowledge these numbers are the smallest ones achieved by the blockcipher modes for parallel AE and inverse-free parallel AE.

Keywords: Authenticated encryption · State size · OCB · OTR · Phash.

1 Introduction

Authenticated encryption (AE) is a symmetric-key cryptographic scheme that provides confidentiality and authenticity of a message simultaneously. For example, GCM [19] and CCM [20] are the current NIST standard AE modes and used in TLS [40,37] and many other protocols. Among many criteria, the *state size* of AE has become an important one as well as the speed, since it is a key factor determining the size of hardware implementation. It is the memory size needed to implement the cryptosystem, in which we exclude core implementation (*e.g.*blockcipher) including key register. Thus we only count the memory size for the implementation of the mode of operation itself.

With the rise of lightweight cryptography, a number of small-state AE schemes have been proposed. CLOC and SILC proposed by Iwata *et al.* [25,26] in 2014

have $2n$-bit state using n-bit blockcipher. In 2017, Chakraborti *et al.* proposed COFB [16] which has $1.5n$-bit state size. Finally, Naito *et al.* proposed SAEB [36] and achieved n-bit state size which is essentially minimum as a mode of n-bit blockcipher. In the realm of permutation-based cryptography, the sponge AE schemes are known to have small state size [13]. However, these AEs are essentially serial to achieve small state size. Ideally, we want an AE scheme to perform good on a wide range of platforms, and parallelizability is very effective particularly for software on high-end to middle-end platforms. For example, AES runs about $4 \sim 8$ times faster in parallel on CPUs with AES instructions (AESNI), and the bitslice implementation of lightweight blockciphers typically run significantly (often by a order of magnitude) faster than the single-block implementation [12,29] on modern CPUs with SIMD instructions. This observation and the current research trend in serial AEs of small-state size suggest a natural question: *can we reduce the state size of a parallel AE?*

To answer the above question, we study the seminal OCB mode of operation from the state size perspective. OCB has been known to be the most efficient parallel AE. It consists of three versions, namely OCB1 [39], OCB2 [38] and OCB3 [27], and the latest OCB3 is in the final portfolio of CAESAR competition and was standardized in RFC [1]. In the submissions to the NIST Lightweight Cryptography Standardization project [2], the structure of OCB has been adopted by a number of schemes. Among the three versions of OCB, OCB2 has the smallest state size (*e.g.* OCB3 needs around n blocks in memory for internal mask generation). Note that OCB2 has been shown to be insecure by Inoue *et al.* [23]; we employed the fix of OCB2 suggested in [23] called OCB2f (for convention, we use "OCB2" to mean this fix unless otherwise stated). The original OCB2 needs $3n$-bit state, consisting of the blockcipher state and the mask applied to the blockcipher, and the checksum value to create the authentication tag. The last one is essentially a sum of the n-bit plaintext blocks.

We propose a way to reduce OCB's state size. In our method, we halve the length of checksum and we can reduce $0.5n$-bit state size from the original OCB. An important feature of our method is that it does not lose efficiency (the number of blockcipher calls needed) nor the essential bit security of OCB. When our method is instantiated with n-bit blockcipher, it needs $m + O(1)$ blockcipher calls to encrypt m-block input (this feature is called rate-1). Moreover, it has $n/2$-bit security despite of the trade-off relationship between the state size and security. We find that halving the checksum value does not harm the bit security of OCB2, and with a careful (though simple) handling of last block, we actually achieve $2.5n$-bit state size with our proposal called OCB-hc (for half-checksum).

One of the factors that increases the implementation size is the need of blockcipher inverse in its circuit. OCB needs the inverse, while Minematsu's OTR [31] derived from OCB is inverse-free. The state size of OTR is $4n$ bits as its operates on $2n$-bit blocks, thus larger than OCB2. However, thanks to the inverse-freeness, the total implementation size is expected to be smaller, which is also beneficial to high-throughput implementation (see the results of ATHENA benchmark [1]

[1] https://cryptography.gmu.edu/athena

Table 1. Comparison of existing schemes and ours. State size excludes the key register. Rate is the number of input blocks processed in one primitive call.

Scheme	State size (bit)	Security	Rate	Inverse free	Parallelizable
OCB [39,38,27]	$3n$	$O(2^{n/2})$	1	-	✓
OTR [31]	$4n$	$O(2^{n/2})$	1	✓	✓
CLOC, SILC [25,26]	$2n$	$O(2^{n/2})$	1/2	✓	-
COFB [16]	$1.5n$	$O(2^{n/2})$	1	✓	-
SAEB [36]	n	$O(2^{n/2})$	1/2	✓	-
OCB-hc (Ours)	$2.5n$	$O(2^{n/2})$	1	-	✓
OTR-hc (Ours)	$3.5n$	$O(2^{n/2})$	1	✓	✓

and [41]). Using a similar technique as OCB-hc, we propose OTR-hc that has $3.5n$-bit state with $n/2$-bit security.

We remark that improving OCB in any metric without losing the essential properties is already very tough. All versions of OCB have been extensively studied from various perspective, such as the provable security perspective [14,7] or the efficiency of mask generation scheme [22,33], or the misuse resistance [8,4] or the security beyond $O(2^{n/2})$ queries [21]. However, its general structure which determines the state size profile is already considered to be optimal since the inception. To the best of our knowledge, there is no previous work to reduce the state size, and $2.5n$-bit state of OCB-hc is the smallest among the known parallel AE modes. Likewise, $3.5n$-bit state size of OTR-hc is the smallest among the known inverse-free, parallel AE modes, to our knowledge. See Table 1.

Our technique can be applied to some variants of OCB as well, such as OPP [22] which has a much larger block size than OCB-AES and thus the gain is larger.

2 Preliminaries

2.1 Notation

Let $\mathbb{N}$ be the set of natural numbers. For $n \in \mathbb{N}$, we define $\{0,1\}^n$ as the set of n-bit strings and $\{0,1\}^*$ as the set of all binary strings, including the empty string ε. For $A, B \in \{0,1\}^*$, $A \,\|\, B$ denotes the concatenation of A and B. The bit length of a string A is denoted by $|A|$, and $|A|_n := \lceil |A|/n \rceil$. Dividing a string A into blocks of n bits is denoted by $A[1] \,\|\, \cdots \,\|\, A[m] \xleftarrow{n} A$, where $m = |A|_n$ and $|A[i]| = n$, $|A[m]| \leq n$ for $1 \leq i \leq m-1$. For $t \in \mathbb{N}$ and $t \leq |A|$, $\mathtt{msb}_t(A)$ denotes the first t bits of A and $\mathtt{lsb}_t(A)$ denotes the last t bits of A. A sequence of i zeros (ones) is written as 0^i (1^i). When $|A| = n' < n$, we define $\mathtt{ozp}(A) := A \,\|\, 10^{n-n'-1}$, where $10^0 = 1$. When $|A| = n' = n$, $\mathtt{ozp}(A) := A$. When the element K is uniformly and randomly chosen from the set $\mathcal{K}$, it is denoted by $K \xleftarrow{\$} \mathcal{K}$.

2.2 (Tweakable) Blockcipher

Let $\mathcal{K}$ and $\mathcal{M}$ be the set of keys and messages, respectively. Let $\mathcal{T}$ be the set of tweaks, where a tweak is a public parameter. A tweakable blockcipher (TBC) [28] is a function $\widetilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ s.t. $\widetilde{E}(K, T, \cdot)$ is a permutation on $\mathcal{M}$ for $\forall (K, T) \in \mathcal{K} \times \mathcal{T}$. It is also denoted by $\widetilde{E}_K^T$, $\widetilde{E}^T$ or $\widetilde{E}$, where $K \in \mathcal{K}$ and $T \in \mathcal{T}$. If $\mathcal{T}$ is singleton (and we thus omit it from the notation) it means a plain blockcipher. Namely, a blockcipher E is defined as $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ s.t. $E(K, \cdot)$ is a permutation on $\mathcal{M}$ for $\forall K \in \mathcal{K}$ and also denoted by E_K or E.

Let $\mathsf{Perm}(n)$ denote the set of all permutations on $\{0,1\}^n$. A tweakable permutation is a function $\pi : \{0,1\}^t \times \{0,1\}^n \to \{0,1\}^n$ s.t. for $\forall T \in \{0,1\}^t$, $\pi(T, \cdot) \in \mathsf{Perm}(n)$. Let $\widetilde{\mathsf{Perm}}(t, n)$ denote the set of above all functions π. Let P s.t. $\mathsf{P} \xleftarrow{\$} \mathsf{Perm}(n)$ be a uniform random permutation (URP) and $\widetilde{\mathsf{P}}$ s.t. $\widetilde{\mathsf{P}} \xleftarrow{\$} \widetilde{\mathsf{Perm}}(t, n)$ be a tweakable URP (TURP). A blockcipher E or a TBC $\widetilde{E}$ is said to be secure if it is computationally hard to distinguish from the ideal primitive with oracle access. More precisely, let $\mathcal{A}$ be an adversary who (possibly adaptively) queries to an oracle O and subsequently outputs a bit. We write $\Pr[\mathcal{A}^O \to 1]$ to denote the probability that this bit is 1. We define the notions of advantage of $\mathcal{A}$ as

$$
\begin{aligned}
\mathbf{Adv}_E^{\mathrm{prp}}(\mathcal{A}) &:= |\Pr[\mathcal{A}^E \to 1] - \Pr[\mathcal{A}^{\mathsf{P}} \to 1]|, \\
\mathbf{Adv}_E^{\mathrm{sprp}}(\mathcal{A}^{\pm}) &:= |\Pr[(\mathcal{A}^{\pm})^{E, E^{-1}} \to 1] - \Pr[(\mathcal{A}^{\pm})^{\mathsf{P}, \mathsf{P}^{-1}} \to 1]|, \\
\mathbf{Adv}_{\widetilde{E}}^{\mathrm{tprp}}(\mathcal{A}) &:= |\Pr[\mathcal{A}^{\widetilde{E}} \to 1] - \Pr[\mathcal{A}^{\widetilde{\mathsf{P}}} \to 1]|, \\
\mathbf{Adv}_{\widetilde{E}}^{\mathrm{tsprp}}(\mathcal{A}^{\pm}) &:= |\Pr[(\mathcal{A}^{\pm})^{\widetilde{E}, \widetilde{E}^{-1}} \to 1] - \Pr[(\mathcal{A}^{\pm})^{\widetilde{\mathsf{P}}, \widetilde{\mathsf{P}}^{-1}} \to 1]|,
\end{aligned}
$$

where the first and the third notions are for adversaries with encryption oracle (thus chosen-plaintext queries), and the second and the fourth are for adversaries with encryption and decryption oracles (thus chosen-ciphertext queries).

When the advantage is sufficiently low, E or $\widetilde{E}$ is said to be secure against the underlying adversary.

2.3 Authenticated Encryption

Let $\mathcal{K}$, $\mathcal{M}_{ae}$ and $\mathcal{N}_{ae}$ be the set of keys, messages and nonce, respectively. Let $\mathcal{A}_{ae}$ be the set of associated data (AD), which is data not encrypted but authenticated, and it can be empty. For convention, by saying AE we may mean AEAD. If we want to explicitly mean AE with no AD, (*i.e.* $\mathcal{A}_{ae}$ is empty) we call it *plain* AE. Suppose AE.$\mathcal{E}$ and AE.$\mathcal{D}$ as an encryption function and a decryption function of AE, respectively. We suppose that AE.$\mathcal{E}$ and AE.$\mathcal{D}$ share the key $K \in \mathcal{K}$ as input. For encryption, the sender inputs a nonce $N \in \mathcal{N}_{ae}$, an associated data $A \in \mathcal{A}_{ae}$ and a message $M \in \mathcal{M}_{ae}$ to AE.$\mathcal{E}_K$. Then she gets a ciphertext $C \in \mathcal{M}_{ae}$ and a tag $T \in \{0,1\}^\tau$ as the output, where τ is the length of tag. The sender sends the tuple (N, A, C, T), and the receiver inputs them

to AE.$\mathcal{D}_K$ for decryption. AE.$\mathcal{D}_K$ outputs a message M' if the verification is success, otherwise outputs $\perp$, which means that the verification failed.

The security of AE scheme can be evaluated by two criteria: privacy and authenticity. Following the existing work [11,39,38], we use the term *privacy* to mean confidentiality. For privacy, we define the privacy advantage as the probability that the adversary successfully distinguishes the encryption function of AE from the *random-bit oracle*, $\$(*,*,*)$, which returns random bits of length $|M|+|T|$ for any query (N,A,M): $\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{AE}}(\mathcal{A}) := |\Pr[\mathcal{A}^{\mathrm{AE}.\mathcal{E}} \to 1] - \Pr[\mathcal{A}^{\$} \to 1]|$. Here, we assume $\mathcal{A}$ is nonce-respecting, that is, $\mathcal{A}$ does not repeat nonce N in the encryption queries. For authenticity, we define the authenticity advantage as the probability that the adversary creates a successful forgery by accessing encryption and decryption functions of AE. It is defined as $\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{AE}}(\mathcal{A}) := \Pr[\mathcal{A}^{\mathrm{AE}.\mathcal{E},\mathrm{AE}.\mathcal{D}}\text{forges.}]$, where $\mathcal{A}^{\mathrm{AE}.\mathcal{E},\mathrm{AE}.\mathcal{D}}$ forges if $\mathcal{A}$ receives $M' \neq \perp$ from AE.$\mathcal{D}$ by querying (N',A',C',T') while (N',A',M') has never been queried to AE.$\mathcal{E}$. As well as the privacy case, $\mathcal{A}$ is assumed to be nonce-respecting in its encryption queries, however no restriction on the nonce values in the decryption queries.

2.4 Computation on Galois Field

Let $\mathbb{F}_{p^n}$ be a finite filed, where characteristic p is prime and extension degree $n \in \mathbb{N}$. We focus on the case $n = 128$. Following [38,24], we use the lexicographically-first polynomial for defining the field and thus $\mathbb{F}_{2^{128}} := \mathbb{F}_2[x]/(x^{128}+x^7+x^2+x+1)$ and obtain $\mathbb{F}_{2^{128}} = \langle x \rangle$. We regard an element of $\mathbb{F}_{2^{128}}$ as a polynomial of x. For $\forall a \in \{0,1\}^{128}$, we also regard it as a coefficient vector of an element in $\mathbb{F}_{2^{128}}$. Thus, the primitive root x is interpreted as 2 in the decimal representation. For $a \in \mathbb{F}_{2^{128}}$, let $2a$ denote a multiplication by x and a, which is called doubling [38]. In $\mathbb{F}_{2^{128}}$, $2a := (a \ll 1)$ if $\mathtt{msb}_1(a) = 0$ and $2a := (a \ll 1) \oplus (0^{120}10^41^3)$ if $\mathtt{msb}_1(a) = 1$, where $(a \ll 1)$ is the left-shift of one bit. For $c \in \mathbb{N}$, we can calculate $2^c a$ by repeating doubling of a for c-times, and $3a = 2a \oplus a$.

3 Review of OCB and OTR

3.1 OCB

OCB is a blockcipher mode of operation for AE scheme proposed at [39,38,27]. It is parallelizable, and is a rate-1 scheme which needs one blockcipher call to process one message block. It also has provable security based on the pseudo-randomness of underlying blockcipher. The security bound of OCB is $O(\sigma^2/2^n)$, which is called birthday-bound security, where σ is the number of access to n-bit blockcipher. OCB encrypts a message in a mode similar to ECB, where the blockcipher has input and output masks, and computes the sum of message blocks, called checksum. The authentication tag is an encryption of the checksum. Although OCB was initially proposed as a plain AE [39], it can be converted into AEAD by using PMAC [38] or Phash [27] for AD and taking the XOR of the output and the tag of (plain-AE) OCB. There are three versions for OCB:

OCB1 [39], OCB2 [38], OCB3 [27]. Among them, OCB2 has the smallest state size of $3n$ bits, consisting of n-bit memory for processing of one message block, the value of the mask, and the checksum. As described before, since OCB2 has shown to be insecure by Inoue *et al.* [23], this paper focuses on the fix suggested by [23] called OCB2f, which has the same $3n$-bit state. We simply call it OCB2 or even OCB as the version of OCB that we study, if no confusion is possible. OBC2 can be interpreted as a TBC mode for AE, which we call ΘCB. The TBC used in ΘCB is a blockcipher mode called XEX*.

Let us review the specific (information-theoretic) security bound of OCB2 when it is instantiated with an n-bit URP P. Throughout the paper, we use a subscript to denote the underlying component, hence $\mathrm{OCB2}_{\mathsf{P}}$ is the target scheme. We write $\Theta\mathrm{CB}_{\widetilde{\mathsf{P}}}$ to denote ΘCB using TURP $\widetilde{\mathsf{P}}$. For n-bit tag case, and for the privacy-adversary $\mathcal{A}$ and the authenticity-adversary $\mathcal{A}^{\pm}$, the security bounds of $\mathrm{OCB2}_{\mathsf{P}}$ ($\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{OCB2}_{\mathsf{P}}}(\mathcal{A})$, $\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{OCB2}_{\mathsf{P}}}(\mathcal{A}^{\pm})$) are given as follows [38,30,23]:

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{OCB2}_{\mathsf{P}}}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{tprp}}_{\mathrm{XEX}^*_{\mathsf{P}}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{priv}}_{\Theta\mathrm{CB}_{\widetilde{\mathsf{P}}}}(\mathcal{A}) \le \frac{4.5\sigma^2_{\mathrm{priv}}}{2^n} + 0,$$
$$\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{OCB2}_{\mathsf{P}}}(\mathcal{A}^{\pm}) \le \mathbf{Adv}^{\mathrm{tsprp}}_{\mathrm{XEX}^*_{\mathsf{P}}}(\mathcal{B}^{\pm}) + \mathbf{Adv}^{\mathrm{auth}}_{\Theta\mathrm{CB}_{\widetilde{\mathsf{P}}}}(\mathcal{A}^{\pm}) \le \frac{4.5\sigma^2_{\mathrm{auth}}}{2^n} + \frac{q_d}{2^n - 1},$$

where $\mathcal{B}$ (resp. $\mathcal{B}^{\pm}$) is the adversary performing chosen-plaintext attack (resp. chosen-ciphertext attack), σ_{priv} (resp. σ_{auth}) is the total number of queried blocks in privacy (resp. authenticity) game and q_d is the number of queries to verification (decryption) oracle. Since OCB3 can be also interpreted as a TBC mode, we can derive similar security bounds to OCB2 as above [27].

3.2 OTR

OTR is an AEAD blockcipher mode of operation proposed by Minematsu [31]. It is a parallelizable, rate-1 scheme. Whereas OCB needs blockcipher decryption for the entire decryption, OTR does not need it for both encryption and decryption, hence it is called inverse-free. As well as OCB, it has provable security based on the pseudorandomness of blockcipher, with security bound of $O(\sigma^2/2^n)$, where σ is the number of access to n-bit blockcipher [2]. OTR encrypts a message by using two-round Feistel permutation based on a blockcipher with an input mask, and computes the checksum as a sum of even-numbered message blocks. The authentication tag is an encryption of the checksum. The state size of OTR is $4n$ bits. It is composed of $2n$-bit memory for processing two message blocks (*i.e.* one Feistel chunk), and each n-bit memory for the value of the mask and the checksum. As well as OCB, OTR can be interpreted as a mode of TBC, which we call ΘTR (originally $\mathbb{OTR}$). The TBC used in ΘTR is a blockcipher mode called XE [38]. The security bound of $\mathrm{OTR}_{\mathsf{P}}$ can be bounded by a hybrid argument similar to $\mathrm{OCB}_{\mathsf{P}}$. A tweakable uniform random function (TURF) is

[2] Bost and Sanders [15] pointed a problem of the first version of OTR [31] regarding its instantiation of XE. Therefore we here refer OTR of the fixed versions [32].

denoted by $\widetilde{\mathsf{R}} : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$, where $\mathcal{T}$ is the same tweak space as XE. It is essentially a random function on the whole input domain.

For n-bit tag and for the privacy-adversary $\mathcal{A}$ and the authenticity-adversary $\mathcal{A}^{\pm}$, the security bounds of $\mathrm{OTR}_{\mathsf{P}}$ ($\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{OTR}_{\mathsf{P}}}(\mathcal{A})$, $\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{OTR}_{\mathsf{P}}}(\mathcal{A}^{\pm})$) are given as follows:

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{OTR}_{\mathsf{P}}}(\mathcal{A}) \le \mathbf{Adv}^{\mathtt{cpa}}_{\mathrm{XE}_{\mathsf{P}},\widetilde{\mathsf{R}}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{priv}}_{\Theta\mathrm{TR}_{\widetilde{\mathsf{R}}}}(\mathcal{A}) \le \frac{6\sigma^2_{\mathrm{priv}}}{2^n} + 0,$$
$$\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{OTR}_{\mathsf{P}}}(\mathcal{A}^{\pm}) \le \mathbf{Adv}^{\mathtt{cpa}}_{\mathrm{XE}_{\mathsf{P}},\widetilde{\mathsf{R}}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{auth}}_{\Theta\mathrm{TR}_{\widetilde{\mathsf{R}}}}(\mathcal{A}^{\pm}) \le \frac{6\sigma^2_{\mathrm{auth}}}{2^n} + \frac{q_d}{2^n},$$

where $\mathbf{Adv}^{\mathtt{cpa}}_{\mathrm{XE}_{\mathsf{P}},\widetilde{\mathsf{R}}}(\mathcal{B})$ is the probability which the adversary $\mathcal{B}$ performing chosen-plaintext attack can distinguish XE_{P} from $\widetilde{\mathsf{R}}$. The parameter σ_{priv} (resp. σ_{auth}) is the total number of queried blocks in privacy (resp. authenticity) game and q_d is the number of queries to the decryption oracle.

4 Our Proposals

4.1 Overview

As we mentioned in Section 3, the security bounds of OCB and OTR are evaluated using the hybrid argument: the bound of OCB is a sum of the bound of XEX* and that of ΘCB. Similarly, the bound of OTR is a sum of the bound of XE and that of ΘTR. One can find that ΘCB and ΘTR have beyond-birthday-bound security (namely perfect privacy and n-bit authenticity), however the total security of OCB and OTR are $n/2$ bits because of the birthday bounds of XEX* and XE. This gap implies a potential improvement in size, by trading the state size of ΘCB and ΘTR for security, while maintaining the overall $n/2$-bit security of OCB and OTR. We found that such a trading-off is indeed possible by reducing the length of checksum by $n/2$ bits, which we call half-checksum method.

Actually, this gap has been exploited in the literature. For example, Naito's XKX [35,34] provides a beyond-birthday-bound secure implementation of TBC and he proposed it to be used within a mode similar to ΘCB so that the resulting AE has beyond-birthday-bound security.

4.2 OCB-hc

We apply the half-checksum method mentioned in Section 4.1 to OCB. The resultant scheme is denoted by OCB-hc. While we first propose OCB-hc as a plain AE with $n/2$-bit tag length, we will extend it to an AEAD in Section 5. In the following, we fix the tag length to be $n/2$ bits as it is essentially minimum to achieve $n/2$-bit security. In case a longer tag is required, Section 5 will also provide an extension to the case of the arbitrary tag length up to n bits.

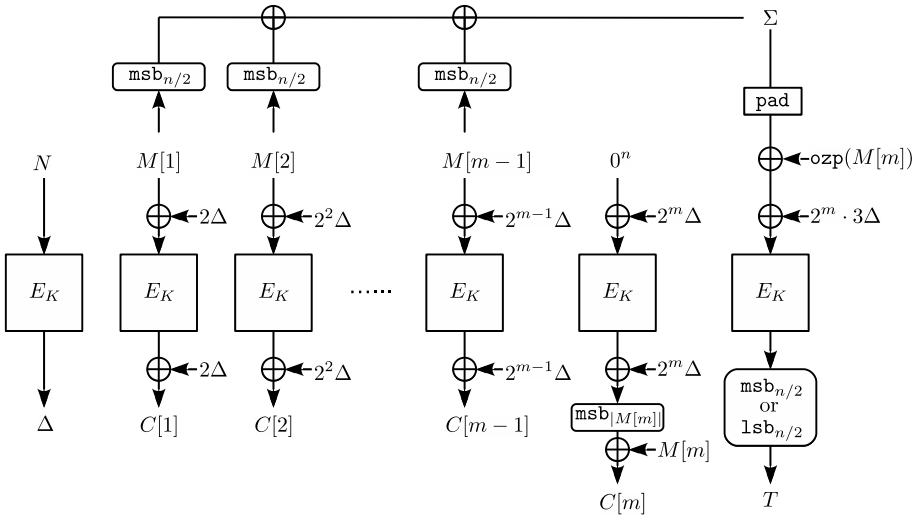

Fig. 1. The encryption of OCB-hc$_{E_K}$, where E_K is any n-bit blockcipher. The function pad denotes the zero padding to n bits.

Specification. We show OCB-hc in Fig. 1 and Fig. 2. As mentioned, the tag is $n/2$ bits and AD is empty. Let E_K be an n-bit blockcipher. We define the encryption function of OCB-hc$_{E_K}$ as OCB-hc.$\mathcal{E}_{E_K} : (N, M) \mapsto (C, T)$, where $(N, M) \in \{0,1\}^n \times \{0,1\}^*$ and $(C, T) \in \{0,1\}^* \times \{0,1\}^{n/2}$. We also define the decryption function as OCB-hc.$\mathcal{D}_{E_K} : (N, C, T) \mapsto M$ or $\perp$, where $(N, C, T) \in \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^{n/2}$ and $M \in \{0,1\}^*$. The structure of OCB-hc is generally the same as OCB, except that it computes the $n/2$-bit checksum of message blocks (say the first $n/2$ bits; in fact any bits would work fine). This is also different in the last message block $M[m]$, which may be partial. Following OCB, we take an XOR of $M[m]$ and the checksum padded to n bits, which is needed for security.

State size. Since the checksum is halved, it is easy to see that the state size of OCB-hc is reduced to $2.5n$ bits until the last message block. If $0 < M[m] \leq n/2$, the state size remains $2.5n$ bits since the checksum needs only $n/2$-bit memory. However if $n/2 < M[m] \leq n$, the state size seemingly increases to at most $3n$ bits, implying no gain. We can avoid this by only changing the computation procedure described above: we add $M[m]$ (more precisely, ozp$(M[m])$) not to the checksum, but to the mask used in the last block for the tag (See line 10, 11 in Algorithm: OCB-hc.$\mathcal{E}$ and Algorithm: OCB-hc.$\mathcal{D}$ in Fig.2). This will not change the algorithm. Since the mask is consistently n bits, this will not increase the size. Therefore, OCB-hc works with $2.5n$-bit state for any plaintext.

Algorithm: OCB-hc.$\mathcal{E}_{E_K}(N, M)$	**Algorithm:** OCB-hc.$\mathcal{D}_{E_K}(N, C, T)$
1. $M[1] \| \cdots \| M[m-1] \| M[m] \xleftarrow{n} M$	1. $C[1] \| \cdots \| C[m-1] \| C[m] \xleftarrow{n} C$
2. $\Delta \leftarrow 2E_K(N)$, $\Sigma \leftarrow 0^{n/2}$	2. $\Delta \leftarrow 2E_K(N)$, $\Sigma \leftarrow 0^{n/2}$
3. **for** $i \leftarrow 1$ **to** $m-1$ **do**	3. **for** $i \leftarrow 1$ **to** $m-1$ **do**
4. $\quad C[i] \leftarrow E_K(M[i] \oplus \Delta) \oplus \Delta$	4. $\quad M[i] \leftarrow D_K(C[i] \oplus \Delta) \oplus \Delta$
5. $\quad \Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(M[i])$	5. $\quad \Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(M[i])$
6. $\quad \Delta \leftarrow 2\Delta$	6. $\quad \Delta \leftarrow 2\Delta$
7. Pad $\leftarrow E_K(0^n \oplus \Delta) \oplus \Delta$	7. Pad $\leftarrow E_K(0^n \oplus \Delta) \oplus \Delta$
8. $\Delta \leftarrow 3\Delta$	8. $\Delta \leftarrow 3\Delta$
9. $C[m] \leftarrow \mathtt{msb}_{\lvert M[m] \rvert}(\text{Pad} \oplus \mathtt{ozp}(M[m]))$	9. $M[m] \leftarrow \mathtt{msb}_{\lvert C[m] \rvert}(\text{Pad}) \oplus C[m]$
10. $\Delta \leftarrow \Delta \oplus \mathtt{ozp}(M[m])$	10. $\Delta \leftarrow \Delta \oplus \mathtt{ozp}(M[m])$
11. Tag $\leftarrow E_K(\Sigma \| 0^{n/2} \oplus \Delta)$	11. Tag $\leftarrow E_K(\Sigma \| 0^{n/2} \oplus \Delta)$
12. **if** $\lvert M[m] \rvert = n$ **then** $T \leftarrow \mathtt{msb}_{n/2}(\text{Tag})$	12. **if** $\lvert M[m] = n$ **then** $T' \leftarrow \mathtt{msb}_{n/2}(\text{Tag})$
13. **else** $T \leftarrow \mathtt{lsb}_{n/2}(\text{Tag})$	13. **else** $T' \leftarrow \mathtt{lsb}_{n/2}(\text{Tag})$
14. **return** $(C[1] \| \cdots \| C[m], T)$	14. **if** $T = T'$ **then return** $M[1] \| \cdots \| M[m]$
	15. **else return** $\perp$

Fig. 2. The algorithm of OCB-hc. E_K is any n-bit blockcipher, and D_K is the decryption of E_K.

4.3 Security of OCB-hc

The security bounds of OCB-hc are shown below. We assume the underlying blockcipher is an n-bit URP, P. When the underlying blockcipher is a PRP, the security bounds are derived from ours using a standard technique [9], thus we omitted.

Theorem 1.

$$\mathbf{Adv}^{\text{priv}}_{\text{OCB-hc}\mathsf{P}}(\mathcal{A}) \le \frac{4.5\sigma^2_{\text{priv}}}{2^n}, \ \mathbf{Adv}^{\text{auth}}_{\text{OCB-hc}\mathsf{P}}(\mathcal{A}^{\pm}) \le \frac{4.5\sigma^2_{\text{auth}}}{2^n} + \frac{4q_d}{2^{n/2}},$$

where $\mathcal{A}$, $\mathcal{A}^{\pm}$ are the adversaries against OCB-hc$\mathsf{P}$ *and σ_{priv}, σ_{auth} and q_d are the parameters for $\mathcal{A}$ and $\mathcal{A}^{\pm}$. The parameter σ_{priv} (resp. σ_{auth}) is the number of accesses to P in privacy (resp. authenticity) game. The parameter q_d is the number of queries to the decryption oracle in authenticity game.*

Proof. Let $i \in \mathbb{N}$, $j \in \{0, 1, 2, 3\}$. We define two TBCs XEX_{E_K} and XE_{E_K} as follows.

$$\text{XEX}^{N,i}_{E_K}(M) = E_K(M \oplus 2^i E_K(N)) \oplus 2^i E_K(N),$$
$$\text{XE}^{N,i,j}_{E_K}(M) = E_K(M \oplus 2^i 3^j E_K(N)).$$

Then we combine them to one TBC denoted by $\text{XEX}^*_{E_K}$. $\text{XEX}^{*N,b,i,j}_{E_K}(M) = \text{XEX}^{N,i}_{E_K}(M)$ if $b = 1$, $\text{XEX}^{*N,b,i,j}_{E_K}(M) = \text{XE}^{N,i,j}_{E_K}(M)$ if $b = 0$. We also define ΘCB-hc$_{\widetilde{E}}$ as a TBC mode for plain AE in Fig.3 for the security proof of OCB-hc. When $\widetilde{E}$ is XEX^*_E, ΘCB-hc$_{\widetilde{E}}$ is equivalent to OCB-hc$_E$. Let $\widetilde{\mathsf{P}}$ denote a TURP which has the same arguments as XEX^*. We define $\mathbf{Adv}^{\text{cpa-nr}}_{F,G}(\mathcal{A})$ (resp.

Algorithm: ΘCB-hc.$\mathcal{E}_{\tilde{E}}(N, M)$	**Algorithm:** ΘCB-hc.$\mathcal{D}_{\tilde{E}}(N, C, T)$
1. $M[1] \| \cdots \| M[m-1] \| M[m] \xleftarrow{n} M$	1. $C[1] \| \cdots \| C[m-1] \| C[m] \xleftarrow{n} C$
2. $\Sigma \leftarrow 0^{n/2}$	2. $\Sigma \leftarrow 0^{n/2}$
3. **for** $i \leftarrow 1$ **to** $m-1$ **do**	3. **for** $i \leftarrow 1$ **to** $m-1$ **do**
4. $\quad C[i] \leftarrow \tilde{E}^{N,1,i,0}(M[i])$	4. $\quad M[i] \leftarrow \tilde{D}^{N,1,i,0}(C[i])$
5. $\quad \Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(M[i])$	5. $\quad \Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(M[i])$
6. Pad $\leftarrow \tilde{E}^{N,1,m,0}(0^n)$	6. Pad $\leftarrow \tilde{E}^{N,1,m,0}(0^n)$
7. $C[m] \leftarrow M[m] \oplus \mathrm{msb}_{\lvert M[m]\rvert}(\mathrm{Pad})$	7. $M[m] \leftarrow C[m] \oplus \mathrm{msb}_{\lvert C[m]\rvert}(\mathrm{Pad})$
8. $\mathtt{Checksum} \leftarrow (\Sigma \| 0^{n/2}) \oplus \mathtt{ozp}(M[m])$	8. $\mathtt{Checksum} \leftarrow (\Sigma \| 0^{n/2}) \oplus \mathtt{ozp}(M[m])$
9. Tag $\leftarrow \tilde{E}^{N,0,m,1}(\mathtt{Checksum})$	9. Tag $\leftarrow \tilde{E}^{N,0,m,1}(\mathtt{Checksum})$
10. **if** $\lvert M[m]\rvert = n$ **then** $T \leftarrow \mathtt{msb}_{n/2}(\mathrm{Tag})$	10. **if** $\lvert M[m] = n$ **then** $T' \leftarrow \mathtt{msb}_{n/2}(\mathrm{Tag})$
11. **else** $T \leftarrow \mathtt{lsb}_{n/2}(\mathrm{Tag})$	11. **else** $T' \leftarrow \mathtt{lsb}_{n/2}(\mathrm{Tag})$
12. **return** $(C[1] \| \cdots \| C[m], T)$	12. **if** $T = T'$ **then return** $M[1] \| \cdots \| M[m]$
	13. **else return** $\perp$

Fig. 3. The algorithm of ΘCB-hc. $\tilde{E}$ is any TBC which has the same arguments as XEX*, and $\tilde{D}$ is the decryption of $\tilde{E}$.

$\mathbf{Adv}^{\text{cca-nr}}_{F,G}(\mathcal{A})$) as the probability that the chosen-plaintext attack (resp. chosen-ciphertext attack) adversary $\mathcal{A}$, who is nonce-respecting in encryption queries, can distinguish F from G. Then we obtain

$$\begin{aligned}
\mathbf{Adv}^{\text{priv}}_{\text{OCB-hc}_\mathsf{P}}(\mathcal{A}) &\le \mathbf{Adv}^{\text{cpa-nr}}_{\text{OCB-hc}_\mathsf{P},\Theta\text{CB-hc}_{\widetilde{\mathsf{P}}}}(\mathcal{A}) + \mathbf{Adv}^{\text{priv}}_{\Theta\text{CB-hc}_{\widetilde{\mathsf{P}}}}(\mathcal{A}) \\
&= \mathbf{Adv}^{\text{tprp}}_{\text{XEX}^*_{\widetilde{\mathsf{P}}}}(\mathcal{B}) + \mathbf{Adv}^{\text{priv}}_{\Theta\text{CB-hc}_{\widetilde{\mathsf{P}}}}(\mathcal{A}) \\
&\le \frac{4.5\sigma^2_{\text{priv}}}{2^n} + 0 \quad \text{and} \qquad (1) \\
\mathbf{Adv}^{\text{auth}}_{\text{OCB-hc}_\mathsf{P}}(\mathcal{A}^{\pm}) &\le \mathbf{Adv}^{\text{cca-nr}}_{\text{OCB-hc}_\mathsf{P},\Theta\text{CB-hc}_{\widetilde{\mathsf{P}}}}(\mathcal{A}^{\pm}) + \mathbf{Adv}^{\text{auth}}_{\Theta\text{CB-hc}_{\widetilde{\mathsf{P}}}}(\mathcal{A}^{\pm}) \\
&= \mathbf{Adv}^{\text{tsprp}}_{\text{XEX}^*_{\widetilde{\mathsf{P}}}}(\mathcal{B}^{\pm}) + \mathbf{Adv}^{\text{auth}}_{\Theta\text{CB-hc}_{\widetilde{\mathsf{P}}}}(\mathcal{A}^{\pm}) \\
&\le \frac{4.5\sigma^2_{\text{auth}}}{2^n} + \frac{4q_d}{2^{n/2}}, \qquad (2)
\end{aligned}$$

where $\mathcal{B}$ (resp. $\mathcal{B}^{\pm}$) is the adversary which can simulate $\mathcal{A}$ (resp. $\mathcal{A}^{\pm}$). The first terms of (1), (2) are derived from [38] and [33]. The derivations of the second terms of (1), (2) are described below.

Privacy. Every TURP invoked in the privacy game has the different tweak since the adversary is nonce-respecting. Thus, we have $\mathbf{Adv}^{\text{priv}}_{\Theta\text{CB-hc}_{\widetilde{\mathsf{P}}}}(\mathcal{A}) = 0$.

Authenticity.

Lemma 1. *The authenticity advantage of* ΘCB-hc$_{\widetilde{\mathsf{P}}}$ *is*

$$\mathbf{Adv}^{\text{auth}}_{\Theta\text{CB-hc}_{\widetilde{\mathsf{P}}}}(\mathcal{A}^{\pm}) \le \frac{4q_d}{2^{n/2}},$$

where q_d *denotes the number of verification (decryption) queries.*

Proof. We start with the case $q_d = 1$. Without loss of generality, the adversary performs the decryption query after all encryption queries. Suppose that she obtains the transcript $z = \{(N_1, M_1, C_1, T_1), \ldots, (N_q, M_q, C_q, T_q)\}$ in encryption query, and she queries (N', C', T') in decryption query. Let Z be the set of all transcripts, and T^* be the valid tag for (N', C'). We define the function ifPad : $\{0,1\}^* \to \{0,1\}$ as follows.

$$\mathrm{ifPad}(M) = \begin{cases} 0 & \text{if } |M| = 0 \bmod n; \\ 1 & \text{otherwise.} \end{cases}$$

Then we obtain the following equations.

$$\begin{aligned} \mathbf{Adv}^{\mathrm{auth}}_{\Theta\mathrm{CB\text{-}hc}_{\widetilde{\mathrm{P}}}}(\mathcal{A}^{\pm}) &= \Pr[T' = T^*] \\ &= \sum_z \Pr[T' = T^*, Z = z] \\ &= \sum_z \Pr[T' = T^* \mid Z = z] \Pr[Z = z]. \end{aligned}$$

We define $\mathrm{FP}_z := \Pr[T' = T^* \mid Z = z]$ and evaluate $\max_z \mathrm{FP}_z$ as below.

1. Let $N' \neq N_i$ for $1 \leq \forall i \leq q$. Since the TURP which returns valid T^* takes a new tweak, the adversary has no information about T^*. Thus $\mathrm{FP}_z \leq 1/2^{n/2}$ holds.
2. Let $N' = N_\alpha$, $\alpha \in \{1, 2, \ldots, q\}$, $C' \neq C_\alpha$. We divide the cases with the value of $|C'|$ as follows.
   (a) Let $|C'|_n \neq |C_\alpha|_n$. The tweak of the TURP which outputs T^* is different from that of TURPs which are invoked in encryption query. Thus $\mathrm{FP}_z \leq 1/2^{n/2}$ holds.
   (b) Let $|C'|_n = |C_\alpha|_n$ and $\mathrm{ifPad}(C') \neq \mathrm{ifPad}(C_\alpha)$. Suppose that $\mathtt{Checksum}^*$ and M^* are the valid checksum and message for (N', C'), respectively, and $\mathtt{Checksum}_\alpha$ is the value of the checksum for $(N_\alpha, M_\alpha, C_\alpha, T_\alpha)$. The adversary can make $\mathtt{Checksum}^*$ equal to $\mathtt{Checksum}_\alpha$ by using padding. When $\mathtt{Checksum}^* \neq \mathtt{Checksum}_\alpha$, $\mathrm{FP}_z \leq 2^{n/2}/(2^n - 1)$ holds. When $\mathtt{Checksum}^* = \mathtt{Checksum}_\alpha$, $\mathrm{FP}_z \leq 1/(2^{n/2})$ holds since $\mathrm{ifPad}(C') \neq \mathrm{ifPad}(C_\alpha)$ and the adversary obtains no information about T^* from T_α.
   (c) Let $|C'|_n = |C_\alpha|_n$ and $\mathrm{ifPad}(C') = \mathrm{ifPad}(C_\alpha)$. Suppose $|C'|_n = |C_\alpha|_n = m$. We consider the following cases.

   Case e_1: When $C' \neq C_\alpha$, $\mathtt{Checksum}^* = \mathtt{Checksum}_\alpha$ holds.
   Case e_2: When $C' \neq C_\alpha$, $T' = T^*$ holds.

   We first evaluate $\Pr[e_1 \mid Z = z] = \Pr[\mathtt{Checksum}^* = \mathtt{Checksum}_\alpha \mid Z = z]$. When $C'[m] \neq C_\alpha[m]$ and $C'[i] = C_\alpha[i]$ for $\forall i \in \{1, \ldots, m-1\}$, we obtain $\Pr[e_1 \mid Z = z] = 0$ since $\mathtt{ozp}(M^*[m]) \neq \mathtt{ozp}(M_\alpha[m])$ holds. Then suppose $C'[u] \neq C_\alpha[u]$ for $\exists u\{1, \ldots, m-1\}$. We obtain following

evaluation.

$$\begin{aligned}
&\Pr[e_1 \mid Z = z] \\
&= \Pr\left[\left(\mathtt{msb}_{n/2}(M^*[u]) \,\|\, 0^{n/2}\right) \oplus \left(\mathtt{msb}_{n/2}(M_\alpha[u]) \,\|\, 0^{n/2}\right) = \delta \mid Z = z\right] \\
&\leq \frac{2^{n/2}}{2^n - 1},
\end{aligned}$$

where $\delta = \left(\mathtt{msb}_{n/2}(M^*[u]) \,\|\, 0^{n/2}\right) \oplus \left(\mathtt{msb}_{n/2}(M_\alpha[u]) \,\|\, 0^{n/2}\right) \oplus \mathtt{Checksum}^* \oplus \mathtt{Checksum}_\alpha$. Thus, $\Pr[e_1 \mid Z = z] \leq 2/2^{n/2}$ is obtained. Then we evaluate $\Pr[e_2|\bar{e_1}, Z = z]$. In this case, the TURP outputting T^* and the TURP outputting T_α take the same tweak, and $\mathrm{ifPad}(C') = \mathrm{ifPad}(C_\alpha)$ holds. However, $\mathtt{Checksum}^* \neq \mathtt{Checksum}_\alpha$ holds, and we obtain $\Pr[e_2|\bar{e_1}, Z = z] \leq 2^{n/2}/(2^n - 1)$.

From above, we obtain the following evaluation.

$$\begin{aligned}
\mathrm{FP}_z &= \Pr[e_2|Z = z] \\
&\leq \Pr[e_2 \cap \bar{e_1}|Z = z] + \Pr[e_1|Z = z] \\
&\leq \Pr[e_2|\bar{e_1}, Z = z] + \Pr[e_1|Z = z] \\
&\leq \frac{2^{n/2}}{2^n - 1} + \frac{2^{n/2}}{2^n - 1} \leq \frac{4}{2^{n/2}}.
\end{aligned}$$

From the evaluations of the above cases, we obtain

$$\mathbf{Adv}^{\mathrm{auth}}_{\Theta\mathrm{CB\text{-}hc}}(\mathcal{A}^{\pm}) \leq \sum_z \max_z \mathrm{FP}_z \cdot \Pr[Z = z] \leq \frac{4}{2^{n/2}}.$$

For the case $q_d > 1$, we apply the generic conversion from $q_d = 1$ to $q_d > 1$ as shown by [10], which multiplies q_d to the above. This concludes the proof.

4.4 OTR-hc

We propose another plain AE scheme denoted by OTR-hc which is obtained by applying half-checksum method to OTR. As well as OCB-hc, we first propose OTR-hc as a plain AE with $n/2$-bit tag. The extension to AEAD with possibly longer tag is possible with a method applied to OCB-hc (See Section 5).

Specification. We show OTR-hc in Fig. 4 and Fig. 5. Let E_K be an n-bit blockcipher. We define the encryption function of OTR-hc$_{E_K}$ as OTR-hc.$\mathcal{E}_{E_K} : (N, M) \mapsto (C, T)$, where $(N, M) \in \{0,1\}^n \times \{0,1\}^*$ and $(C, T) \in \{0,1\}^* \times \{0,1\}^{n/2}$. We also define the decryption function as OTR-hc.$\mathcal{D}_{E_K} : (N, C, T) \mapsto M$ or $\perp$, where $(N, C, T) \in \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^{n/2}$ and $M \in \{0,1\}^*$. OTR-hc

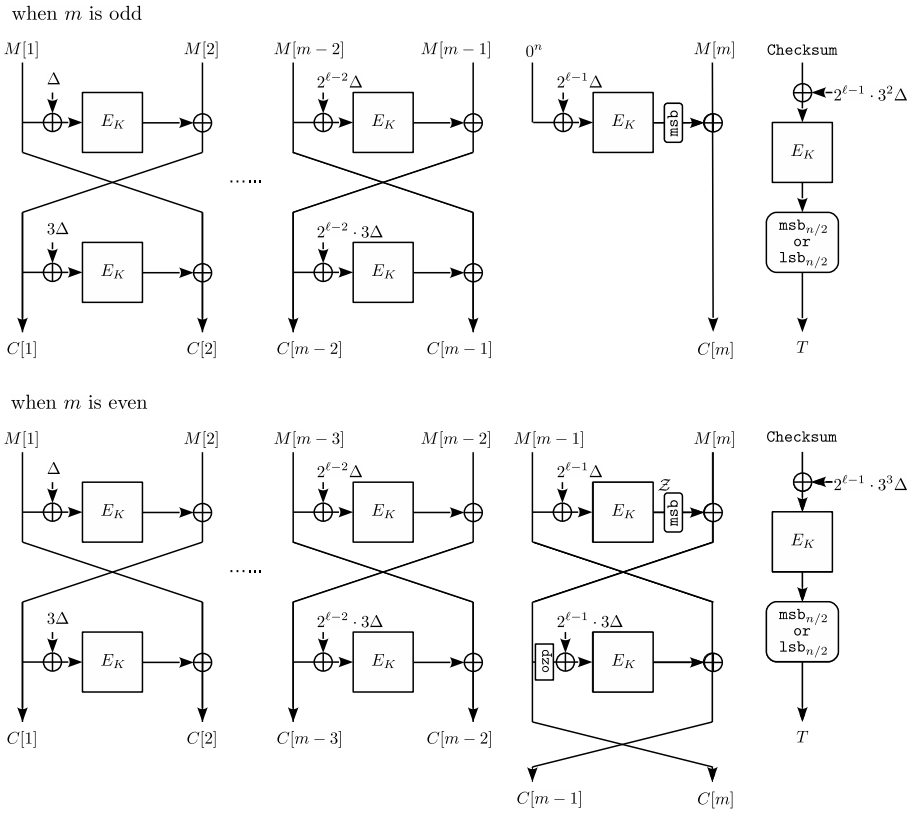

Fig. 4. The encryption of OTR-hc$_{E_K}$, where E_K is any n-bit blockcipher and $\Delta = E_K(N)$. When the number of input blocks m is an odd number, $\mathtt{Checksum} = (\Sigma \,\|\, 0^{n/2}) \oplus \mathtt{ozp}(M[m])$, where $\Sigma = \bigoplus_{i=1}^{(m-1)/2} \mathtt{msb}_{n/2}(M[2i])$. Otherwise, $\mathtt{Checksum} = \left(\bigoplus_{i=1}^{(m-2)/2} \mathtt{msb}_{n/2}(M[2i]) \oplus \mathtt{msb}_{n/2}(\mathcal{Z})\right) \,\|\, 0^{n/2}$.

encrypts message with 2-round Feistel based on XE. An input to $2n$-bit Feistel permutation is called a chunk. The checksum is computed by XORing the most significant $n/2$ bits of the right halves of the chunk (*i.e.* the even-numbered message blocks) except the last chunk. When the number of message blocks, m, is odd, we take an XOR of $M[m]$ and the padded checksum. When m is even, we will take an XOR of $\mathtt{msb}_{n/2}(\mathcal{Z})$ and the checksum in the last chunk so that any small difference in $C[m-1]$ or $C[m]$ (typically between the encryption and decryption queries sharing the nonce) will yield the n-bit difference of $\mathcal{Z}$.

State size. When m is odd, OTR-hc has $3.5n$-bit state size following the procedure described above because the last chunk has only one block. When m is even, it also has $3.5n$-bit state size since the checksum can be computed in

Algorithm: OTR-hc.$\mathcal{E}_{E_K}(N, M)$

$M[1] \| \cdots \| M[m-1] \| M[m] \xleftarrow{n} M$
$\Delta \leftarrow E_K(N)$, $\ell \leftarrow \lceil \frac{m}{2} \rceil$, $\Sigma \leftarrow 0^{n/2}$
for $i \leftarrow 1$ **to** $\ell - 1$ **do**
  $C[2i-1] \leftarrow E_K(M[2i-1] \oplus \Delta) \oplus M[2i]$
  $C[2i] \leftarrow E_K(C[2i-1] \oplus 3\Delta) \oplus M[2i-1]$
  $\Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(M[2i])$
  $\Delta \leftarrow 2\Delta$
if m **is odd**
  $C[m] \leftarrow \mathtt{msb}_{|M[m]|}(E_K(0^n \oplus \Delta)) \oplus M[m]$
  $\mathtt{Checksum} \leftarrow \Sigma \| 0^{n/2} \oplus \mathtt{ozp}(M[m])$
else
  $\mathcal{Z} \leftarrow E_K(M[m-1] \oplus \Delta)$, $\Delta \leftarrow 3\Delta$
  $\Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(\mathcal{Z})$
  $C[m] \leftarrow \mathtt{msb}_{|M[m]|}(\mathcal{Z}) \oplus M[m]$
  $C[m-1] \leftarrow E_K(\mathtt{ozp}(C[m]) \oplus \Delta) \oplus M[m-1]$
  $\mathtt{Checksum} \leftarrow \Sigma \| 0^{n/2}$
Tag $\leftarrow E_K(\mathtt{Checksum} \oplus 3^2\Delta)$
if $|M[m]| = n$ **then** $T \leftarrow \mathtt{msb}_{n/2}(\text{Tag})$
else $T \leftarrow \mathtt{lsb}_{n/2}(\text{Tag})$
return $(C[1] \| \cdots \| C[m], T)$

Algorithm: OTR-hc.$\mathcal{D}_{E_K}(N, C, T)$

$C[1] \| \cdots \| C[m-1] \| C[m] \xleftarrow{n} C$
$\Delta \leftarrow E_K(N)$, $\ell \leftarrow \lceil \frac{m}{2} \rceil$, $\Sigma \leftarrow 0^{n/2}$
for $i \leftarrow 1$ **to** $\ell - 1$ **do**
  $M[2i-1] \leftarrow E_K(C[2i-1] \oplus 3\Delta) \oplus C[2i]$
  $M[2i] \leftarrow E_K(M[2i-1] \oplus \Delta) \oplus C[2i-1]$
  $\Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(M[2i])$
  $\Delta \leftarrow 2\Delta$
if m **is odd**
  $M[m] \leftarrow \mathtt{msb}_{|C[m]|}(E_K(0^n \oplus \Delta)) \oplus C[m]$
  $\mathtt{Checksum} \leftarrow \Sigma \| 0^{n/2} \oplus \mathtt{ozp}(M[m])$
else
  $M[m-1] \leftarrow E_K(\mathtt{ozp}(C[m]) \oplus 3\Delta) \oplus C[m-1]$
  $\mathcal{Z} \leftarrow E_K(M[m-1] \oplus \Delta)$, $\Delta \leftarrow 3\Delta$
  $\Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(\mathcal{Z})$
  $M[m] \leftarrow \mathtt{msb}_{|C[m]|}(\mathcal{Z}) \oplus C[m]$
  $\mathtt{Checksum} \leftarrow \Sigma \| 0^{n/2}$
Tag $\leftarrow E_K(\mathtt{Checksum} \oplus 3^2\Delta)$
if $|M[m]| = n$ **then** $T' \leftarrow \mathtt{msb}_{n/2}(\text{Tag})$
else $T' \leftarrow \mathtt{lsb}_{n/2}(\text{Tag})$
if $T = T'$ **then return** $M[1] \| \cdots \| M[m]$
else return $\perp$

Fig. 5. The algorithm of OTR-hc. E_K is any n-bit blockcipher.

$n/2$ bits. Thus, the state size of OTR-hc is $3.5n$ bits. Unlike OCB-hc, we do not have to derive an alternative procedure for the last chunk.

4.5 Security of OTR-hc

We here show the security bounds of OTR-hc. As in the security proof of OCB-hc, we assume the underlying blockcipher is an n-bit URP, P and omit the case when the underlying blockcipher is a PRP.

Theorem 2. *The security bounds of* OTR-hc$_{\mathsf{P}}$ *are evaluated as follows:*

$$\mathbf{Adv}^{\mathrm{priv}}_{\text{OTR-hc}_{\mathsf{P}}}(\mathcal{A}) \leq \frac{5\sigma^2_{\mathrm{priv}}}{2^n}, \ \mathbf{Adv}^{\mathrm{auth}}_{\text{OTR-hc}_{\mathsf{P}}}(\mathcal{A}^{\pm}) \leq \frac{5\sigma^2_{\mathrm{auth}}}{2^n} + \frac{2.5q_d}{2^{n/2}},$$

where $\mathcal{A}$, $\mathcal{A}^{\pm}$ are the adversaries against OTR-hc *and σ_{priv}, σ_{auth}, q_d are the parameters for $\mathcal{A}$, $\mathcal{A}^{\pm}$. The parameter σ_{priv} (resp. σ_{auth}) is the number of accesses to P in privacy game (resp. authenticity game) and q_d is the number of queries to the decryption oracle in authenticity game.*

Proof. To evaluate the security bound of OTR-hc, we define the TBC mode for plain AE, which is denoted by ΘTR-hc$_{\widetilde{E}}$ in Fig.6. When $\widetilde{E}$ is XE$_E$, ΘTR-hc$_{\widetilde{E}}$ is equivalent to OTR-hc$_E$. Let $\widetilde{\mathsf{R}}$ denote a TURF which has the same arguments as XE. For privacy-adversary $\mathcal{A}$ and authenticity-adversary $\mathcal{A}^{\pm}$, we obtain following

Algorithm: ΘTR-hc.$\mathcal{E}_{\widetilde{E}}(N, M)$

$M[1] \| \cdots \| M[m-1] \| M[m] \xleftarrow{n} M$
$\ell \leftarrow \lceil \frac{m}{2} \rceil$, $\Sigma \leftarrow 0^{n/2}$
for $i \leftarrow 1$ **to** $\ell - 1$ **do**
  $C[2i-1] \leftarrow \tilde{E}^{N,i-1,0}(M[2i-1]) \oplus M[2i]$
  $C[2i] \leftarrow \tilde{E}^{N,i-1,1}(C[2i-1]) \oplus M[2i-1]$
  $\Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(M[2i])$
if m **is odd**
  $C[m] \leftarrow \mathtt{msb}_{|M[m]|}(\tilde{E}^{N,\ell-1,0}(0^n)) \oplus M[m]$
  $\mathtt{Checksum} \leftarrow (\Sigma \| 0^{n/2}) \oplus \mathtt{ozp}(M[m])$
else
  $\mathcal{Z} \leftarrow \tilde{E}^{N,\ell-1,0}(M[m-1])$
  $\Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(\mathcal{Z})$
  $C[m] \leftarrow \mathtt{msb}_{|M[m]|}(\mathcal{Z}) \oplus M[m]$
  $C[m-1] \leftarrow \tilde{E}^{N,\ell-1,1}(\mathtt{ozp}(C[m])) \oplus M[m-1]$
  $\mathtt{Checksum} \leftarrow \Sigma \| 0^{n/2}$
if m **is odd then** Tag $\leftarrow \tilde{E}^{N,\ell-1,2}(\mathtt{Checksum})$
else Tag $\leftarrow \tilde{E}^{N,\ell-1,3}(\mathtt{Checksum})$
if $|M[m]| = n$ **then** $T \leftarrow \mathtt{msb}_{n/2}(\text{Tag})$
else $T \leftarrow \mathtt{lsb}_{n/2}(\text{Tag})$
return $(C[1] \| \cdots \| C[m]) \| T$

Algorithm: ΘTR-hc.$\mathcal{D}_{\widetilde{E}}(N, C, T)$

$C[1] \| \cdots \| C[m-1] \| C[m] \xleftarrow{n} C$
$\ell \leftarrow \lceil \frac{m}{2} \rceil$, $\Sigma \leftarrow 0^{n/2}$
for $i \leftarrow 1$ **to** $\ell - 1$ **do**
  $M[2i-1] \leftarrow \tilde{E}^{N,i-1,1}(C[2i-1]) \oplus C[2i]$
  $M[2i] \leftarrow \tilde{E}^{N,i-1,0}(M[2i-1]) \oplus C[2i-1]$
  $\Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(M[2i])$
if m **is odd**
  $M[m] \leftarrow \mathtt{msb}_{|C[m]|}(\tilde{E}^{N,\ell-1,0}(0^n)) \oplus C[m]$
  $\mathtt{Checksum} \leftarrow (\Sigma \| 0^{n/2}) \oplus \mathtt{ozp}(M[m])$
else
  $M[m-1] \leftarrow \tilde{E}^{N,\ell-1,1}(\mathtt{ozp}(C[m])) \oplus C[m-1]$
  $\mathcal{Z} \leftarrow \tilde{E}^{N,\ell-1,0}(M[m-1])$
  $M[m] \leftarrow \mathtt{msb}_{|C[m]|}(\mathcal{Z}) \oplus C[m]$
  $\Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(\mathcal{Z})$
  $\mathtt{Checksum} \leftarrow \Sigma \| 0^{n/2}$
if m **is odd then** Tag $\leftarrow \tilde{E}^{N,\ell-1,2}(\mathtt{Checksum})$
else Tag $\leftarrow \tilde{E}^{N,\ell-1,3}(\mathtt{Checksum})$
if $|M[m]| = n$ **then** $T' \leftarrow \mathtt{msb}_{n/2}(\text{Tag})$
else $T' \leftarrow \mathtt{lsb}_{n/2}(\text{Tag})$
if $T = T'$ **then return** $M[1] \| \cdots \| M[m]$
else return $\perp$

Fig. 6. The algorithm of ΘTR-hc. $\widetilde{E}$ is any TBC which has the same arguments as XE.

security bounds of OTR-hc$_{\mathsf{P}}$.

$$\begin{aligned}
\mathbf{Adv}^{\text{priv}}_{\text{OTR-hc}_{\mathsf{P}}}(\mathcal{A}) &\le \mathbf{Adv}^{\text{cpa-nr}}_{\text{OTR-hc}_{\mathsf{P}},\Theta\text{TR-hc}_{\widetilde{\mathsf{R}}}}(\mathcal{A}) + \mathbf{Adv}^{\text{priv}}_{\Theta\text{TR-hc}_{\widetilde{\mathsf{R}}}}(\mathcal{A}) \\
&= \mathbf{Adv}^{\text{cpa-nr}}_{\text{XE}_{\mathsf{P}},\widetilde{\mathsf{R}}}(\mathcal{B}) + \mathbf{Adv}^{\text{priv}}_{\Theta\text{TR-hc}_{\widetilde{\mathsf{R}}}}(\mathcal{A}) \\
&\le \frac{5\sigma^2_{\text{priv}}}{2^n} + 0 \quad \text{and} \qquad (3) \\
\mathbf{Adv}^{\text{auth}}_{\text{OTR-hc}_{\mathsf{P}}}(\mathcal{A}^{\pm}) &\le \mathbf{Adv}^{\text{cca-nr}}_{\text{OTR-hc}_{\mathsf{P}},\Theta\text{TR-hc}_{\widetilde{\mathsf{R}}}}(\mathcal{A}^{\pm}) + \mathbf{Adv}^{\text{auth}}_{\Theta\text{TR-hc}_{\widetilde{\mathsf{R}}}}(\mathcal{A}^{\pm}) \\
&= \mathbf{Adv}^{\text{cpa-nr}}_{\text{XE}_{\mathsf{P}},\widetilde{\mathsf{R}}}(\mathcal{B}^{\pm}) + \mathbf{Adv}^{\text{auth}}_{\Theta\text{TR-hc}_{\widetilde{\mathsf{R}}}}(\mathcal{A}^{\pm}) \\
&\le \frac{5\sigma^2_{\text{auth}}}{2^n} + \frac{2.5q_d}{2^{n/2}}, \qquad (4)
\end{aligned}$$

where $\mathcal{B}$ (resp. $\mathcal{B}^{\pm}$) is the adversary which can simulate $\mathcal{A}$ (resp. $\mathcal{A}^{\pm}$). The first terms of (3), (4) are derived from [31]. The second terms of (3), (4) are described below.

Privacy. As in the case of ΘCB-hc, every TURF invoked in the privacy game has a different tweak because the adversary is nonce-respecting. Therefore, we have $\mathbf{Adv}^{\text{priv}}_{\Theta\text{TR-hc}_{\widetilde{\mathsf{R}}}}(\mathcal{A}) = 0$.

Authenticity.

Lemma 2. *The authenticity advantage of* ΘTR-hc$_{\widetilde{\mathsf{R}}}$ *is*

$$\mathbf{Adv}^{\mathrm{auth}}_{\Theta\text{TR-hc}_{\widetilde{\mathsf{R}}}}(\mathcal{A}^{\pm}) \leq \frac{2.5q_d}{2^{n/2}},$$

where q_d *denotes the number of decryption queries.*

Proof. We start with the case $q_d = 1$. Without loss of generality, we assume that the adversary performs decryption query after all encryption queries. As in the security proof of OCB-hc, suppose that she obtains the transcript $z = \{(N_1, M_1, C_1, T_1), \ldots, (N_q, M_q, C_q, T_q)\}$ in encryption query, and then she queries (N', C', T') in decryption query. Let Z be the set of all transcripts, and T^* be the valid tag for (N', C'). We define $\mathrm{FP}_z := \Pr[T' = T^* \mid Z = z]$ and evaluate $\max_z \mathrm{FP}_z$ as below.

1. Let $N' \neq N_i$, $1 \leq \forall i \leq q$. Since the TURF which returns valid T^* takes a new tweak, the adversary has no information about T^*. Thus $\mathrm{FP}_z \leq 1/2^{n/2}$ holds.
2. Let $N' = N_\alpha$, $\alpha \in \{1, 2, \ldots, q\}$, $C' \neq C_\alpha$. We divide the cases with the value of $|C'|$ as follows.
   (a) Let $|C'|_{2n} \neq |C_\alpha|_{2n}$. The tweak of TURF which outputs T^* is different from that of TURFs which are invoked in encryption query. Thus $\mathrm{FP}_z \leq 1/2^{n/2}$ holds.
   (b) Let $|C'|_{2n} = |C_\alpha|_{2n}$ and $|C'|_n \neq |C_\alpha|_n$. As above, the tweak of TURF which outputs T^* is different from that of TURFs which are invoked in encryption query. Thus $\mathrm{FP}_z \leq 1/2^{n/2}$ holds.
   (c) Let $|C'|_n = |C_\alpha|_n$ and $\mathrm{ifPad}(C') \neq \mathrm{ifPad}(C_\alpha)$. Let $|C'|_n = |C_\alpha|_n = m$. We first consider the case that m is odd. Suppose that $\mathtt{Checksum}^*$ and M^* are the valid checksum and message for (N', C'), respectively, and $\mathtt{Checksum}_\alpha$ is the value of the checksum for $(N_\alpha, M_\alpha, C_\alpha, T_\alpha)$. The adversary can make $\mathtt{Checksum}^*$ equal to $\mathtt{Checksum}_\alpha$ by using padding. However, we obtain $\mathrm{FP}_z \leq 1/2^{n/2}$ no matter if $\mathtt{Checksum}^* \neq \mathtt{Checksum}_\alpha$ holds or not since $\mathrm{ifPad}(C') \neq \mathrm{ifPad}(C_\alpha)$ and the adversary obtains no information about T^* from T_α. Regarding to the case that m is even, we can discuss in the same way as above.
   (d) Let $|C'|_n = |C_\alpha|_n$ and $\mathrm{ifPad}(C') = \mathrm{ifPad}(C_\alpha)$. Suppose $|C'|_n = |C_\alpha|_n = m$ and $CC[1] \,\|\, CC[2] \,\|\, \cdots \,\|\, CC[\ell] \xleftarrow{2n} C$. We consider the following cases.
   Case e_1**:** When $CC'[i] \neq CC_\alpha[i]$ for $\exists i \in \{1, \ldots, \ell\}$, $M^*[2i-1] = M_\alpha[2i-1]$ holds.
   Case e_2**:** When $C' \neq C_\alpha$, $\mathtt{Checksum}^* = \mathtt{Checksum}_\alpha$ holds.
   Case e_3**:** When $C' \neq C_\alpha$, $T' = T^*$ holds.
   We first evaluate $\Pr[e_1 \mid Z = z] = \Pr[M^*[2i-1] = M_\alpha[2i-1] \mid Z = z]$. Let $i \in \{1, \ldots, \ell - 1\}$. When $C'[2i-1] = C_\alpha[2i-1]$, $C'[2i] \neq C_\alpha[2i]$ has to hold. Thus we obtain $\Pr[e_1 \mid Z = z] = 0$ since $\widetilde{\mathsf{R}}^{N,i-1,1}(C'[2i-1]) \oplus C'[2i] \neq \widetilde{\mathsf{R}}^{N,i-1,1}(C_\alpha[2i-1]) \oplus C_\alpha[2i]$ always holds. Then let $C'[2i-1] \neq$

$C_\alpha[2i-1]$. $\Pr[e_1 \mid Z = z] \le 1/2^n$ holds because $\widetilde{\mathsf{R}}^{N,i-1,1}(C'[2i-1])$ is unpredictable for the adversary. When $i = \ell$ and m is even, $\Pr[e_1 \mid Z = z] \le 1/2^n$ holds from the almost same discussion as above. When $i = \ell$ and m is odd, $\Pr[e_1 \mid Z = z] = 0$ holds.
Secondly, we evaluate $\Pr[e_2 \mid \bar{e_1}, Z = z]$. Let m is odd. When $C'[m] \neq C_\alpha[m]$ and $CC'[i] = CC_\alpha[i]$ for $\forall i \in \{1, \ldots, \ell - 1\}$, we obtain $\Pr[e_2 \mid \bar{e_1}, Z = z] = 0$ since $\mathtt{ozp}(M^*[m]) \neq \mathtt{ozp}(M_\alpha[m])$ holds. Then, suppose $CC'[u] \neq CC_\alpha[u]$ for $\exists u \in \{1, \ldots, \ell - 1\}$. We obtain the following evaluation.

$$
\begin{aligned}
&\Pr[e_2 \mid \bar{e_1}, Z = z] \\
&= \Pr[\mathtt{msb}_{n/2}(M^*[2u]) \,\|\, 0^{n/2} \oplus \mathtt{msb}_{n/2}(M_\alpha[2u]) \,\|\, 0^{n/2} = \delta \;\mid \bar{e_1}, Z = z],
\end{aligned}
$$

where $\delta = \mathtt{msb}_{n/2}(M^*[2u]) \,\|\, 0^{n/2} \oplus \mathtt{msb}_{n/2}(M_\alpha[2u]) \,\|\, 0^{n/2} \oplus \mathtt{Checksum}^* \oplus \mathtt{Checksum}_\alpha$,

$$
\begin{aligned}
&= \Pr[\mathtt{msb}_{n/2}(\widetilde{\mathsf{R}}^{N,u-1,0}(M^*[2u-1]) \oplus C'[2u-1]) \,\|\, 0^{n/2} \\
&\quad \oplus \mathtt{msb}_{n/2}(\widetilde{\mathsf{R}}^{N,u-1,0}(M_\alpha[2u-1]) \oplus C_\alpha[2u-1]) \,\|\, 0^{n/2} = \delta \mid \bar{e_1}, Z = z] \\
&\le 1/2^{n/2}.
\end{aligned}
$$

The last line is derived since $\bar{e_1}$ and $\widetilde{\mathsf{R}}^{N,u-1,0}(M^*[2u-1])$ is unpredictable. Thus, we obtain $\Pr[e_2 \mid \bar{e_1}, Z = z] \le 1/2^{n/2}$ when m is odd. When m is even, $\Pr[e_2 \mid \bar{e_1}, Z = z] \le 1/2^{n/2}$ also holds from the almost same discussion as above. Then we evaluate $\Pr[e_3 \mid \bar{e_2}, \bar{e_1}, Z = z]$. In this case, the TURF outputting T^* and the TURF outputting T_α take the same tweak, and $\mathrm{ifPad}(C') = \mathrm{ifPad}(C_\alpha)$ holds. However $\mathtt{Checksum}^* \neq \mathtt{Checksum}_\alpha$ holds, and we obtain $\Pr[e_3 \mid \bar{e_2}, \bar{e_1}, Z = z] \le 1/2^{n/2}$.

From above, we obtain the following evaluation.

$$
\begin{aligned}
\mathrm{FP}_z &= \Pr[e_3 \mid Z = z] \\
&\le \Pr[e_3 \cap (\overline{e_1 \cup e_2}) \mid Z = z] + \Pr[e_2 \cap \bar{e_1} \mid Z = z] + \Pr[e_1 \mid Z = z] \\
&\le \Pr[e_3 \mid \bar{e_2}, \bar{e_1}, Z = z] + \Pr[e_2 \mid \bar{e_1}, Z = z] + \Pr[e_1 \mid Z = z] \\
&\le \frac{1}{2^{n/2}} + \frac{1}{2^{n/2}} + \frac{1}{2^n} \le \frac{2.5}{2^{n/2}}
\end{aligned}
$$

From the evaluations of above cases, we obtain

$$
\mathbf{Adv}^{\mathrm{auth}}_{\Theta\mathrm{TR\text{-}hc}}(\mathcal{A}^{\pm}) \le \sum_z \max_z \mathrm{FP}_z \cdot \Pr[Z = z] \le \frac{2.5}{2^{n/2}}.
$$

For the case $q_d > 1$, we use [10] again. This completes the proof.

5 Extensions

In this section, we show extensions of our proposals. First, we show how to extend the tag length of OCB-hc to up to n bits. Second, we propose an extension of OCB-hc to AEAD, denoted by OCB-hc-AD, which is the mode of operation for AEAD with $2.5n$-bit state size. OCB-hc-AD is a combination of OCB-hc and a variant of Phash [27] with half-checksum method. OTR-hc can be extended to have arbitrary tag length up to n bits and AEAD in the same manner as OCB-hc, which we omit here.

5.1 Arbitrary tag length

When tag length τ is less than $n/2$ bits, we can change line 12 and 13 of OCB-hc.$\mathcal{E}$ in Fig. 2 as follows.

$$\begin{aligned}&\text{line 12}: \ \textbf{if}\ |M[m]| = n\ \textbf{then}\ T \leftarrow \mathtt{msb}_\tau(\mathrm{Tag}),\\ &\text{line 13}: \ \textbf{else}\ T \leftarrow \mathtt{lsb}_\tau(\mathrm{Tag}).\end{aligned}$$

For decryption, we can change OCB-hc.$\mathcal{D}$ accordingly. When $\tau > n/2$, we can change line 8 and 12–14 of OCB-hc.$\mathcal{E}$ in Fig.2 as follows.

$$\begin{aligned}&\text{line 8}: \ \textbf{if}\ |M[m]| = n\ \textbf{then}\ \Delta \leftarrow 3\Delta, \textbf{else}\ \Delta \leftarrow 3^2\Delta,\\ &\text{line 12–14}: \ \textbf{return}(C[1] \,\|\, \cdots \,\|\, C[m], \mathtt{msb}_\tau(\mathrm{Tag})).\end{aligned}$$

For decryption, we can change OCB-hc.$\mathcal{D}$ accordingly. Thus, we have to use the different masks in the encryption of the checksum, depending on whether the message is full n bits or partial, which is the same as the original OCB and OTR.

5.2 OCB-hc with AD

Our extension of OCB-hc to AEAD, denoted by OCB-hc-AD, is shown in Fig.7. OCB-hc-AD consists of the plain-AE core OCB-hc$'$ and the authentication core Phash-hc (Fig. 8 in Appendix A). The way of combination is similar to ΘCB3† proposed by Naito [34]. In OCB-hc-AD, Phash-hc processes AD and then OCB-hc$'$ processes a message using the output of Phash-hc as the initial value of the checksum. Note that the initial value of the checksum was $0^{n/2}$ in the case of OCB-hc. This way of combination is suitable when AD is processed first. If the message is processed before AD, one can combine OCB-hc$'$ and Phash-hc by XORing the tag of plain-AE OCB-hc$'$ and the output of Phash-hc. This combination is similar to OCB3 or AEM [27,38].

Specification. We show OCB-hc-AD in Fig.7. For simplicity, the tag is $n/2$ bits. Let E_K be an n-bit blockcipher. We define the encryption function of OCB-hc-AD$_{E_K}$ as OCB-hc-AD.$\mathcal{E}_{E_K} : (N, A, M) \mapsto (C, T)$, where $(N, A, M) \in \{0,1\}^{\leq n-1} \times \{0,1\}^* \times \{0,1\}^*$ and $(C, T) \in \{0,1\}^* \times \{0,1\}^{n/2}$. We also define

the decryption function as OCB-hc-AD.$\mathcal{D}_{E_K} : (N, A, C, T) \mapsto M$ or $\perp$, where $(N, A, C, T) \in \{0,1\}^{\leq n-1} \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^{n/2}$ and $M \in \{0,1\}^*$. OCB-hc$'$ is the same algorithm as OCB-hc except the length of nonce and the initial value of the checksum. We restrict the length of nonce to less than n bits because Phash-hc always uses 0^n as a nonce and so OCB-hc$'$ cannot use 0^n as a nonce. The initial value of the checksum of OCB-hc$'$ is an output of Phash-hc. Phash-hc computes the sum of the most significant $n/2$ bits of encrypted massage by XE.

State size. OCB-hc$'$ has $2.5n$-bit state size as OCB-hc$'$ and OCB-hc are almost the same. Phash-hc also has $2.5n$-bit state size, which includes n-bit memory for message block and mask, and $0.5n$-bit memory for sum of encrypted message. Therefore, the state size of OCB-hc-AD is $2.5n$ bits.

5.3 Security of OCB-hc-AD

We here show the security bounds of OCB-hc-AD. For security analysis of OCB-hc-AD, we define ΘCB-hc-AD as a TBC mode for AEAD in Fig.7. We also define ΘCB-hc$'$ and ℙhash-hc as TBC versions of OCB-hc$'$ and Phash-hc, respectively in Fig.7. When $\widetilde{E}$ is instantiated by XE_E, ℙhash-hc$_{\widetilde{E}}$ is equivalent to Phash-hc$_E$. In this subsection, we first show the security of ℙhash-hc. Then we evaluate the security bounds of OCB-hc-AD using hybrid argument.

Lemma 3. *Let $\forall A, A' \in \{0,1\}^*$ and $A \neq A'$. Suppose the underlying TBC of* ℙhash-hc *is a TURP denoted by $\widetilde{\mathsf{P}}$, which has the same arguments as* XE. ℙhash-hc$_{\widetilde{\mathsf{P}}}$ *has a following property.*

$$\max_{\forall \delta \in \{0,1\}^{n/2}} \Pr\left[\mathbb{P}\text{hash-hc}_{\widetilde{\mathsf{P}}}(A) \oplus \mathbb{P}\text{hash-hc}_{\widetilde{\mathsf{P}}}(A') = \delta\right] \leq \frac{2}{2^{n/2}}.$$

The proof is described in Appendix A.

Then we show the security bounds of OCB-hc-AD. As in the security proofs of OCB-hc and OTR-hc, we assume the underlying blockcipher is an n-bit URP denoted by P and omit the case when the underlying blockcipher is a PRP.

Theorem 3. *The security bounds of* OCB-hc-AD$_{\mathsf{P}}$ *are evaluated as follows:*

$$\mathbf{Adv}^{\mathrm{priv}}_{\text{OCB-hc-AD}_{\mathsf{P}}}(\mathcal{A}) \leq \frac{4.5\sigma^2_{\mathrm{priv}}}{2^n}, \ \mathbf{Adv}^{\mathrm{auth}}_{\text{OCB-hc-AD}_{\mathsf{P}}}(\mathcal{A}^{\pm}) \leq \frac{4.5\sigma^2_{\mathrm{auth}}}{2^n} + \frac{4q_d}{2^{n/2}},$$

where $\mathcal{A}$, $\mathcal{A}^{\pm}$ are the adversaries against OCB-hc-AD *and σ_{priv}, σ_{auth}, q_d are the parameters for $\mathcal{A}$, $\mathcal{A}^{\pm}$. The parameter σ_{priv} (resp. σ_{auth}) is the number of accesses to P in privacy game (resp. authenticity game) and q_d is the number of queries to the decryption oracle in the authenticity game.*

We prove Theorem 3 in Appendix B.

Algorithm: OCB-hc-AD.$\mathcal{E}_{E_K}(N, A, M)$

1. Auth $\leftarrow$ Phash-hc$_{E_K}(A)$,
2. **return** OCB-hc′.$\mathcal{E}_{E_K}(N, \text{Auth}, M)$

Algorithm: OCB-hc-AD.$\mathcal{D}_{E_K}(N, A, C, T)$

1. Auth $\leftarrow$ Phash-hc$_{E_K}(A)$,
2. **return** OCB-hc′.$\mathcal{D}_{E_K}(N, \text{Auth}, C, T)$

Algorithm: OCB-hc′.$\mathcal{E}_{E_K}(N, \text{Auth}, M)$

1. $M[1] \| \cdots \| M[m-1] \| M[m] \xleftarrow{n} M$
2. $\Delta \leftarrow 2E_K(\mathtt{ozp}(N))$, $\Sigma \leftarrow$ Auth
3. **for** $i \leftarrow 1$ **to** $m-1$ **do**
4. $\quad C[i] \leftarrow E_K(M[i] \oplus \Delta) \oplus \Delta$
5. $\quad \Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(M[i])$
6. $\quad \Delta \leftarrow 2\Delta$
7. Pad $\leftarrow E_K(0^n \oplus \Delta) \oplus \Delta$
8. $\Delta \leftarrow 3\Delta$
9. $C[m] \leftarrow \mathtt{msb}_{\mathtt{len}(M[m])}(\text{Pad} \oplus \mathtt{ozp}(M[m]))$
10. $\Delta \leftarrow \Delta \oplus \mathtt{ozp}(M[m])$
11. Tag $\leftarrow E_K(\Sigma \| 0^{n/2} \oplus \Delta)$
12. **if** $|M[m]| = n$ **then** $T \leftarrow \mathtt{msb}_{n/2}(\text{Tag})$
13. **else** $T \leftarrow \mathtt{lsb}_{n/2}(\text{Tag})$
14. **return** $(C[1] \| \cdots \| C[m], T)$

Algorithm: OCB-hc′.$\mathcal{D}_{E_K}(N, \text{Auth}, C, T)$

1. $C[1] \| \cdots \| C[m-1] \| C[m] \xleftarrow{n} C$
2. $\Delta \leftarrow 2E_K(\mathtt{ozp}(N))$, $\Sigma \leftarrow$ Auth
3. **for** $i \leftarrow 1$ **to** $m-1$ **do**
4. $\quad M[i] \leftarrow D_K(C[i] \oplus \Delta) \oplus \Delta$
5. $\quad \Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(M[i])$
6. $\quad \Delta \leftarrow 2\Delta$
7. Pad $\leftarrow E_K(0^n \oplus \Delta) \oplus \Delta$
8. $\Delta \leftarrow 3\Delta$
9. $M[m] \leftarrow \mathtt{msb}_{\mathtt{len}(C[m])}(\text{Pad}) \oplus C[m]$
10. $\Delta \leftarrow \Delta \oplus \mathtt{ozp}(M[m])$
11. Tag $\leftarrow E_K(\Sigma \| 0^{n/2} \oplus \Delta)$
12. **if** $|M[m] = n$ **then** $T' \leftarrow \mathtt{msb}_{n/2}(\text{Tag})$
13. **else** $T' \leftarrow \mathtt{lsb}_{n/2}(\text{Tag})$
14. **if** $T = T'$ **then return** $M[1] \| \cdots \| M[m]$
15. **else return** $\perp$

Algorithm: Phash-hc$_{E_K}(A)$

1. **if** $A = \varepsilon$ **then** Auth $\leftarrow 0^{n/2}$, **return** Auth
2. $A[1] \| \cdots \| A[a-1] \| A[a] \xleftarrow{n} A$
3. $\Delta \leftarrow 2E_K(0^n)$, Auth $\leftarrow 0^{n/2}$
4. **for** $i \leftarrow 1$ **to** $a-1$ **do**
5. $\quad$ Auth $\leftarrow$ Auth $\oplus$ $\mathtt{msb}_{n/2}(E_K(A[i] \oplus \Delta))$
6. $\quad \Delta \leftarrow 2\Delta$
7. $Y \leftarrow E_K(\mathtt{ozp}(A[a]) \oplus \Delta)$
8. **if** $|A[a]| = n$ **then** Auth $\leftarrow$ Auth$\oplus\mathtt{msb}_{n/2}(Y)$
9. **else** Auth $\leftarrow$ Auth $\oplus$ $\mathtt{lsb}_{n/2}(Y)$
10. **return** Auth

Algorithm: $\mathbb{P}$hash-hc$_{\widetilde{E}}(A)$

1. **if** $A = \varepsilon$ **then** Auth $\leftarrow 0^{n/2}$, **return** Auth
2. $A[1] \| \cdots \| A[a-1] \| A[a] \xleftarrow{n} A$
3. Auth $\leftarrow 0^{n/2}$
4. **for** $i \leftarrow 1$ **to** $a-1$ **do**
5. $\quad$ Auth $\leftarrow$ Auth $\oplus$ $\mathtt{msb}_{n/2}(\widetilde{E}^{0^n,0,i,0}(A[i]))$
6. $Y \leftarrow \widetilde{E}^{0^n,0,a,0}(\mathtt{ozp}(A[a]))$
7. **if** $|A[a]| = n$ **then** Auth $\leftarrow$ Auth$\oplus\mathtt{msb}_{n/2}(Y)$
8. **else** Auth $\leftarrow$ $\mathtt{lsb}_{n/2}(Y)$
9. **return** Auth

Algorithm: ΘCB-hc-AD.$\mathcal{E}_{\widetilde{E}}(N, A, M)$

1. Auth $\leftarrow$ $\mathbb{P}$hash-hc$_{\widetilde{E}}(A)$,
2. **return** ΘCB-hc′.$\mathcal{E}_{\widetilde{E}}(N, \text{Auth}, M)$

Algorithm: ΘCB-hc-AD.$\mathcal{D}_{\widetilde{E}}(N, A, C, T)$

1. Auth $\leftarrow$ $\mathbb{P}$hash-hc$_{\widetilde{E}}(A)$,
2. **return** ΘCB-hc′.$\mathcal{D}_{\widetilde{E}}(N, \text{Auth}, C, T)$

Algorithm: ΘCB-hc′.$\mathcal{E}_{\widetilde{E}}(N, \text{Auth}, M)$

1. $M[1] \| \cdots \| M[m-1] \| M[m] \xleftarrow{n} M$
2. $\Sigma \leftarrow$ Auth, $N \leftarrow \mathtt{ozp}(N)$
3. **for** $i \leftarrow 1$ **to** $m-1$ **do**
4. $\quad C[i] \leftarrow \tilde{E}^{N,1,i,0}(M[i])$
5. $\quad \Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(M[i])$
6. Pad $\leftarrow \tilde{E}^{N,1,m,0}(0^n)$
7. $C[m] \leftarrow M[m] \oplus \text{msb}_{|M[m]|}(\text{Pad})$
8. $\mathtt{Checksum} \leftarrow (\Sigma \| 0^{n/2}) \oplus \mathtt{ozp}(M[m])$
9. Tag $\leftarrow \tilde{E}^{N,0,m,1}(\mathtt{Checksum})$
10. **if** $|M[m]| = n$ **then** $T \leftarrow \mathtt{msb}_{n/2}(\text{Tag})$
11. **else** $T \leftarrow \mathtt{lsb}_{n/2}(\text{Tag})$
12. **return** $(C[1] \| \cdots \| C[m], T)$

Algorithm: ΘCB-hc.$\mathcal{D}_{\widetilde{E}}(N, \text{Auth}, C, T)$

1. $C[1] \| \cdots \| C[m-1] \| C[m] \xleftarrow{n} C$
2. $\Sigma \leftarrow$ Auth, $N \leftarrow \mathtt{ozp}(N)$
3. **for** $i \leftarrow 1$ **to** $m-1$ **do**
4. $\quad M[i] \leftarrow \tilde{D}^{N,1,i,0}(C[i])$
5. $\quad \Sigma \leftarrow \Sigma \oplus \mathtt{msb}_{n/2}(M[i])$
6. Pad $\leftarrow \tilde{E}^{N,1,m,0}(0^n)$
7. $M[m] \leftarrow C[m] \oplus \text{msb}_{|C[m]|}(\text{Pad})$
8. $\mathtt{Checksum} \leftarrow (\Sigma \| 0^{n/2}) \oplus \mathtt{ozp}(M[m])$
9. Tag $\leftarrow \tilde{E}^{N,0,m,1}(\mathtt{Checksum})$
10. **if** $|M[m] = n$ **then** $T' \leftarrow \mathtt{msb}_{n/2}(\text{Tag})$
11. **else** $T' \leftarrow \mathtt{lsb}_{n/2}(\text{Tag})$
12. **if** $T = T'$ **then return** $M[1] \| \cdots \| M[m]$
13. **else return** $\perp$

Fig. 7. The algorithms of OCB-hc-AD and ΘCB-hc-AD. E_K is any blockcipher and D_K is the decryption of E_K. $\widetilde{E}$ is any TBC which has the same arguments as XEX* and $\tilde{D}$ is the decryption of $\widetilde{E}$. Note that $\widetilde{E}$ in **Algorithm:** $\mathbb{P}$hash-hc$_{\widetilde{E}}(A)$ can also be interpreted as a TBC which has the same arguments as XE.

6 Discussion on the security bounds of proposals

In the preceding section, we proved OCB-hc and OTR-hc keep the birthday-bound security as their originals (OCB and OTR). We here compare the security bounds of our proposals when the security parameters (*e.g.* the number of queries) are less than $2^{n/2}$.

For privacy-adversary, our proposals and originals have the exactly same security bounds, respectively, thus we focus on the authenticity-adversary.

We first compare the security bound of OCB-hc to that of OCB. For arbitrary tag length τ up to n, the security bound of $\mathbf{Adv}^{\mathrm{auth}}_{\text{OCB-hc}_\mathsf{P}}(\mathcal{A}^{\pm})$ is evaluated to $4.5\sigma^2_{\mathrm{auth}}/2^n + 2^{n-\tau}q_d/(2^n-1) + 2^{n/2}q_d/(2^n-1)$ in the same manner as the proof in Section 4.3. The security bound of $\mathbf{Adv}^{\mathrm{auth}}_{\text{OCB}_\mathsf{P}}(\mathcal{A}^{\pm})$ is evaluated to $4.5\sigma^2_{\mathrm{auth}}/2^n + 2^{n-\tau}q_d/(2^n-1)$. In the case of $\tau = n/2$, OCB-hc and OCB have the same security bounds except the constant factor. Therefore, $\mathbf{Adv}^{\mathrm{auth}}_{\text{OCB-hc}_\mathsf{P}}(\mathcal{A}^{\pm})$ and $\mathbf{Adv}^{\mathrm{auth}}_{\text{OCB}_\mathsf{P}}(\mathcal{A}^{\pm})$ grow with the same rate except the constant factor when $0 < \sigma_{\mathrm{auth}}, q_d < 2^{n/2}$. In the case of $\tau < n/2$ and $\sigma_{\mathrm{auth}} \approx q_d$, the security bounds of OCB-hc and OCB are $O(q_d/2^\tau)$. Therefore, there is no difference in their bounds except the constant factor when $0 < \sigma_{\mathrm{auth}}, q_d < 2^{n/2}$ similarly to the case of $\tau = n/2$. In the case of $n/2 < \tau \leq n$, the security bound of OCB-hc still has the term $O(q_d/2^{n/2})$, which is not included by that of OCB. If we assume $\sigma_{\mathrm{auth}} \approx q_d$ and $0 < \sigma_{\mathrm{auth}}, q_d < 2^{n/2}$, we obtain $O(\sigma^2/2^n) < O(q_d/2^{n/2})$. Therefore, $\mathbf{Adv}^{\mathrm{auth}}_{\text{OCB}_\mathsf{P}}(\mathcal{A}^{\pm}) < \mathbf{Adv}^{\mathrm{auth}}_{\text{OCB-hc}_\mathsf{P}}(\mathcal{A}^{\pm})$ always holds when $0 < \sigma_{\mathrm{auth}} \approx q_d < 2^{n/2}$. It indicates the security bound of OCB is better when $n/2 < \tau \leq n$ and $0 < \sigma_{\mathrm{auth}} \approx q_d < 2^{n/2}$. The comparison of OTR-hc with OTR will be similar as above, thus we omit the details.

7 Conclusion

In this paper, we have proposed the half-checksum method to reduce the state size of parallel AE mode of operations having birthday-bound security. It maintains the bit security and overall efficiency. We have applied it to two representative parallel AE modes, OCB and OTR, to derive the concrete instantiations, OCB-hc and OTR-hc. They have almost same properties of OCB and OTR (*e.g.* parallelizability, efficiency, bit security, etc) except the reduced state size. When n is block length of the underlying blockcipher, OCB-hc has $2.5n$-bit state size, and OTR-hc has $3.5n$-bit state size. To the best of our knowledge, they achieve the smallest state size among the parallel, rate-1 AE modes of birthday security. Our method is applicable to other schemes having a similar structure as OCB or OTR, such as OPP [22]. While OCB-hc and OTR-hc are plain AE of fixed $n/2$-bit tag length, we presented the natural extensions of them to AEAD with arbitrary tag length up to n bits, without loss of security and increase of state size. It would be interesting to consider if we can apply the same method to other types of parallel AE, such as parallel online AE including COLM [3], COPA [5,6] and ELmD [17,18]. In addition, further study in hardware are required to evaluate actual circuit gain of our proposals. Finally, it would be natural to ask if

the state size figures of our proposals are the theoretical minimum for parallel AE mode of birthday-bound security.

Acknowledgements

We would like to thank the anonymous reviewers for their comments and suggestions.

References

1. The OCB Authenticated-Encryption Algorithm. IRTF RFC 7253 (2014)
2. NIST Lightweight Cryptography Standardization (2019), https://csrc.nist.gov/Projects/Lightweight-Cryptography
3. Andreeva, E., Bogdanov, A., Datta, N., Luykx, A., Mennink, B., Nandi, M., Tischhauser, E., Yasuda, K.: Colm v1. Submission to CAESAR competition (2015)
4. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Mouha, N., Yasuda, K.: How to securely release unverified plaintext in authenticated encryption. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 8873, pp. 105–125. Springer (2014)
5. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: Parallelizable and authenticated online ciphers. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 8269, pp. 424–443. Springer (2013)
6. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: Aes–copa v.2. Submission to CAESAR competition (2015)
7. Aoki, K., Yasuda, K.: The security of the OCB mode of operation without the SPRP assumption. In: ProvSec. Lecture Notes in Computer Science, vol. 8209, pp. 202–220. Springer (2013)
8. Ashur, T., Dunkelman, O., Luykx, A.: Boosting authenticated encryption robustness with minimal modifications. In: CRYPTO (3). Lecture Notes in Computer Science, vol. 10403, pp. 3–33. Springer (2017)
9. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: FOCS. pp. 394–403. IEEE Computer Society (1997)
10. Bellare, M., Goldreich, O., Mityagin, A.: The power of verification queries in message authentication and authenticated encryption. Cryptology ePrint Archive, Report 2004/309 (2004), https://eprint.iacr.org/2004/309
11. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: ASIACRYPT. Lecture Notes in Computer Science, vol. 1976, pp. 531–545. Springer (2000)
12. Benadjila, R., Guo, J., Lomné, V., Peyrin, T.: Implementing lightweight block ciphers on x86 architectures. In: Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 8282, pp. 324–351. Springer (2013)
13. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Duplexing the sponge: Single-pass authenticated encryption and other applications. In: Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 7118, pp. 320–337. Springer (2011)
14. Bhaumik, R., Nandi, M.: Improved security for OCB3. In: ASIACRYPT (2). Lecture Notes in Computer Science, vol. 10625, pp. 638–666. Springer (2017)

15. Bost, R., Sanders, O.: Trick or tweak: On the (in)security of otr's tweaks. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 10031, pp. 333–353 (2016)
16. Chakraborti, A., Iwata, T., Minematsu, K., Nandi, M.: Blockcipher-based authenticated encryption: How small can we go? In: CHES. Lecture Notes in Computer Science, vol. 10529, pp. 277–298. Springer (2017)
17. Datta, N., Nandi, M.: Elme: A misuse resistant parallel authenticated encryption. In: ACISP. Lecture Notes in Computer Science, vol. 8544, pp. 306–321. Springer (2014)
18. Datta, N., Nandi, M.: Elmd v2.0. Submission to CAESAR competition (2015)
19. Dworkin, M.: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. NIST-SP 800-38D (2007)
20. Dworkin, M.: Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality. NIST-SP 800-38C (2007)
21. Ferguson, N.: Collision attacks on OCB. Comments to NIST (2002)
22. Granger, R., Jovanovic, P., Mennink, B., Neves, S.: Improved masking for tweakable blockciphers with applications to authenticated encryption. In: EUROCRYPT (1). Lecture Notes in Computer Science, vol. 9665, pp. 263–293. Springer (2016)
23. Inoue, A., Iwata, T., Minematsu, K., Poettering, B.: Cryptanalysis of ocb2: Attacks on authenticity and confidentiality. Cryptology ePrint Archive, Report 2019/311 (2019), https://eprint.iacr.org/2019/311
24. Iwata, T., Kurosawa, K.: OMAC: one-key CBC MAC. In: FSE. Lecture Notes in Computer Science, vol. 2887, pp. 129–153. Springer (2003)
25. Iwata, T., Minematsu, K., Guo, J., Morioka, S.: CLOC: authenticated encryption for short input. In: FSE. Lecture Notes in Computer Science, vol. 8540, pp. 149–167. Springer (2014)
26. Iwata, T., Minematsu, K., Guo, J., Morioka, S., Kobayashi, E.: CLOC and SILC v3. Submission to the CAESAR competition (2016)
27. Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers. pp. 306–327 (2011). https://doi.org/10.1007/978-3-642-21702-9_18, https://doi.org/10.1007/978-3-642-21702-9_18
28. Liskov, M., Rivest, R.L., Wagner, D.A.: Tweakable block ciphers. In: CRYPTO. Lecture Notes in Computer Science, vol. 2442, pp. 31–46. Springer (2002)
29. Matsuda, S., Moriai, S.: Lightweight cryptography for the cloud: Exploit the power of bitslice implementation. In: CHES. Lecture Notes in Computer Science, vol. 7428, pp. 408–425. Springer (2012)
30. Minematsu, K.: Improved security analysis of XEX and LRW modes. In: Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 4356, pp. 96–113. Springer (2006)
31. Minematsu, K.: Parallelizable rate-1 authenticated encryption from pseudorandom functions. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 8441, pp. 275–292. Springer (2014)
32. Minematsu, K.: Aes-otr v3. Submission to CAESAR competition (2016)
33. Minematsu, K., Matsushima, T.: Generalization and extension of XEX* mode. IEICE Transactions **92-A**(2), 517–524 (2009), http://search.ieice.org/bin/summary.php?id=e92-a_2_517&category=A&year=2009&lang=E&abst=
34. Naito, Y.: Improved xkx-based aead scheme: Removing the birthday terms. Latincrypt 2017 (2017)

35. Naito, Y.: Tweakable blockciphers for efficient authenticated encryptions with beyond the birthday-bound security. IACR Trans. Symmetric Cryptol. **2017**(2), 1–26 (2017)
36. Naito, Y., Matsui, M., Sugawara, T., Suzuki, D.: SAEB: A lightweight blockcipher-based AEAD mode of operation. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2018**(2), 192–217 (2018). https://doi.org/10.13154/tches.v2018.i2.192-217, https://doi.org/10.13154/tches.v2018.i2.192-217
37. Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446 (Aug 2018). https://doi.org/10.17487/RFC8446, https://rfc-editor.org/rfc/rfc8446.txt
38. Rogaway, P.: Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In: Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings. pp. 16–31 (2004). https://doi.org/10.1007/978-3-540-30539-2_2, https://doi.org/10.1007/978-3-540-30539-2_2
39. Rogaway, P., Bellare, M., Black, J., Krovetz, T.: OCB: a block-cipher mode of operation for efficient authenticated encryption. In: CCS 2001, Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, Pennsylvania, USA, November 6-8, 2001. pp. 196–205 (2001). https://doi.org/10.1145/501983.502011, https://doi.org/10.1145/501983.502011
40. T. Dierks, E.R.: The Transport Layer Security (TLS) Protocol Version 1.2. IETF, RFC 5246 (2008)
41. Ueno, R., Homma, N., Iida, T., Minematsu, K.: High throughput/gate fn-based hardware architectures for aes-otr. In: 2019 IEEE International Symposium on Circuits and Systems (ISCAS). pp. 1–4 (2019)

A Proof of security of $\mathbb{P}$hash-hc

We here show the proof of Lemma 3. Note that the underlying TURP $\widetilde{\mathsf{P}}$ has the same arguments as XE in Lemma 3, however we here write $\widetilde{\mathsf{P}}$ with the arguments of XEX* following Fig. 7. Thus we always use $\widetilde{\mathsf{P}}^{*,0,*,*}$ in this proof.

Proof. We define $\mathrm{XorColl}_\delta := \Pr\left[\mathbb{P}\text{hash-hc}_{\widetilde{\mathsf{P}}}(A) \oplus \mathbb{P}\text{hash-hc}_{\widetilde{\mathsf{P}}}(A') = \delta\right]$.

1. Let $A = \varepsilon$ and $A' \neq \varepsilon$.
   (i) We first consider the case of $|A'|_n = 1$. Suppose $\mathrm{ifPad}(A') = 0$ without loss of generality. In this case,

$$\begin{aligned}\mathbb{P}\text{hash-hc}_{\widetilde{\mathsf{P}}}(A) \oplus \mathbb{P}\text{hash-hc}_{\widetilde{\mathsf{P}}}(A') &= \mathbb{P}\text{hash-hc}_{\widetilde{\mathsf{P}}}(A') \\ &= \mathtt{msb}_{n/2}(\widetilde{\mathsf{P}}^{0^n,0,1,0}(\mathtt{ozp}(A'[1])))\end{aligned}$$

   holds. Thus we obtain $\mathrm{XorColl}_{\forall\delta} \leq 1/2^{n/2}$.
   (ii) Let $|A'|_n > 1$. $\mathbb{P}\text{hash-hc}(A')$ is a sum of the most (or least) significant $n/2$ bits of message blocks encrypted by TURPs which are invoked with respective different tweaks. Thus $\mathrm{XorColl}_{\forall\delta} = \Pr\left[\mathbb{P}\text{hash-hc}_{\widetilde{\mathsf{P}}}(A') = \delta\right] \leq 1/2^{n/2}$. This discussion can be applied to the case that $A \neq \varepsilon$ and $A' = \varepsilon$. In following cases, we suppose $A \neq \varepsilon$ and $A' \neq \varepsilon$.

2. Let $|A|_n = |A'|_n$ and $\mathrm{ifPad}(A) = \mathrm{ifPad}(A')$. Suppose $|A|_n = |A'|_n = a$. Without loss of generality, we suppose $\mathrm{ifPad}(A) = \mathrm{ifPad}(A') = 0$. Since $A \neq A'$, there exists $u \in \{1, \ldots, a\}$ such that $A[u] \neq A'[u]$. For $\exists \gamma \in \{0,1\}^{n/2}$, $\mathbb{P}\text{hash-hc}_{\widetilde{\mathsf{P}}}(A) \oplus \mathbb{P}\text{hash-hc}_{\widetilde{\mathsf{P}}}(A') = \mathtt{msb}_{n/2}\left(\widetilde{\mathsf{P}}^{0^n,0,u,0}(\mathtt{ozp}(A[u]))\right) \oplus \mathtt{msb}_{n/2}\left(\widetilde{\mathsf{P}}^{0^n,0,u,0}(\mathtt{ozp}(A'[u]))\right) \oplus \gamma$ holds. Then we obtain

$$\begin{aligned}
&\mathrm{XorColl}_\delta \\
&= \Pr\left[\mathtt{msb}_{n/2}\left(\widetilde{\mathsf{P}}^{0^n,0,u,0}(\mathtt{ozp}(A[u]))\right) \oplus \widetilde{\mathsf{P}}^{0^n,0,u,0}(\mathtt{ozp}(A'[u]))\right) = \delta \oplus \gamma\Big] \\
&\leq 2^{n/2}/(2^n - 1) \leq 2/2^{n/2}.
\end{aligned}$$

3. Let $|A|_n = |A'|_n$ and $\mathrm{ifPad}(A) \neq \mathrm{ifPad}(A')$. Suppose $|A|_n = |A'|_n = a$. Without loss of generality, we suppose $\mathrm{ifPad}(A) = 0$. Since $\mathrm{ifPad}(A) \neq \mathrm{ifPad}(A')$ holds, the case which satisfies $A[a] \neq A'[a]$ and $A[a] = \mathtt{ozp}(A'[a])$ can occur. When $A[a] = \mathtt{ozp}(A'[a])$, we obtain the following evaluation.

$$\begin{aligned}
&\mathrm{XorColl}_{\forall\delta} \\
&= \Pr\left[\mathtt{msb}_{n/2}\left(\widetilde{\mathsf{P}}^{0^n,0,a,0}(A[a])\right) \oplus \mathtt{lsb}_{n/2}\left(\widetilde{\mathsf{P}}^{0^n,0,a,0}(\mathtt{ozp}(A'[a]))\right) = \delta \oplus \gamma\right] \\
&\leq 1/2^{n/2},
\end{aligned}$$

where $\gamma = \mathbb{P}\text{hash-hc}(A) \oplus \mathbb{P}\text{hash-hc}(A') \oplus \mathtt{msb}_{n/2}\left(\widetilde{\mathsf{P}}^{0^n,0,a,0}(A[a])\right) \oplus \mathtt{lsb}_{n/2}\left(\widetilde{\mathsf{P}}^{0^n,0,a,0}(\mathtt{ozp}(A'[a]))\right)$. When $A[a] \neq \mathtt{ozp}(A'[a])$, we also obtain

$$\begin{aligned}
&\mathrm{XorColl}_{\forall\delta} \\
&= \Pr\left[\mathtt{msb}_{n/2}\left(\widetilde{\mathsf{P}}^{0^n,0,a,0}(A[a])\right) \oplus \mathtt{lsb}_{n/2}\left(\widetilde{\mathsf{P}}^{0^n,0,a,0}(\mathtt{ozp}(A'[a]))\right) = \delta \oplus \gamma\right] \\
&\leq 2^{n/2}/(2^n - 1) \leq 2/2^{n/2}.
\end{aligned}$$

From these discussions, $\mathrm{XorColl}_{\forall\delta} \leq 2/2^{n/2}$ holds.

4. Let $|A|_n \neq |A'|_n$. Suppose $|A|_n = a$ and $|A'|_n = a'$. We also suppose $|A|_n < |A'|_n$ and $\mathrm{ifPad}(A') = 0$ without loss of generality. There exists $u \in \mathbb{N}$ such that $a + 1 \leq u \leq a'$ and we obtain the following evaluation.

$$\mathrm{XorColl}_{\forall\delta} = \Pr\left[\mathtt{msb}_{n/2}\left(\widetilde{\mathsf{P}}^{0^n,0,u,0}(\mathtt{ozp}(A'[u]))\right) = \delta \oplus \gamma\right] \leq 1/2^{n/2},$$

where $\gamma = \mathbb{P}\text{hash-hc}(A) \oplus \mathbb{P}\text{hash-hc}(A') \oplus \mathtt{msb}_{n/2}\left(\widetilde{\mathsf{P}}^{0^n,0,u,0}(\mathtt{ozp}(A'[u]))\right)$.

From above four cases, $\max_{\forall\delta \in \{0,1\}^{n/2}} \Pr\left[\mathbb{P}\text{hash-hc}_{\widetilde{\mathsf{P}}}(A) \oplus \mathbb{P}\text{hash-hc}_{\widetilde{\mathsf{P}}}(A') = \delta\right] \leq 2/2^{n/2}$ holds.

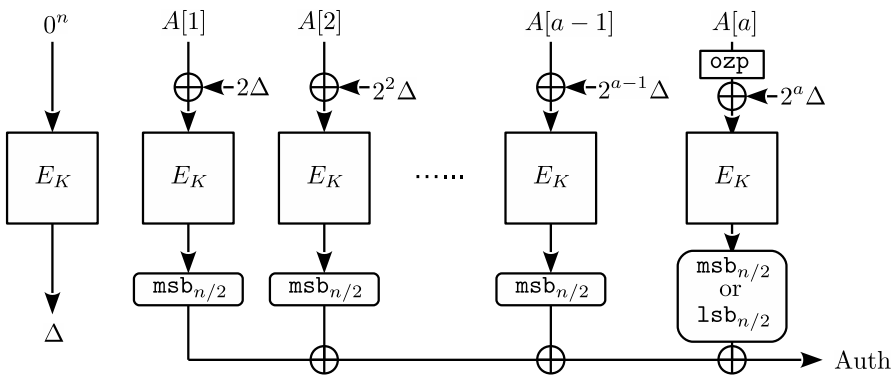

Fig. 8. The algorithm of Phash-hc$_{E_K}$, where E_K is any blockcipher.

B Proof of the security of OCB-hc-AD

We here show the proof of Theorem 3.

Proof. We obtain the following evaluations using hybrid argument.

$$\begin{aligned}
\mathbf{Adv}^{\text{priv}}_{\text{OCB-hc-AD}_\mathsf{P}}(\mathcal{A}) &\le \mathbf{Adv}^{\text{cpa-nr}}_{\text{OCB-hc-AD}_\mathsf{P},\Theta\text{CB-hc-AD}_{\tilde{\mathsf{P}}}}(\mathcal{A}) + \mathbf{Adv}^{\text{priv}}_{\Theta\text{CB-hc-AD}_{\tilde{\mathsf{P}}}}(\mathcal{A}) \\
&= \mathbf{Adv}^{\text{tprp}}_{\text{XEX}^*_{\tilde{\mathsf{P}}}}(\mathcal{B}) + \mathbf{Adv}^{\text{priv}}_{\Theta\text{CB-hc-AD}_{\tilde{\mathsf{P}}}}(\mathcal{A}) \\
&\le \frac{4.5\sigma^2_{\text{priv}}}{2^n} + 0, \qquad (5) \\
\mathbf{Adv}^{\text{auth}}_{\text{OCB-hc-AD}_\mathsf{P}}(\mathcal{A}^{\pm}) &\le \mathbf{Adv}^{\text{cca-nr}}_{\text{OCB-hc-AD}_\mathsf{P},\Theta\text{CB-hc-AD}_{\tilde{\mathsf{P}}}}(\mathcal{A}^{\pm}) + \mathbf{Adv}^{\text{auth}}_{\Theta\text{CB-hc-AD}_{\tilde{\mathsf{P}}}}(\mathcal{A}^{\pm}) \\
&= \mathbf{Adv}^{\text{tsprp}}_{\text{XEX}^*_{\tilde{\mathsf{P}}}}(\mathcal{B}^{\pm}) + \mathbf{Adv}^{\text{auth}}_{\Theta\text{CB-hc-AD}_{\tilde{\mathsf{P}}}}(\mathcal{A}^{\pm}) \\
&\le \frac{4.5\sigma^2_{\text{auth}}}{2^n} + \frac{4q_d}{2^{n/2}}, \qquad (6)
\end{aligned}$$

where $\mathcal{B}$ (resp. $\mathcal{B}^{\pm}$) is the adversary which can simulate $\mathcal{A}$ (resp. $\mathcal{A}^{\pm}$). The first terms of (5), (6) are derived from [38], [33]. The second terms of (5), (6) are described below.

Privacy. Similarly to ΘCB-hc and ΘTR-hc, $\mathbf{Adv}^{\text{priv}}_{\Theta\text{CB-hc-AD}_{\tilde{\mathsf{P}}}}(\mathcal{A}) = 0$ holds since the adversary follows nonce-respecting.

Authenticity. For simplicity, we suppose that the adversary can query to the decryption oracle only once. Without loss of generality, the adversary performs decryption query after all encryption queries. Suppose that she obtains the transcript $z = \{(N_1, M_1, A_1, C_1, T_1), \ldots, (N_q, M_q, A_q, C_q, T_q)\}$ in encryption query, and she queries (N', A', C', T') in decryption query. Let Z be the set of all transcripts, and T^* be the valid tag for (N', A', C'). Then we define $\text{FP}_z := \Pr[T' = T^* \mid Z = z]$ and evaluate $\max_z \text{FP}_z$ as below.

1. Let $N' \neq N_i$, $1 \leq \forall i \leq q$. As in the proof of ΘCB-hc, $\mathrm{FP}_z \leq 1/2^{n/2}$ holds.
2. Let $N' = N_\alpha$, $\alpha \in \{1, 2, \ldots, q\}$, $A' = A_\alpha$, $C' \neq C_\alpha$. In this case, we can evaluate FP_z in the same manner as the proof of ΘCB-hc. Thus $\mathrm{FP}_z \leq 4/2^{n/2}$ holds.
3. Let $N' = N_\alpha$, $\alpha \in \{1, 2, \ldots, q\}$, $A' \neq A_\alpha$. We suppose that $\mathtt{Checksum}^*$ is the valid checksum corresponding to (N', A', C') and that $\mathtt{Checksum}_\alpha$ is the value of the checksum corresponding to $(N_\alpha, A_\alpha, C_\alpha)$. Let e_1 is the event which $\mathtt{Checksum}^* = \mathtt{Checksum}_\alpha$ holds. Recall that

$$\mathtt{Checksum} = \left(\left(\mathbb{P}\text{hash-hc}(A) \oplus \bigoplus_{i=1}^{m-1} \mathtt{msb}_{n/2}(M[i]) \right) \| \, 0^{n/2} \right) \oplus \mathtt{ozp}(M[m]).$$

From the property of $\mathbb{P}$hash-hc mentioned in Lemma 3, we obtain the following evaluation.

$$\begin{aligned} \Pr[e_1 \mid Z = z] &= \Pr[\mathbb{P}\text{hash-hc}(A') \| \, 0^{n/2} \oplus \mathbb{P}\text{hash-hc}(A_\alpha) \| \, 0^{n/2} = \gamma \mid Z = z] \\ &\leq \frac{2}{2^{n/2}}, \end{aligned}$$

where $\gamma = \left(\bigoplus_{i=1}^{m'-1} \mathtt{msb}_{n/2}(M^*[i]) \right) \| \, 0^{n/2} \Big) \oplus \Big(\bigoplus_{i=1}^{m_\alpha - 1} \mathtt{msb}_{n/2}(M_\alpha[i]) \right) \| \, 0^{n/2} \Big) \oplus \mathtt{ozp}(M^*[m']) \oplus \mathtt{ozp}(M_\alpha[m_\alpha])$. Then we can evaluate a forgery probability as follows:

$$\begin{aligned} \mathrm{FP}_z &\leq \Pr[T' = T^* \mid \bar{e_1}, Z = z] \Pr[e_1 \mid Z = z] \\ &\leq \frac{2^{n/2}}{2^n - 1} + \frac{2}{2^{n/2}} \leq \frac{4}{2^{n/2}}. \end{aligned}$$

From the evaluations of above cases, we obtain

$$\mathbf{Adv}^{\mathrm{auth}}_{\Theta\text{CB-hc-AD}}(\mathcal{A}^{\pm}) \leq \sum_z \max_z \mathrm{FP}_z \cdot \Pr[Z = z] \leq \frac{4}{2^{n/2}}.$$

When the adversary queries to the decryption oracle q_d times, we obtain

$$\mathbf{Adv}^{\mathrm{auth}}_{\Theta\text{CB-hc-AD}}(\mathcal{A}^{\pm}) \leq \frac{4q_d}{2^{n/2}}$$

by using a technique from [10].