Parallelizable Authenticated Encryption from Functions

Kazuhiko Minematsu[1]

NEC Corporation, Japan, k-minematsu@ah.jp.nec.com

Abstract. A new authenticated encryption (AE) mode for blockcipher is presented. The proposed scheme has attractive features for fast and compact operation. It requires rate-1 blockcipher call, and uses the encryption function of a blockcipher for both encryption and decryption. Moreover, the scheme enables one-pass, parallel operation under two-block partition. The proposed scheme thus attains similar characteristics as the seminal OCB mode, without using the inverse blockcipher. The key idea of our proposal is a novel usage of two-round Feistel permutation, where the round functions are derived from the theory of tweakable blockcipher.

We also describe an instantiation of our idea using a non-invertible primitive, such as a keyed hash function.

Keywords: Authenticated Encryption, Blockcipher Mode, Pseudorandom Function, OCB.

1 Introduction

Authenticated Encryption. Authenticated encryption, AE for short, is a method to simultaneously provide message confidentiality and integrity (authentication) using a symmetric-key cryptographic function. Although a secure AE function can be basically obtained by an adequate composition of secure encryption and message authentication [6, 19], this requires at least two independent keys, and the composition methods in practice (say, AES + HMAC in TLS) frequently deviate from what proved to be secure [24]. Considering this situation, there have been numerous efforts devoted to efficient, one-key constructions. Among many approaches to AE, blockcipher mode of operation is one of the most popular ones. We have CCM [1], GCM [2], EAX [8], OCB [20, 26, 29] and the variants [14, 18], and CCFB [22], to name a few. We have some standards, such as NIST SP 800-38C (CCM) and 38D (GCM), and ISO/IEC 19772.

This paper presents a new AE mode using a blockcipher, or more generally, a pseudorandom function (PRF). Our proposal has a number of desirable features for fast and compact operations. Specifically, when the underlying n-bit blockcipher is E_K (where K denotes the key), the properties of our proposal can be summarized as follows.

- The key is one blockcipher key, K.
- Encryption and decryption can be done by the encryption function of E_K.
- For s-bit input, the number of E calls is $\lceil s/n \rceil + 2$ or 3, i.e., rate-1 processing, for both encryption and decryption.
- On-line, One-pass and parallel encryption and decryption, under two-block partition.
- Provable secure up to $2^{n/2}$ input blocks, based on the assumption that E is a PRF (or a pseudorandom permutation, PRP).

These features are realized with a novel usage of two-round Feistel permutation, where internal round functions are PRF with input masking. From this we call our proposal OTR, for Offset Two-round. Table 1 provides a summary of properties of popular AE modes and ours. The proposed scheme generates input maskings to E using GF(2^n) constant multiplications, called GF doubling [26], which is a quite popular tool for mode design. However, since our proposal is rather generic and not restricted to GF doubling-based masking, we may have variants with alternative masking method, such as Gray code [20, 29] or word-oriented LFSR [10, 20, 31]. We also remark that Liting et al.'s iFeed mode [32] has similar properties to ours, without introducing 2-block partition. However, its decryption can not be parallelized. In return for these attractive features, one potential drawback of OTR is that it inherently needs two-block partition (though the message itself can be of any length in bits), which implies more state memories required than that of OCB. The parallelizability of our scheme is up to the half of the message blocks, while OCB has full parallelizability, up to the number of message blocks.

We also warn that the security is proved for the standard nonce-respecting adversary [27], i.e. the encryption never processes duplicate nonces (or initial vectors), see Section 2.4. Some recent proposals have a provable security under nonce-non-respecting adversary, or even security without nonce (called on-line encryption) [3,13]. However we do not claim any security guarantee for such adversaries.

Table 1. A comparison of AE modes. Calls denotes the number of calls for message M and header A for single-block nonce, without constants.

Mode	Calls	On-line	Parallel	Primitive	Remark
CCM [1]	$\|A\|_n + 2\|M\|_n$	no	no	E	
GCM [2]	$\|M\|_n$ [E] and $\|A\|_n + \|M\|_n$ [GFmul]	yes	yes	E, GFmul	GFmul over GF(2^{128})
EAX [8]	$\|A\|_n + 2\|M\|_n$	yes	no	E	
OCB [20,26,29]	$\|A\|_n + \|M\|_n$	yes	yes	E, E^{-1}	
CCFB [22]	$\|A\|_n + c\|M\|_n$, for some $1 < c < 2$	yes	no	E	Sec. degrades as $c \to 1$
OTR	$\|A\|_n + \|M\|_n$	yes	yes	E	2-block partition

Benefits of inverse-freeness. The use of blockcipher inversion, as in OCB, has mainly two drawbacks, as discussed by Iwata and Yasuda [17]. The first is on the efficiency. The integration of encryption and decryption functions increases size, e.g. footprint of hardware, or memory of software. Moreover, some ciphers have unequal speed for enc/dec. For AES, its decryption is slower than encryption on some (typically constrained) platforms. This property is the initial design choice [11], in preference of encryption-only mode, e.g., CTR, OFB, and CFB. For instance, an AES implementation on Atmel AVR by Osvik et al. [23] has about 45% slower decryption than encryption. IDEA is another example, where decryption is exceptionally slower than encryption on microcontrollers [25]. The uneven performance figures of blockcipher enc/dec functions may cause problems in practice, when the mode uses both functions.

The second is on the security. Usually the security of a mode using both enc/dec functions of a blockcipher, denoted by E and E^{-1}, needs (E, E^{-1}) to be a strong pseudorandom permutation, SPRP, for its provable security. In contrast, when the mode uses only E, the security assumption is relaxed to PRP or PRF.

In addition, the inverse-freeness allows instantiations using non-blockcipher primitives. We provide an example that uses a variable-input-length PRF, which may be instantiated with HMAC based on a hash function, or a dedicated scheme, such as SipHash [4]. In this case the input masking may be simplified to input prepending (with a small computation overhead), and the security of such an instantiation is roughly n bits when the underlying PRF has n-bit output.

Hardware assistance. We remark that some software platforms have hardware-assisted blockcipher, most notably AESNI in Intel and AMD CPUs. AESNI enables the same performance for AES encryption and decryption. Therefore, when our proposal uses AESNI, the performance would be roughly similar to that of OCB-AES with AESNI, though the increased number of states may degrade the result. We have other SW platforms where hardware AES is available but the decryption is slower (e.g., see [15]). Basically, the value of our proposal is *not* to provide the fastest operation on modern CPUs, instead, to increase the availablity of OCB-like performance for various platforms, using single algorithm.

2 Preliminaries

2.1 Basic Notations

Let $\mathbb{N} = \{1, 2, \ldots, \}$. Let $\{0,1\}^*$ be the set of all finite-length binary strings, including the empty string ε. The bit length of a binary string X is written as $|X|$, and let $|X|_a \overset{\text{def}}{=} \max\{\lceil |X|/a \rceil, 1\}$ (if $X = \varepsilon$ we have $|X|_a = 1$ for any $a \geq 1$). Here $|\varepsilon| = 0$. A concatenation of $X, Y \in \{0,1\}^*$ is written as $X \| Y$ or simply XY. A sequence of a zeros (ones) is denoted by 0^a (1^a). For $k \geq 0$, let $\{0,1\}^{>k} \overset{\text{def}}{=} \bigcup_{i=k+1,\ldots} \{0,1\}^i$ and $(\{0,1\}^n)^{>k} \overset{\text{def}}{=} \bigcup_{j=k+1,\ldots} (\{0,1\}^n)^j$, and $(\{0,1\}^n)^+ \overset{\text{def}}{=} (\{0,1\}^n)^{>0}$. We also define $\{0,1\}^{\geq k}$, $(\{0,1\}^n)^{\geq k}$, $\{0,1\}^{<k}$, $(\{0,1\}^n)^{<k}$, $\{0,1\}^{\leq k}$, and $(\{0,1\}^n)^{\leq k}$ analogously.

For $X \in \{0,1\}^*$, let $X[1] \| X[2] \| \ldots \| X[x] \overset{n}{\leftarrow} X$ denote the n-bit block partitioning of X, i.e., $X[1] \| X[2] \| \ldots \| X[x] = X$ where $x = |X|_n$, and $|X[i]| = n$ for $i < x$ and $|X[x]| \leq n$. If $X = \varepsilon$ the parsing makes $x = 1$, $X[1] = \varepsilon$. The sequence of first c bits of $X \in \{0,1\}^*$ is denoted by $\mathrm{msb}_c(X)$. We have $\mathrm{msb}_0(X) = \varepsilon$ for any X. For $X, Y \in \{0,1\}^*$, let $X \oplus_{\text{end}} Y$ be the XOR of X into the end of Y if $|X| \leq |Y|$, i.e. $X \oplus_{\text{end}} Y = (0^{|Y|-|X|} \| X) \oplus Y$. Otherwise $X \oplus_{\text{end}} Y = X \oplus (0^{|X|-|Y|} \| Y)$.

For a finite set $\mathcal{X}$, if X is uniformly chosen from $\mathcal{X}$ we write $X \overset{\$}{\leftarrow} \mathcal{X}$. We assume $X \oplus Y$ is ε if X or Y is ε. For a binary string X with $0 \leq |X| < n$, $X10^*$ denotes the padding written as $X \| 1 \| 0^{n-|X|-1}$, and when $|X| = n$, $X10^*$ denotes X.

Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a keyed function with key $K \in \mathcal{K}$. We may simply write $F_K : \mathcal{X} \to \mathcal{Y}$ if key space is obvious, or even write it as F if being keyed with K is obvious. A tweakable keyed function with tweak space $\mathcal{T}$ is written as $\widetilde{F} : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \to \mathcal{Y}$ or $\widetilde{F}_K : \mathcal{T} \times \mathcal{X} \to \mathcal{Y}$. Instead of writing $\widetilde{F}_K(T, X)$, we may write as $\widetilde{F}_K^{\langle T \rangle}(X)$. Keyed permutation and its tweakable version are defined similarly. For keyed permutation E_K, its inversion is denoted by E_K^{-1}.

2.2 Pseudorandom Function

Let $\mathrm{Func}(n, m)$ be the set of all functions $\{0,1\}^n \to \{0,1\}^m$. We may abbreviate $\mathrm{Func}(n, n)$ to $\mathrm{Func}(n)$. In addition, let $\mathrm{Perm}(n)$ be the set of all permutations over $\{0,1\}^n$. A uniform random function (URF) having n-bit input and m-bit output is the set $\mathrm{Func}(n, m)$ with uniform distribution over $\mathrm{Func}(n, m)$. It is denoted by R, and the corresponding sampling is written as $\mathsf{R} \overset{\$}{\leftarrow} \mathrm{Func}(n, m)$. An n-bit uniform random permutation (URP) is the set $\mathrm{Perm}(n)$ with uniform distribution over $\mathrm{Perm}(n)$. It is denoted by P, and the corresponding sampling is written as $\mathsf{P} \overset{\$}{\leftarrow} \mathrm{Perm}(n)$.

We also define tweakable URF and URP. Let $\mathcal{T}$ be a set of tweak and $\mathrm{Func}^{\mathcal{T}}(n, m)$ be a set of functions $\mathcal{T} \times \{0,1\}^n \to \{0,1\}^m$. A tweakable URF with tweak $T \in \mathcal{T}$, and n-bit input, m-bit output is written as $\widetilde{\mathsf{R}} \overset{\$}{\leftarrow} \mathrm{Func}^{\mathcal{T}}(n, m)$. Note that if $\mathcal{T} = \{0,1\}^t$, $\mathrm{Func}^{\mathcal{T}}(n, m)$ has the same cardinality as $\mathrm{Func}(n + t, m)$, hence $\widetilde{\mathsf{R}}$ is simply realized with URF of $(n + s)$-bit input. In addition, let $\mathrm{Perm}^{\mathcal{T}}(n)$ be a set of functions $\mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ such that, for any $f \in \mathrm{Perm}^{\mathcal{T}}(n)$ and $t \in \mathcal{T}$, $f(t, *)$ is a permutation. A tweakable n-bit URP with tweak $T \in \mathcal{T}$ is defined as $\widetilde{\mathsf{P}} \overset{\$}{\leftarrow} \mathrm{Perm}^{\mathcal{T}}(n)$.

2.3 Galois Field

Following [8], an n-bit string X may be viewed as an element of $\mathrm{GF}(2^n)$ by taking X as a coefficient vector of the polynomial in $\mathrm{GF}(2^n)$. We write $2X$ to denote the multiplication of 2 and X over $\mathrm{GF}(2^n)$, where 2 denotes the generator of the field $\mathrm{GF}(2^n)$. This operation is called *doubling*. We also write $3L$ and $4L$ to denote $2L \oplus L$ and $2(2L)$. The doubling is efficiently implemented by one-bit shift with conditional XOR of a constant, see e.g. [16].

2.4 Security Notions

For c oracles, $O_1, O_2, \ldots, O_c$, we write $\mathcal{A}^{O_1,O_2,\ldots,O_c}$ to represent the adversary $\mathcal{A}$ accessing these c oracles in an arbitrarily order. If O and O' are oracles having the same input and output domains, we say they are compatible.

Let F_K and G'_K be two compatible keyed functions having n-bit input and m-bit output, with $K \in \mathcal{K}$ and $K' \in \mathcal{K}'$ (key spaces are not necessarily the same). Let $\mathcal{A}$ be an adversary trying distinguish them using chosen-plaintext queries. Then the advantage of $\mathcal{A}$ is defined as

$$\mathtt{Adv}^{\mathtt{cpa}}_{F_K,G_{K'}}(\mathcal{A}) \overset{\text{def}}{=} \Pr[K \overset{\$}{\leftarrow} \mathcal{K} : \mathcal{A}^{F_K} \Rightarrow 1] - \Pr[K' \overset{\$}{\leftarrow} \mathcal{K}' : \mathcal{A}^{G_{K'}} \Rightarrow 1].$$

The above definition can be naturally extended to the case where F_K or G'_K is a URF $\mathsf{R} \overset{\$}{\leftarrow} \mathrm{Func}(n,m)$, and we have

$$\mathtt{Adv}^{\mathtt{prf}}_{F_K}(\mathcal{A}) \overset{\text{def}}{=} \mathtt{Adv}^{\mathtt{cpa}}_{F_K,\mathsf{R}}(\mathcal{A}).$$

Similarly, for tweakable keyed function $\widetilde{F}_K : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^m$ and $\widetilde{\mathsf{R}} \overset{\$}{\leftarrow} \mathrm{Func}^{\mathcal{T}}(n,m)$, we have

$$\mathtt{Adv}^{\mathtt{prf}}_{\widetilde{F}_K}(\mathcal{A}) \overset{\text{def}}{=} \mathtt{Adv}^{\mathtt{cpa}}_{\widetilde{F}_K,\widetilde{\mathsf{R}}}(\mathcal{A}).$$

We stress that $\mathcal{A}$ in the above is allowed to choose tweaks, arbitrarily and adaptively.

For keyed n-bit permutations, E_K and $G_{K'}$, we have

$$\mathtt{Adv}^{\mathtt{prp}}_{E_K}(\mathcal{A}) \overset{\text{def}}{=} \mathtt{Adv}^{\mathtt{cpa}}_{E_K,\mathsf{P}}(\mathcal{A}),$$
$$\mathtt{Adv}^{\mathtt{cca}}_{E_K,G_{K'}}(\mathcal{A}) \overset{\text{def}}{=} \Pr[K \overset{\$}{\leftarrow} \mathcal{K} : \mathcal{A}^{E_K,E_K^{-1}} \Rightarrow 1] - \Pr[K' \overset{\$}{\leftarrow} \mathcal{K}' : \mathcal{A}^{G_{K'},G_{K'}^{-1}} \Rightarrow 1], \text{ and}$$
$$\mathtt{Adv}^{\mathtt{sprp}}_{E_K}(\mathcal{A}) \overset{\text{def}}{=} \mathtt{Adv}^{\mathtt{cca}}_{E_K,\mathsf{P}}(\mathcal{A}),$$

where P is an n-bit URP.

Security Notions for AEs. Following [8, 27], we introduce two security notions, privacy and authenticity, to model the security of OTR. Let $\mathsf{AE}[\tau]$ be an AE compatible with OTR having τ-bit tag. The encryption and decryption algorithms are $\mathsf{AE}\text{-}\mathcal{E}_\tau$ and $\mathsf{AE}\text{-}\mathcal{D}_\tau$. A CPA-adversary $\mathcal{A}$ against $\mathsf{AE}[\tau]$ accesses $\mathsf{AE}\text{-}\mathcal{E}_\tau$. Let $(N_1, A_1, M_1), \ldots, (N_q, A_q, M_q)$ be all the encryption queries made by $\mathcal{A}$. We define $\mathcal{A}$'s parameter list to be (q, σ_A, σ_M), where $\sigma_A \overset{\text{def}}{=} \sum_{i=1}^{q} |H_i|_n$ and $\sigma_M \overset{\text{def}}{=} \sum_{i=1}^{q} |M_i|_n$. We also define random-bit oracle, \$, which takes $(N, A, M) \in \mathcal{N} \times \{0,1\}^* \times \{0,1\}^*$ and returns $(C, T) \overset{\$}{\leftarrow} \{0,1\}^{|M|} \times \{0,1\}^\tau$. The privacy notion for CPA-adversary $\mathcal{A}$ is defined as

$$\mathtt{Adv}^{\mathtt{priv}}_{\mathsf{AE}[\tau]}(\mathcal{A}) \overset{\text{def}}{=} \Pr[K \overset{\$}{\leftarrow} \mathcal{K} : \mathcal{A}^{\mathsf{AE}\text{-}\mathcal{E}_\tau} \Rightarrow 1] - \Pr[\mathcal{A}^{\$} \Rightarrow 1]. \tag{1}$$

We assume $\mathcal{A}$ in the privacy notion is nonce-respecting, i.e., all N_is are distinct. Similarly, a CCA-adversary $\mathcal{A}$ against $\mathsf{AE}[\tau]$ accesses $\mathsf{AE}\text{-}\mathcal{E}_\tau$ and $\mathsf{AE}\text{-}\mathcal{D}_\tau$. Let $(N_1, A_1, M_1), \ldots, (N_q, A_q, M_q)$ and $(N'_1, A'_1, C'_1, T'_1), \ldots, (N'_{q_v}, A'_{q_v}, C'_{q_v}, T'_{q_v})$ be all the encryption and decryption queries made by $\mathcal{A}$. We define $\mathcal{A}$'s parameter list to be $(q, q_v, \sigma_A, \sigma_M, \sigma_{A'}, \sigma_{C'})$, where $\sigma_{N'} \overset{\text{def}}{=} \sum_{i=1}^{q_v} |N'_i|_n$ and $\sigma_{A'} \overset{\text{def}}{=} \sum_{i=1}^{q_v} |A'_i|_n$ and $\sigma_{C'} \overset{\text{def}}{=} \sum_{i=1}^{q_v} |C'_i|_n$. Here, σ_A and σ_M are defined as above. The authenticity notion for the CCA-adversary $\mathcal{A}$ is defined as

$$\mathtt{Adv}^{\mathtt{auth}}_{\mathsf{AE}[\tau]}(\mathcal{A}) \overset{\text{def}}{=} \Pr[K \overset{\$}{\leftarrow} \mathcal{K} : \mathcal{A}^{\mathsf{AE}\text{-}\mathcal{E}_\tau,\mathsf{AE}\text{-}\mathcal{D}_\tau} \text{ forges }], \tag{2}$$

where $\mathcal{A}$ forges if $\mathsf{AE}\text{-}\mathcal{D}_\tau$ returns a bit string (other than $\perp$) for a query (N'_i, A'_i, C'_i, T'_i) for some $1 \le i \le q_v$ such that $(N'_i, A'_i, C'_i, T'_i) \ne (N_j, A_j, C_j, T_j)$ for all $1 \le j \le q$. We assume $\mathcal{A}$

in the authenticity notion is always nonce-respecting with respect to encryption queries; using the same N for encryption and decryption queries is allowed, and the same N can be repeated within decryption queries, i.e. N_i is different from N_j for any $j \neq i$ but N'_i may be equal to N_j or $N'_{i'}$ for some j and $i' \neq i$.

Let $\mathbf{F} = (F^e_K, F^d_K)$ and $\mathbf{G} = (G^e_{K'}, G^d_{K'})$ be the pairs of functions that are compatible with $(\mathsf{AE}\text{-}\mathcal{E}_\tau, \mathsf{AE}\text{-}\mathcal{D}_\tau)$. We define

$$\mathtt{Adv}^{\mathtt{cca\text{-}nr}}_{\mathbf{F},\mathbf{G}}(\mathcal{A}) \overset{\text{def}}{=} \Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{F^e_K, F^d_K} \Rightarrow 1] - \Pr[K' \xleftarrow{\$} \mathcal{K}' : \mathcal{A}^{G^e_{K'}, G^d_{K'}} \Rightarrow 1], \tag{3}$$

where the underlying $\mathcal{A}$ is assumed to be nonce-respecting for encryption queries.

3 Specification (Blockcipher-based, GF doubling masks)

We present an AEAD scheme based on an $E_K : \{0,1\}^n \to \{0,1\}^n$, which is denoted by $\mathrm{OTR}[E, \tau]$, where $\tau \in \{1, \ldots, n\}$ denotes the length of tag. The encryption function of $\mathrm{OTR}[E, \tau]$ is denoted by OTR-$\mathcal{E}_{E,\tau}$. The input to OTR-$\mathcal{E}_{E,\tau}$ consists of a nonce $N \in \mathcal{N} = \{0,1\}^{\leq n-1} \setminus \{\varepsilon\}$, a header (or associated data) $A \in \mathcal{A} = \{0,1\}^*$, and a plaintext $M \in \mathcal{M} = \{0,1\}^*$. The output consists of $C \in \{0,1\}^*$ and $T \in \{0,1\}^\tau$, where $|C| = |M|$. The decryption function is denoted by OTR-$\mathcal{D}_{E,\tau}$. It takes $(N, C, A, T) \in \mathcal{N} \times \mathcal{A} \times \mathcal{M} \times \{0,1\}^\tau$, and outputs a plaintext M with $|M| = |C|$ if input is determined as valid, or error symbol $\perp$ if determined as invalid.

These two functions are further decomposed into three functions, the encryption and decryption cores, EF_E, DF_E, and the authentication core, AF_E.

Fig. 1 and Fig. 2 depict the scheme. As shown by Fig. 2, OTR consists of two-round Feistel permutations using a blockcipher taking a distinct input mask in each round. To authenticate the plaintext a check sum is computed for the right part of two-round Feistel (namely the even plaintext blocks), and the tag is derived from the encrypting the check sum with an input mask.

4 Security Bounds

We provide the security bounds of OTR. For simplicity we assume the underlying blockcipher is an n-bit URP, P. The computational counterparts are fairly straightforward, thus omitted.

Theorem 1. *Fix $\tau \in \{1, \ldots, n\}$. For any CPA-adversary $\mathcal{A}$ with parameter (q, σ_A, σ_M),*

$$\mathtt{Adv}^{\mathtt{priv}}_{\mathrm{OTR}[\mathsf{P},\tau]}(\mathcal{A}) \leq \frac{6\sigma^2_{\mathtt{priv}}}{2^n}$$

holds for $\sigma_{\mathtt{priv}} = q + \sigma_A + \sigma_M$.

Theorem 2. *Fix $\tau \in \{1, \ldots, n\}$. For any CCA-adversary $\mathcal{A}$ with parameter $(q, q_v, \sigma_A, \sigma_M, \sigma_{A'}, \sigma_{C'})$,*

$$\mathtt{Adv}^{\mathtt{auth}}_{\mathrm{OTR}[\mathsf{P},\tau]}(\mathcal{A}) \leq \frac{6\sigma^2_{\mathtt{auth}}}{2^n} + \frac{q_v}{2^\tau}$$

holds for $\sigma_{\mathtt{auth}} = q + q_v + \sigma_A + \sigma_M + \sigma_{A'} + \sigma_{C'}$.

5 Proofs of Theorems 1 and 2

5.1 Generic AEAD Construction based on Tweakable URF

To understand the proofs of the above theorems, we first provide a generic construction behind OTR and show the security bounds for it.

Algorithm OTR-$\mathcal{E}_{E,\tau}(N, A, M)$

```
$(C, TE) \leftarrow \mathrm{EF}_E(N, M)$
**if** $A \neq \varepsilon$ **then** $TA \leftarrow \mathrm{AF}_E(A)$
**else** $TA \leftarrow 0^n$
$T \leftarrow \mathrm{msb}_\tau(TE \oplus TA)$
**return** $(C, T)$
```

Algorithm OTR-$\mathcal{D}_{E,\tau}(N, C, A, T)$

```
$(M, TE) \leftarrow \mathrm{DF}_E(N, C)$
**if** $A \neq \varepsilon$ **then** $TA \leftarrow \mathrm{AF}_E(A)$
**else** $TA \leftarrow 0^n$
$\widehat{T} \leftarrow \mathrm{msb}_\tau(TE \oplus TA)$
**if** $\widehat{T} = T$ **return** $M$
**else return** $\perp$
```

Algorithm $\mathrm{EF}_E(N, M)$

```
$\Sigma \leftarrow 0^n$
$L \leftarrow E(N10^*), L' \leftarrow 4L$
$M[1] \| M[2] \| \ldots \| M[m] \xleftarrow{n} M$
**for** $i = 1$ **to** $\lfloor m/2 \rfloor - 1$ **do**
    $C[2i-1] \leftarrow E(L' \oplus M[2i-1]) \oplus M[2i]$
    $C[2i] \leftarrow E(L' \oplus L \oplus C[2i-1]) \oplus M[2i-1]$
    $\Sigma \leftarrow \Sigma \oplus M[2i]$
    $L' \leftarrow 2L'$
**if** $m$ **is even**
    $Z \leftarrow E(L' \oplus M[m-1])$
    $C[m] \leftarrow \mathrm{msb}_{|M[m]|}(Z) \oplus M[m]$
    $C[m-1] \leftarrow E(L' \oplus L \oplus C[m]10^*) \oplus M[m-1]$
    $\Sigma \leftarrow \Sigma \oplus Z \oplus C[m]10^*$
    $L_{\mathrm{last}} \leftarrow L' \oplus L$
**if** $m$ **is odd**
    $C[m] \leftarrow \mathrm{msb}_{|M[m]|}(E(L')) \oplus M[m]$
    $\Sigma \leftarrow \Sigma \oplus M[m]10^*$
    $L_{\mathrm{last}} \leftarrow L'$
**if** $|M[m]| \neq n$ **then** $TE \leftarrow E(3L_{\mathrm{last}} \oplus \Sigma)$
**else** $TE \leftarrow E(3L_{\mathrm{last}} \oplus L \oplus \Sigma)$
$C \leftarrow C[1] \| C[2] \| \ldots \| C[m]$
**return** $(C, TE)$
```

Algorithm $\mathrm{DF}_E(N, C)$

```
$\Sigma \leftarrow 0^n$
$L \leftarrow E(N10^*), L' \leftarrow 4L$
$C[1] \| C[2] \| \ldots \| C[m] \xleftarrow{n} C$
**for** $i = 1$ **to** $\lfloor m/2 \rfloor - 1$ **do**
    $M[2i-1] \leftarrow E(L' \oplus L \oplus C[2i-1]) \oplus C[2i]$
    $M[2i] \leftarrow E(L' \oplus M[2i-1]) \oplus C[2i-1]$
    $\Sigma \leftarrow \Sigma \oplus M[2i]$
    $L' \leftarrow 2L'$
**if** $m$ **is even**
    $M[m-1] \leftarrow E(L' \oplus L \oplus C[m]10^*) \oplus C[m-1]$
    $Z \leftarrow E(L' \oplus M[m-1])$
    $M[m] \leftarrow \mathrm{msb}_{|C[m]|}(Z) \oplus C[m]$
    $\Sigma \leftarrow \Sigma \oplus Z \oplus C[m]10^*$
    $L_{\mathrm{last}} \leftarrow L' \oplus L$
**if** $m$ **is odd**
    $M[m] \leftarrow \mathrm{msb}_{|C[m]|}(E(L')) \oplus C[m]$
    $\Sigma \leftarrow \Sigma \oplus M[m]10^*$
    $L_{\mathrm{last}} \leftarrow L'$
**if** $|C[m]| \neq n$ **then** $TE \leftarrow E(3L_{\mathrm{last}} \oplus \Sigma)$
**else** $TE \leftarrow E(3L_{\mathrm{last}} \oplus L \oplus \Sigma)$
$M \leftarrow M[1] \| M[2] \| \ldots \| M[m]$
**return** $(M, TE)$
```

Algorithm $\mathrm{AF}_E(A)$

```
$\Xi \leftarrow 0^n$
$Q \leftarrow E(0^n)$, $Q' \leftarrow 4Q$
$A[1] \| A[2] \| \ldots \| A[a] \xleftarrow{n} A$
**for** $i = 1$ **to** $a - 1$ **do**
    $\Xi \leftarrow \Xi \oplus E(Q' \oplus A[i])$
    $Q' \leftarrow 2Q'$
$\Xi \leftarrow \Xi \oplus A[a]10^*$
**if** $|A[a]| \neq n$ **then** $TA \leftarrow E(Q' \oplus Q \oplus \Xi)$
**else** $TA \leftarrow E(Q' \oplus 2Q \oplus \Xi)$
**return** $TA$
```

Fig. 1. The encryption and decryption algorithms of OTR with n-bit blockcipher E. Tag size is $0 < \tau \leq n$.

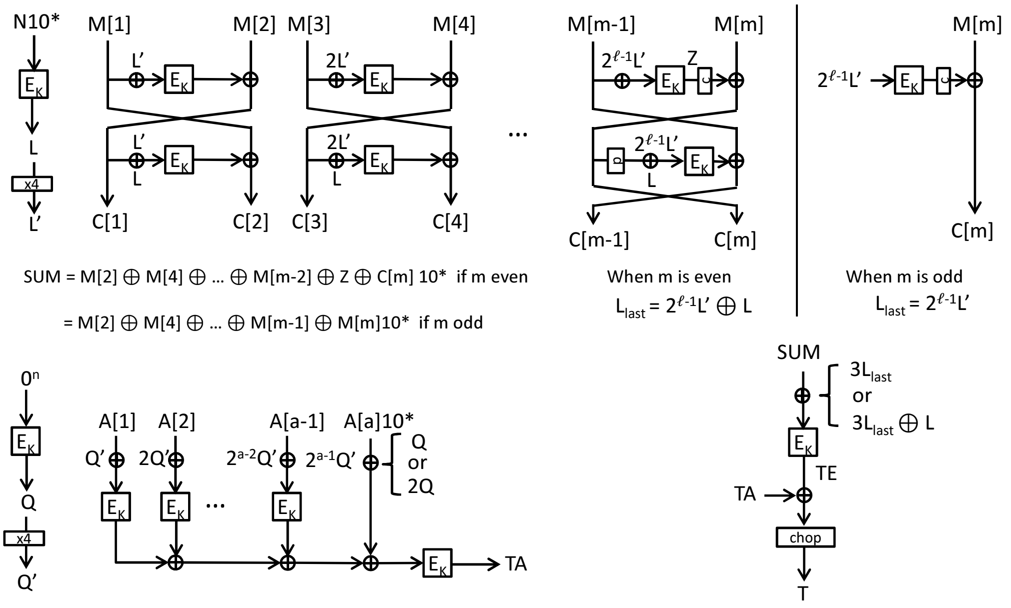

Fig. 2. OTR mode, with Galois field doubling.

We define an AEAD scheme denoted by $\mathbb{OTR}'[\widetilde{\mathsf{R}}, \mathsf{R}^{\infty}, \tau]$. It is compatible to $\mathrm{OTR}[E, \tau]$ and uses a tweakable n-bit URF, $\widetilde{\mathsf{R}} : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$, and an independent VIL-URF, $\mathsf{R}^{\infty} : \{0,1\}^* \to \{0,1\}^n$, Here, tweak $T \in \mathcal{T}$ is represented as a vector, $T = (x, i, \gamma) \in \mathcal{N} \times \mathbb{N} \times \{\mathtt{f}, \mathtt{s}, \mathtt{a}_1, \mathtt{a}_2, \mathtt{b}_1, \mathtt{b}_2\}$. We will augment the domain of γ later. Here $\mathbb{OTR}'[\widetilde{\mathsf{R}}, \mathsf{R}^{\infty}, \tau]$ consists of encryption function, $\mathbb{OTR}'\text{-}\mathcal{E}_{\widetilde{\mathsf{R}}, \mathsf{R}^{\infty}, \tau}$, and decryption function, $\mathbb{OTR}'\text{-}\mathcal{D}_{\widetilde{\mathsf{R}}, \mathsf{R}^{\infty}, \tau}$. The definition of $\mathbb{OTR}'$ is in Fig. 3. Counterparts to EF and DF are denoted by $\mathbb{EF}$ and $\mathbb{DF}$, also shown in Fig. 3.

The bounds of $\mathbb{OTR}'$ are as follows. The proof of Theorem 3 is in Appendix A.

Theorem 3. *Fix $\tau \in \{1, \ldots, n\}$. For any CPA-adversary $\mathcal{A}$,*

$$\mathtt{Adv}^{\mathtt{priv}}_{\mathbb{OTR}'[\widetilde{\mathsf{R}}, \mathsf{R}^{\infty}, \tau]}(\mathcal{A}) = 0.$$

Moreover, for any CCA-adversary $\mathcal{A}$ using q encryption queries and q_v verification queries,

$$\mathtt{Adv}^{\mathtt{auth}}_{\mathbb{OTR}'[\widetilde{\mathsf{R}}, \mathsf{R}^{\infty}, \tau]}(\mathcal{A}) \leq \frac{2q_v}{2^n} + \frac{q_v}{2^{\tau}}.$$

5.2 Tweakable PRFs and Variable-input-length PRFs

In Fig. 4 we define a tweakable URP, $\widetilde{G}[\mathsf{P}]^{\langle N, i, \gamma \rangle}(X)$, where (N, i, γ) denotes a tweak and X denotes an input. It uses an n-bit URP P. We remark that Fig. 4 slightly abuse the notation N to allow 0^n, hence the domain of N is $\mathcal{N}' \overset{\text{def}}{=} \{\{0,1\}^{\leq n-1} \cup \{0^n\}\} \setminus \{\varepsilon\}$. Here $i \in \mathbb{N}$, and γ takes one of 9 values, $\mathcal{W} \overset{\text{def}}{=} \{\mathtt{f}, \mathtt{s}, \mathtt{a}_1, \mathtt{a}_2, \mathtt{b}_1, \mathtt{b}_2, \mathtt{h}, \mathtt{g}_1, \mathtt{g}_2\}$. Note that not all tweaks in $\mathcal{N}' \times \mathbb{N} \times \mathcal{W}$ appear in Fig. 4, say $(0^n, i, \mathtt{f})$. We let them as undefined. It is easy to verify that, if $\mathbb{EF}_{\widetilde{\mathsf{R}}}$ ($\mathbb{DF}_{\widetilde{\mathsf{R}}}$) uses $\widetilde{G}[\mathsf{P}]^{\langle N, i, \gamma \rangle}(*)$ instead of $\widetilde{\mathsf{R}}^{\langle N, i, \gamma \rangle}(*)$, we obtain EF_{P} (DF_{P}). For instance, L_{last} equals to $2^{\ell-1}L' \oplus L$ when message has m blocks and m is even, for $\ell = \lfloor m/2 \rfloor$, thus the last mask is either $3L_{\text{last}}$ (if the last block is shorter than n bits) or $3L_{\text{last}} \oplus L$ (if the last block has n bits), corresponding to the cases $\mathtt{a}_1$ and $\mathtt{a}_2$ in Fig. 4. When m is odd, L_{last} equals to $2^{\ell-1}L'$, and the last mask is either $3L_{\text{last}}$ (case $\mathtt{b}_1$) or $3L_{\text{last}} \oplus L$ (case $\mathtt{b}_2$).

Then we have the following lemma.

Lemma 1. *For any adversary $\mathcal{A}$ accessing $\widetilde{G}[\mathsf{P}]$ with q queries, we have*

$$\mathtt{Adv}^{\mathtt{cpa}}_{\widetilde{G}[\mathsf{P}],\widetilde{\mathsf{R}}}(\mathcal{A}) \le \frac{5q^2}{2^n},$$

where $\widetilde{\mathsf{R}}$ is compatible with $\widetilde{G}[\mathsf{P}]$.

Proof. The crucial observation is that the input mask is differentially uniform. Specifically, we observe that, when $N \ne 0^n$ the set of possible values of Δ shown in Fig. 4 is

$$\begin{aligned}\mathcal{S}_1(L) &\overset{\text{def}}{=} \{2^{i-1}L', 2^{i-1}L' \oplus L, 3(2^{i-1}L' \oplus L), 3(2^{i-1}L \oplus L), 2^{i-1}3L', 2^{i-1}3L' \oplus L\},\\ &= \{2^{i+1}L, 2^{i+1}L \oplus L, 2^{i+2}L \oplus 2^{i+1}L \oplus 2L \oplus L, 2^{i+2}L \oplus 2^{i+1}L \oplus 2L,\\ &\quad 2^{i+2}L \oplus 2^{i+1}L, 2^{i+2}L \oplus 2^{i+1}L \oplus L\},\end{aligned} \tag{4}$$

for $i \ge 1$ and $L' = 4L$, and when $N = 0^n$, the set of possible values of Δ is

$$\begin{aligned}\mathcal{S}_2(Q) &\overset{\text{def}}{=} \{2^{i-1}Q', 2^{i-1}Q' \oplus Q, 2^{i-1}Q' \oplus 2Q\}\\ &= \{2^{i+1}Q, 2^{i+1}Q \oplus Q, 2^{i+1}Q \oplus 2Q\},\end{aligned} \tag{5}$$

for $i \ge 1$ and $Q' = 4Q$. Let $L_1, L_2, Q \xleftarrow{\$} \{0,1\}^n$ be the independent and uniform variables. Then it is easy to see that

$$\max_{\delta \in \{0,1\}^n, X, X' \in \mathcal{S}_1(L_1) \cup \mathcal{S}_1(L_2) \cup \mathcal{S}_2(Q), X \not\equiv X'} \Pr[X \oplus X' = \delta] \le \frac{1}{2^n} \tag{6}$$

holds (here $X \not\equiv X'$ means that X and X' are different in their expressions). By writing Δ of Fig. 4 defined for tweak (N, i, γ) as $\Delta(N, i, \gamma)$, Equation (6) shows that if P is replaced with a URF, R, in Fig. 4, we have

$$\max_{\delta \in \{0,1\}^n} \Pr[\Delta(N, i, \gamma) \oplus \Delta(N', i', \gamma') = \delta] \le \frac{1}{2^n}. \tag{7}$$

for any tweak pairs, $(N, i, \gamma) \ne (N', i', \gamma')$, that are defined in Fig. 4. From Equation (7), we have $\mathtt{Adv}^{\mathtt{cpa}}_{\widetilde{G}[\mathsf{P}],\widetilde{\mathsf{P}}}(\mathcal{A}) \le 4.5q^2/2^n$, where $\widetilde{\mathsf{P}}$ is a tweakable URP compatible with $\widetilde{G}[\mathsf{P}]$, in a similar manner to the security proof of Rogaway's XE construction [26]. A slight generalized form of PRP/PRF switching lemma (e.g., Lemma 1 of [7]) tells that $\mathtt{Adv}^{\mathtt{cpa}}_{\widetilde{\mathsf{P}},\widetilde{\mathsf{R}}}(\mathcal{A}) \le 0.5q^2/2^n$ for q CPA queries, hence the proof is completed as $\mathtt{Adv}^{\mathtt{cpa}}_{\widetilde{G}[\mathsf{P}],\widetilde{\mathsf{R}}}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{cpa}}_{\widetilde{G}[\mathsf{P}],\widetilde{\mathsf{P}}}(\mathcal{A}) + \mathtt{Adv}^{\mathtt{cpa}}_{\widetilde{\mathsf{P}},\widetilde{\mathsf{R}}}(\mathcal{A}) \le 4.5q^2/2^n + 0.5q^2/2^n$. □

Fig. 5 shows a variable-input-length function : $\{0,1\}^* \to \{0,1\}^n$ denoted by $\mathbb{AF}[\widetilde{\mathsf{R}}]$. The internal $\widetilde{\mathsf{R}}$ is a tweakable URF compatible with $\widetilde{G}[\mathsf{P}]$. It is again easy to observe that if $\mathbb{AF}[\widetilde{\mathsf{R}}]$ uses $\widetilde{G}[\mathsf{P}]$ instead of $\widetilde{\mathsf{R}}$, we obtain AF_{P}. We provide the security bound for $\mathbb{AF}[\widetilde{\mathsf{R}}]$, which is as follows.

Lemma 2. *For any adversary $\mathcal{A}$ accessing $\mathbb{AF}[\widetilde{\mathsf{R}}]$ with σ input blocks, we have*

$$\mathtt{Adv}^{\mathtt{prf}}_{\mathbb{AF}[\widetilde{\mathsf{R}}]}(\mathcal{A}) \le \frac{\sigma^2}{2^{n+1}}.$$

The proof of Lemma 2 is almost the same as a part of PMAC proof (the last equation of Appendix E of [28]), thus omitted.

5.3 Deriving PRIV and AUTH

Let $\mathbb{OTR}[\widetilde{\mathsf{R}},\tau]$ be an AEAD consisting of $\mathbb{EF}_{\widetilde{\mathsf{R}}}$, $\mathbb{DF}_{\widetilde{\mathsf{R}}}$, shown in Fig. 3, and $\mathbb{AF}_{\widetilde{\mathsf{R}}}$ shown in Fig. 5. Then, there exist adversaries, $\mathcal{B}$ against $\mathbb{AF}[\widetilde{\mathsf{R}}]$ with σ_A input blocks, and $\mathcal{C}$ against $\widetilde{G}[\mathsf{P}]$ with $\sigma_A + \sigma_M + q$ queries, satisfying

$$\begin{align}
\mathtt{Adv}^{\mathtt{priv}}_{\mathrm{OTR}[\mathsf{P},\tau]}(\mathcal{A}) &= \mathtt{Adv}^{\mathtt{cpa\text{-}nr}}_{\mathrm{OTR}[\mathsf{P},\tau],\$}(\mathcal{A}) \tag{8}\\
&= \mathtt{Adv}^{\mathtt{cpa\text{-}nr}}_{\mathrm{OTR}[\mathsf{P},\tau],\mathbb{OTR}[\widetilde{\mathsf{R}},\tau]}(\mathcal{A}) + \mathtt{Adv}^{\mathtt{cpa\text{-}nr}}_{\mathbb{OTR}[\widetilde{\mathsf{R}},\tau],\mathbb{OTR}'[\widetilde{\mathsf{R}},\mathsf{R}^\infty,\tau]}(\mathcal{A}) + \mathtt{Adv}^{\mathtt{cpa\text{-}nr}}_{\mathbb{OTR}'[\widetilde{\mathsf{R}},\mathsf{R}^\infty,\tau],\$}(\mathcal{A}) \tag{9}\\
&= \mathtt{Adv}^{\mathtt{cpa\text{-}nr}}_{\mathrm{OTR}[\mathsf{P},\tau],\mathbb{OTR}[\widetilde{\mathsf{R}},\tau]}(\mathcal{A}) + \mathtt{Adv}^{\mathtt{cpa}}_{\mathbb{AF}[\widetilde{\mathsf{R}}],\mathsf{R}^\infty}(\mathcal{B}) + \mathtt{Adv}^{\mathtt{cpa\text{-}nr}}_{\mathbb{OTR}'[\widetilde{\mathsf{R}},\mathsf{R}^\infty,\tau],\$}(\mathcal{A}) \tag{10}\\
&\le \mathtt{Adv}^{\mathtt{cpa}}_{\widetilde{G}[\mathsf{P}],\widetilde{\mathsf{R}}}(\mathcal{C}) + \frac{\sigma_A^2}{2^{n+1}} + 0 \text{ (from Lemma 2 and Theorem 3)} \tag{11}\\
&\le \frac{5(\sigma_A + \sigma_M + q)^2}{2^n} + \frac{\sigma_A^2}{2^{n+1}} \text{ (from Lemma 1)} \tag{12}\\
&\le \frac{6\sigma_{\mathtt{priv}}^2}{2^n}. \tag{13}
\end{align}$$

Similarly, for any CCA-adversary $\mathcal{A}$ against $\mathrm{OTR}[\mathsf{P},\tau]$, there exist $\mathcal{B}$ against $\mathbb{AF}[\widetilde{\mathsf{R}}]$ with $\sigma_A + \sigma_{A'}$ input blocks, and $\mathcal{C}$ against $\widetilde{G}[\mathsf{P}]$ with $\sigma_A + \sigma_M + \sigma_{A'} + \sigma_{C'} + q + q_v$ queries, satisfying

$$\begin{align}
\mathtt{Adv}^{\mathtt{auth}}_{\mathrm{OTR}[\mathsf{P},\tau]}(\mathcal{A}) &\le \mathtt{Adv}^{\mathtt{cca\text{-}nr}}_{\mathrm{OTR}[\mathsf{P},\tau],\mathbb{OTR}'[\widetilde{\mathsf{R}},\mathsf{R}^\infty,\tau]}(\mathcal{A}) + \mathtt{Adv}^{\mathtt{auth}}_{\mathbb{OTR}'[\widetilde{\mathsf{R}},\mathsf{R}^\infty,\tau]}(\mathcal{A}) \tag{14}\\
&= \mathtt{Adv}^{\mathtt{cca\text{-}nr}}_{\mathrm{OTR}[\mathsf{P},\tau],\mathbb{OTR}[\widetilde{\mathsf{R}},\tau]}(\mathcal{A}) + \mathtt{Adv}^{\mathtt{cca\text{-}nr}}_{\mathbb{OTR}[\widetilde{\mathsf{R}},\tau],\mathbb{OTR}'[\widetilde{\mathsf{R}},\mathsf{R}^\infty,\tau]}(\mathcal{A}) + \mathtt{Adv}^{\mathtt{auth}}_{\mathbb{OTR}'[\widetilde{\mathsf{R}},\mathsf{R}^\infty,\tau]}(\mathcal{A}) \tag{15}\\
&= \mathtt{Adv}^{\mathtt{cca\text{-}nr}}_{\mathrm{OTR}[\mathsf{P},\tau],\mathbb{OTR}[\widetilde{\mathsf{R}},\tau]}(\mathcal{A}) + \mathtt{Adv}^{\mathtt{cpa}}_{\mathbb{AF}[\widetilde{\mathsf{R}}],\mathsf{R}^\infty}(\mathcal{B}) + \mathtt{Adv}^{\mathtt{auth}}_{\mathbb{OTR}'[\widetilde{\mathsf{R}},\mathsf{R}^\infty,\tau]}(\mathcal{A}) \tag{16}\\
&\le \mathtt{Adv}^{\mathtt{cpa}}_{\widetilde{G}[\mathsf{P}],\widetilde{\mathsf{R}}}(\mathcal{C}) + \frac{(\sigma_A + \sigma_{A'})^2}{2^{n+1}} + \frac{2q_v}{2^n} + \frac{q_v}{2^\tau}(\text{ from Lemma 2 and Theorem 3)} \tag{17}\\
&\le \frac{5(\sigma_A + \sigma_M + \sigma_{A'} + \sigma_{C'} + q + q_v)^2}{2^n} + \frac{(\sigma_A + \sigma_{A'})^2}{2^{n+1}} + \frac{2q_v}{2^n} + \frac{q_v}{2^\tau} \tag{18}\\
&\le \frac{5\sigma_{\mathtt{auth}}^2}{2^n} + \frac{0.5(\sigma_A + \sigma_{A'})^2 + 2\sigma_{A'}}{2^n} + \frac{q_v}{2^\tau} \le \frac{6\sigma_{\mathtt{auth}}^2}{2^n} + \frac{q_v}{2^\tau}. \tag{19}
\end{align}$$

This concludes the proof.

6 Other Instantiations

As the core idea of our proposal is general, it allows various instantiations, by seeing $\mathbb{OTR}$ or $\mathbb{OTR}'$ as a prototype. What we need is just to instantiate $\widetilde{\mathsf{R}}$ accepting n-bit input and tweak (N, i, γ), and producing n-bit output. This paper presented an instantiation with a blockcipher mode using GF doubling-based masking, however, other variants with different masking methods are certainly possible. For example, we can use Gray code [20, 29] or a word-oriented LFSR [10, 20, 31].

Moreover, as we do not need the inversion operation of the underlying E_K, we can use cryptographic primitives other than blockciphers. A typical example is a hash function-based PRF, e.g. HMAC. We can also use components of a hash function, such as a compression function of SHA-2, or a keyless permutation of SHA-3 (Keccak) with Even-Mansour conversion [12] into a keyed permutation. In the latter case the resulting scheme does not need an inversion of the permutation, and there is no output loss like "capacity" bits of SpongeWrap [9].

In these settings, a simple tweaking method by prepending can be an option. As an example we take SipHash [4], which is a variable-input-length (VIL) PRF with 64-bit output. A SipHash-based scheme would be obtained by replacing $\widetilde{\mathsf{R}}^{\langle N,i,\gamma\rangle}(*)$ of $\mathbb{OTR}'$ (Fig. 3) with $\mathrm{SipHash}_K(N\|i\|\gamma\|*)$, and replacing $\mathsf{R}^{\infty}(*)$ with $\mathrm{SipHash}_K(0^n\|0\|\mathtt{h}\|*)$, accompanied with an appropriate input encoding. Fig. 6 depicts the scheme, where γ is encoded as $\gamma = \mathtt{a}_1 \to 0$, $\gamma = \mathtt{a}_2 \to 1$, and so on. As SipHash has an iterative structure, we can efficiently compute $\mathrm{SipHash}_K(N\|i\|\gamma\|X)$ using an intermediate value obtained by the computation of $\mathrm{SipHash}_K(N\|i'\|\gamma'\|X')$, hence we do not need a specific masking function. We remark that this scheme has roughly 64-bit security. The proof is trivial from Theorem 3, combined with the assumption that SipHash is a VIL-PRF.

7 Concluding Remarks

We have presented an authenticated encryption scheme using a PRF. This scheme enables rate-1, on-line, and parallel processing for both encryption and decryption. The core idea of our proposal is to use two-round Feistel with input masking, combined with a message check sum. As a concrete instantiation we provide a blockcipher mode, called OTR, based on a blockcipher encryption function (but not the inverse), which may be seen as an "inverse-free" version of OCB. While the abstract structure has a similarity to OCB mode, our result is not a trivial consequence of previous results. For example if we use 4-round Feistel using an n-bit blockcipher (with input masking) as a $2n$-bit blockcipher and plug it into OCB, the resulting mode is inverse-free and provably secure, since 4-round Feistel is an SPRP, as shown by Luby and Rackoff [21]. However, the rate is degraded to 2, hence no significant gain from the generic composition. As we mentioned in Introduction, our proposal has a higher complexity than OCB outside the blockcipher, hence it is not a good substitute when the blockcipher enc/dec functions are natively supported and equally fast, even though the relaxed security assumption. In contrast, our proposal would be useful for various environments where the use of blockcipher inversion has a non-negligible cost.

Acknowledgments. The author would like to thank Tetsu Iwata for fruitful discussions, and thank Sumio Morioka and Tomoyasu Suzaki for useful comments.

References

1. Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality . NIST Special Publication 800-38C (2004), national Institute of Standards and Technology.
2. 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.
3. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E.W., Yasuda, K.: Parallelizable (authenticated) online ciphers. DIAC 2013: Directions in Authenticated Ciphers (2013), available from `http://2013.diac.cr.yp.to/`
4. Aumasson, J.P., Bernstein, D.J.: SipHash: A Fast Short-Input PRF. In: Galbraith, S.D., Nandi, M. (eds.) INDOCRYPT. Lecture Notes in Computer Science, vol. 7668, pp. 489–508. Springer (2012)
5. Bellare, M., Goldreich, O., Mityagin, A.: The Power of Verification Queries in Message Authentication and Authenticated Encryption. Cryptology ePrint Archive, Report 2004/309 (2004), `http://eprint.iacr.org/`
6. Bellare, M., Namprempre, C.: Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In: Okamoto, T. (ed.) ASIACRYPT. Lecture Notes in Computer Science, vol. 1976, pp. 531–545. Springer (2000)
7. Bellare, M., Rogaway, P.: The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In: Vaudenay, S. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 4004, pp. 409–426. Springer (2006)
8. Bellare, M., Rogaway, P., Wagner, D.: The EAX Mode of Operation. In: Roy and Meier [30], pp. 389–407

9. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. In: Miri, A., Vaudenay, S. (eds.) Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 7118, pp. 320–337. Springer (2011)
10. Chakraborty, D., Sarkar, P.: A general construction of tweakable block ciphers and different modes of operations. IEEE Transactions on Information Theory 54(5), 1991–2006 (2008)
11. Daemen, J., Rijmen, V.: AES Proposal: Rijndael (1999)
12. Even, S., Mansour, Y.: A Construction of a Cipher From a Single Pseudorandom Permutation. In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) ASIACRYPT. Lecture Notes in Computer Science, vol. 739, pp. 210–224. Springer (1991)
13. Fleischmann, E., Forler, C., Lucks, S.: McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. In: Canteaut, A. (ed.) FSE. Lecture Notes in Computer Science, vol. 7549, pp. 196–215. Springer (2012)
14. Gligor, V.D., Donescu, P.: Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes. In: Matsui, M. (ed.) FSE. Lecture Notes in Computer Science, vol. 2355, pp. 92–108. Springer (2001)
15. Gouvêa, C.P.L., López, J.: High Speed Implementation of Authenticated Encryption for the MSP430X Microcontroller. In: Hevia, A., Neven, G. (eds.) LATINCRYPT. Lecture Notes in Computer Science, vol. 7533, pp. 288–304. Springer (2012)
16. Iwata, T., Kurosawa, K.: OMAC: One-Key CBC MAC. In: Johansson, T. (ed.) FSE. Lecture Notes in Computer Science, vol. 2887, pp. 129–153. Springer (2003)
17. Iwata, T., Yasuda, K.: BTM: A Single-Key, Inverse-Cipher-Free Mode for Deterministic Authenticated Encryption. In: Jr., M.J.J., Rijmen, V., Safavi-Naini, R. (eds.) Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 5867, pp. 313–330. Springer (2009)
18. Jutla, C.S.: Encryption Modes with Almost Free Message Integrity. In: Pfitzmann, B. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 2045, pp. 529–544. Springer (2001)
19. Krawczyk, H.: The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?). In: Kilian, J. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 2139, pp. 310–331. Springer (2001)
20. Krovetz, T., Rogaway, P.: The Software Performance of Authenticated-Encryption Modes. In: Joux, A. (ed.) FSE. Lecture Notes in Computer Science, vol. 6733, pp. 306–327. Springer (2011)
21. Luby, M., Rackoff, C.: How to Construct Pseudorandom Permutations from Pseudorandom Functions. SIAM J. Comput. 17(2), 373–386 (1988)
22. Lucks, S.: Two-Pass Authenticated Encryption Faster Than Generic Composition. In: Gilbert, H., Handschuh, H. (eds.) FSE. Lecture Notes in Computer Science, vol. 3557, pp. 284–298. Springer (2005)
23. Osvik, D.A., Bos, J.W., Stefan, D., Canright, D.: Fast Software AES Encryption. In: Hong, S., Iwata, T. (eds.) FSE. Lecture Notes in Computer Science, vol. 6147, pp. 75–93. Springer (2010)
24. Paterson, K.: Authenticated Encryption in TLS. DIAC 2013: Directions in Authenticated Ciphers (2013), available from `http://2013.diac.cr.yp.to/`
25. Rinne, S.: Performance Analysis of Contemporary Light-Weight Cryptographic Algorithms on a Smart Card Microcontroller. SPEED – Software Performance Enhancement for Encryption and Decryption (2007), available from `http://www.hyperelliptic.org/SPEED/start07.html`
26. Rogaway, P.: Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In: Lee, P.J. (ed.) ASIACRYPT. Lecture Notes in Computer Science, vol. 3329, pp. 16–31. Springer (2004)
27. Rogaway, P.: Nonce-Based Symmetric Encryption. In: Roy and Meier [30], pp. 348–359
28. Rogaway, P.: Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. Full version (2013), available from `http://www.cs.ucdavis.edu/~rogaway/papers/`
29. Rogaway, P., Bellare, M., Black, J.: OCB: A block-cipher mode of operation for efficient authenticated encryption. ACM Trans. Inf. Syst. Secur. 6(3), 365–403 (2003)
30. Roy, B.K., Meier, W. (eds.): Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers, Lecture Notes in Computer Science, vol. 3017. Springer (2004)
31. Zeng, G., Han, W., He, K.: High Efficiency Feedback Shift Register: σ-LFSR. Cryptology ePrint Archive, Report 2007/114 (2007), `http://eprint.iacr.org/`
32. Zhang, L., Han, S., Wu, W., Wang, P.: iFeed: the Input-Feed AE Modes. Rump Session of FSE 2013 (2013), slides from `http://fse.2013.rump.cr.yp.to/`

A Proof of Theorem 3

PRIV bound. We observe that the any output block of encryption oracle $\mathbb{OTR}'$-$\mathcal{E}_{\widetilde{\mathsf{R}},\mathsf{R}^\infty,\tau}$, except N, contains an output block of $\widetilde{\mathsf{R}}^{\langle N,i,\gamma\rangle}$, where the tweak (N, i, γ) is uniquely used throughout CPA attack by $\mathcal{A}$. For example any odd ciphertext block contains an output of $\widetilde{\mathsf{R}}^{\langle N,i,\mathtt{f}\rangle}$ for odd i

and any even ciphertext block contains an output of $\widetilde{\mathsf{R}}^{\langle N,i,\mathtt{s}\rangle}$ for even i, and a tag T contains TE, which is an output of $\widetilde{\mathsf{R}}^{\langle N,i,\gamma\rangle}$ for some $\gamma \in \{\mathtt{a}_1, \mathtt{a}_2, \mathtt{b}_1, \mathtt{b}_2\}$ and thus random. Tag T is an XOR of TE and TA and the latter is $\mathsf{R}^{\infty}(A)$ if $A \neq \varepsilon$ and 0^n if $A = \varepsilon$, therefore, T is also independent and random. This implies that the output blocks except the nonce is completely random and independent of the adversary's choice (except the length), thus indistinguishable from those of \$ oracle. PRIV bound is naturally derived from this observation.

AUTH bound. We first consider the case $q_v = 1$. We abbreviate $\mathbb{OTR}'[\widetilde{\mathsf{R}}, \mathsf{R}^{\infty}, \tau]$ as $\mathbb{OTR}'$. Let $\mathcal{A}$ be CCA-adversary against $\mathbb{OTR}'$ with q encryption queries and a verification query. Without loss of generality we can assume $\mathcal{A}$ first performs all encryption queries before the decryption query, which is the best strategy for maximizing the probability of successful forgery.

Following Section 2.4, we denote the i-th encryption query and the answer as (N_i, A_i, M_i) and (C_i, T_i). Here $|M_i| = |C_i|$ and $N_i \neq N_j$ for any $1 \leq i < j \leq q$ from the assumption. Let $M_i[1] \| M_i[2] \| \dots \| M_i[m_i] \xleftarrow{n} M_i$ and $M_i^{\mathsf{c}}[1] \| M_i^{\mathsf{c}}[2] \| \dots \| M_i^{\mathsf{c}}[\ell_i] \xleftarrow{2n} M_i$, where $M_i[j]$ is called a j-th block and $M_i^{\mathsf{c}}[j]$ is called a j-th chunk. Note that $m_i = |M_i|_n$ and $\ell_i = |M_i|_{2n}$ (which equals to $\lfloor m_i/2 \rfloor$). For ciphertext we similarly define $C_i[j]$ and $C_i^{\mathsf{c}}[j]$. The verification query (or forgery attempt) is denoted by (N', A', C', T'). We require $(N', A', C') \neq (N_i, A_i, C_i)$ for all $i = 1, \dots, q$, since forgery attempt with $(N', A', C') = (N_i, A_i, C_i)$ and $T' \neq T_i$ for some i is always rejected.

Let T^* be the true tag value for the forgery attempt. Similarly we define TE^*, TA^* and Σ^* for the corresponding values produced in the decryption of the forgery attempt, which uses (N', A', C'). The forgery attempt is accepted as valid iff $T^* = T'$, where

$$T^* = \mathrm{msb}_\tau(\mathit{TE}^* \oplus \mathit{TA}^*), \text{ and } \mathit{TE}^* = \mathrm{lsb}_n(\mathbb{DF}_{\widetilde{\mathsf{R}}}(N', C')), \text{ and } \mathit{TA}^* = \mathsf{R}^{\infty}(A'), \qquad (20)$$

where $\mathrm{lsb}_n(X)$ denotes the last n bits of X. Let $m' = |C'|_n$ and $\ell' = |C'|_{2n}$. We write $C'[1] \| \dots \| C'[m'] \xleftarrow{n} C'$ and $C'^{\mathsf{c}}[1] \| \dots \| C'^{\mathsf{c}}[\ell'] \xleftarrow{2n} C'$. Note that TE^* is equal to $\widetilde{\mathsf{R}}^{\langle N',\ell',\gamma'\rangle}(\Sigma^*)$, where Σ^* is generated as an internal variable of $\mathbb{DF}_{\widetilde{\mathsf{R}}}(N', C')$ for some $\gamma' \in \{\mathtt{a}_1, \mathtt{a}_2, \mathtt{b}_1, \mathtt{b}_2\}$ uniquely determined by the length of C'. Application of function $\widetilde{\mathsf{R}}^{\langle N',\ell',\gamma'\rangle}$ is called a finalization and the tweak (N', ℓ', γ') is called a finalization tweak.

Let $\mathbf{Z} = \{(N_i, A_i, M_i, C_i, T_i)\}_{i=1,\dots,q}$ be the transcript obtained by encryption queries. Seeing $\mathbf{Z}$ as a random variable, the forgery probability is written as

$$\mathtt{Adv}^{\mathtt{auth}}_{\mathbb{OTR}'}(\mathcal{A}) = \Pr_{\mathcal{A},\mathbb{OTR}'}[T' = T^*] = \sum_{\mathbf{z}} \Pr_{\mathcal{A},\mathbb{OTR}'}[T' = T^* | \mathbf{Z} = \mathbf{z}] \cdot \Pr_{\mathcal{A},\mathbb{OTR}'}[\mathbf{Z} = \mathbf{z}], \qquad (21)$$

where the probability space is defined by the interactive game involving $\mathcal{A}$ and $\mathbb{OTR}'$ (also applies to all probabilities hereafter). In deriving the authenticity bound, we fix adversary $\mathcal{A}$ and define $\mathrm{FP}_{\mathbf{z}}$ as $\Pr[T' = T^* | \mathbf{Z} = \mathbf{z}]$, and bound a maximum of $\mathrm{FP}_{\mathbf{z}}$ for all possible $\mathbf{z}$ with $\mathcal{A}$. This provides the upper bound of $\mathtt{Adv}^{\mathtt{auth}}_{\mathbb{OTR}'}(\mathcal{A})$. Here we can assume $\mathcal{A}$ produces a verification query (N', A', C', T') deterministically from $\mathbf{z}$ so that it maximizes $\mathrm{FP}_{\mathbf{z}}$. Note that, the transcript reveals all the input-output pairs for $\widetilde{\mathsf{R}}$ invoked at encryption queries, except the final one to produce TE_i for all $i \leq q$. Hence (N', A', C', T') can be dependent on these pairs. We perform a case analysis for (N', A', C').

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

The finalization tweak is new, hence the TE^* is independent and uniformly random. Thus $\mathrm{FP}_{\mathbf{z}} \leq 1/2^{\tau}$.

Case 2: $(N', C') = (N_\alpha, C_\alpha)$ **for some** $1 \leq \alpha \leq q$**, and** $A' \neq A_\alpha$**.**

We have $T^* = \mathrm{msb}_\tau(TE_\alpha \oplus TA^*)$. First we observe that, throughout the attack the adversary obtains no knowledge about TA_α for all *non-empty* A_α except trivial collisions, since TA_α is xored with TE_α, which is independent and uniform. Note that for any $i \leq q$ with $A_i = \varepsilon$ we always have $TA_i = 0^n$, and for $A_\alpha \neq \varepsilon$ we have $TA_\alpha = \mathsf{R}^\infty(A_\alpha)$, which is random. If we have $A_\alpha = A_\beta \neq \varepsilon$ the adversary only knows that A_α is random, thus completely unpredictable, and that A_β is equal to A_α. This means that the adversary can not predict TA_α for any non-empty A_α beyond random guess. Based on this observation we do a case analysis.

Case 2-1: $A' \neq^\forall A_i, A' \neq \varepsilon$**.**
We observe that TA^* is random, implying $\mathrm{FP}_\mathbf{z} \leq 1/2^\tau$.

Case 2-2: $A' \neq^\forall A_i, A' = \varepsilon$**.**
We observe that $TA^* = 0^n$ and $T^* = \mathrm{msb}_\tau(TE_\alpha) = \mathrm{msb}_\tau(TA_\alpha \oplus T_\alpha)$ for non-empty A_α. Then TA_α is completely unpredictable to the adversary, and so is T^*. Thus we have $\mathrm{FP}_\mathbf{z} \leq 1/2^\tau$.

Case 2-3: $A' = A_\beta \neq \varepsilon$ **for some** $\beta \neq \alpha$**.**
We observe that $TA^* = TA_\beta$ and $T^* = TA_\beta \oplus TE_\alpha$. As TA_β is completely unpredictable, we have $\mathrm{FP}_\mathbf{z} \leq 1/2^\tau$.

Case 2-4: $A' = A_\beta = \varepsilon$ **for some** $\beta \neq \alpha$**.**
We observe that $TA^* = TA_\beta = 0^n$ and $T^* = TE_\alpha = T_\alpha \oplus TA_\alpha$. As $A_\alpha \neq \varepsilon$ holds TA_α is completely unpredictable, and we have $\mathrm{FP}_\mathbf{z} \leq 1/2^\tau$.

Therefore, for all cases we have $\mathrm{FP}_\mathbf{z} \leq 1/2^\tau$.

Case 3: $N' = N_\alpha$**,** $|C'| = |C_\alpha|$ **and** $C' \neq C_\alpha$ **for some** $1 \leq \alpha \leq q$**.**
Let $(C_\alpha[1] \| \ldots \| C_\alpha[m_\alpha]) \overset{n}{\leftarrow} C_\alpha$ and $(C^\mathsf{c}_\alpha[1] \| \ldots \| C^\mathsf{c}_\alpha[\ell_\alpha]) \overset{2n}{\leftarrow} C_\alpha$, and $(C'[1] \| \ldots \| C'[m']) \overset{n}{\leftarrow} C'$, and $(C'^\mathsf{c}[1] \| \ldots \| C'^\mathsf{c}[\ell']) \overset{2n}{\leftarrow} C'$. Here we have $m' = m_\alpha$ and $\ell' = \ell_\alpha$. Note that we do not pose any condition on A': it can be any value.

Case 3-1: $|C'^\mathsf{c}[\ell']| = 2n$
The finalization tweaks for α-th query and the forgery attempt are the same, i.e. $(N_\alpha, \ell_\alpha, \mathsf{a}_2)$. Then we have $C'^\mathsf{c}[i] \neq C^\mathsf{c}_\alpha[i]$ for some $1 \leq i \leq \ell'$. If $i < \ell_\alpha$, we obtain

$$M^*[2i-1] = \widetilde{\mathsf{R}}^{\langle N', i, \mathsf{s}\rangle}(C'[2i-1]) \oplus C'[2i] \text{ and} \tag{22}$$

$$M^*[2i] = \widetilde{\mathsf{R}}^{\langle N', i, \mathsf{f}\rangle}(M^*[2i-1]) \oplus C'[2i-1] \tag{23}$$

in the decryption process of the forgery attempt. Let e_1 denote the event $M^*[2i-1] = M_\alpha[2i-1]$. If $C'[2i-1] \neq C_\alpha[2i-1]$, e_1 occurs with probability $1/2^n$, and if $C'[2i-1] = C_\alpha[2i-1]$ and $C'[2i] \neq C_\alpha[2i]$, the probability is zero. The event $\overline{e_1}$, i.e. $M^*[2i-1] \neq M_\alpha[2i-1]$, implies that the input to $\widetilde{\mathsf{R}}^{\langle N', i, \mathsf{f}\rangle}$ is new, thus $M^*[2i]$ is uniformly random and independent of any other variables in the transcript. This makes $\varSigma^*$ independent and random under the event $\overline{e_1}$. Let e_2 be the event that $\varSigma^* = \varSigma_\alpha$ (which equals to $M_\alpha[2] \oplus M_\alpha[4] \oplus \ldots \oplus M_\alpha[m_\alpha]$). Then we have

$$\begin{aligned}
\mathrm{FP}_\mathbf{z} &\leq \Pr(\mathrm{msb}_\tau(\widetilde{\mathsf{R}}^{\langle N_\alpha, \ell_\alpha, \mathsf{a}_2\rangle}(\varSigma^*)) = T' | \mathbf{Z} = \mathbf{z}, \overline{e_1 \wedge e_2}) + \Pr(e_2 | \overline{e_1}, \mathbf{Z} = \mathbf{z}) + \Pr(e_1 | \mathbf{Z} = \mathbf{z}) && (24)\\
&\leq \max_{v \in \{0,1\}^\tau} \Pr(\mathrm{msb}_\tau(\widetilde{\mathsf{R}}^{\langle N_\alpha, \ell_\alpha, \mathsf{a}_2\rangle}(\varSigma^*)) = v | \mathbf{Z} = \mathbf{z}, \varSigma^* \neq \varSigma_\alpha) + \frac{1}{2^n} + \frac{1}{2^n} && (25)\\
&\leq \frac{1}{2^\tau} + \frac{2}{2^n}. && (26)
\end{aligned}$$

If $i = \ell_\alpha$, i.e. the difference is in the last chunks, the same analysis holds when we exchange $C'[2i-1]$ and $C'[2i]$. Thus $\mathrm{FP}_\mathbf{z}$ is bounded by $\frac{1}{2^\tau} + \frac{2}{2^n}$ in this case.

Case 3-2: $n < |C'^\mathsf{c}[\ell']| < 2n$**.**
The finalization tweak is $(N_\alpha, \ell_\alpha, \mathsf{a}_1)$, for both α-th encryption query and the forgery attempt. We have $^\exists C'^\mathsf{c}[i] \neq C^\mathsf{c}_\alpha[i]$ for some $1 \leq i \leq \ell'$. If $i < \ell'$ $(= \ell_\alpha)$ the case is the same as Case 3-1. Otherwise we have $C'^\mathsf{c}[j] = C^\mathsf{c}_\alpha[j]$ for all $j = 1, \ldots, \ell' - 1$ and $C'^\mathsf{c}[\ell'] \neq C^\mathsf{c}_\alpha[\ell']$. If we have $C'[m'] \neq C_\alpha[m']$, the event $M^*[m'-1] = M_\alpha[m'-1]$, which we denote by event e_1, has probability $1/2^n$, since $M^*[m'-1] = \widetilde{\mathsf{R}}^{\langle N_\alpha, \ell_\alpha, \mathsf{s}\rangle}(C'[m']10^*) \oplus C'[m'-1]$ and $C'[m']10*$ is a new input to $\widetilde{\mathsf{R}}^{\langle N_\alpha, \ell_\alpha, \mathsf{s}\rangle}$. When $\overline{e_1}$ occurs, $M^*[m'-1]$ is a new input to produce $Z^* = \widetilde{\mathsf{R}}^{\langle N_\alpha, \ell_\alpha, \mathsf{f}\rangle}(M^*[m'-1])$,

which makes Z^* completely random. As $\varSigma^*$ contains $Z^* \oplus C'[m']10^*$, $\varSigma^*$ is also random. If we have $C'[m'] = C_\alpha[m']$ and $C'[m'-1] \neq C_\alpha[m'-1]$, we always have $M^*[m'-1] \neq M_\alpha[m'-1]$, thus $\varSigma^*$ is always random. Therefore, by defining event e_2 as $\varSigma^* = \varSigma_\alpha$, $\mathrm{FP}_{\mathbf{z}}$ is bounded as

$$\begin{aligned}
\mathrm{FP}_{\mathbf{z}} &= \Pr(\mathrm{msb}_\tau(\widetilde{\mathsf{R}}^{\langle N_\alpha,\ell_\alpha,\mathtt{a}_1\rangle}(\varSigma^*)) = \mathrm{msb}_\tau(TA^*) \oplus T' | \overline{e_1 \vee e_2}, \mathbf{Z} = \mathbf{z}) + \Pr(e_2 \vee e_1 | \mathbf{Z} = \mathbf{z}) \\
&\leq \Pr(\mathrm{msb}_\tau(\widetilde{\mathsf{R}}^{\langle N_\alpha,\ell_\alpha,\mathtt{a}_1\rangle}(\varSigma^*)) = \mathrm{msb}_\tau(TA^*) \oplus T' | \overline{e_1} \wedge \overline{e_2}, \mathbf{Z} = \mathbf{z}) \\
&\quad + \Pr(e_2 | \overline{e_1}, \mathbf{Z} = \mathbf{z}) + \Pr(e_1 | \mathbf{Z} = \mathbf{z}) \\
&\leq \max_{v \in \{0,1\}^\tau} \Pr(\mathrm{msb}_\tau(\widetilde{\mathsf{R}}^{\langle N_\alpha,\ell_\alpha,\mathtt{a}_1\rangle}(\varSigma^*)) = v | \varSigma^* \neq \varSigma_\alpha, M^*[m'-1] \neq M_\alpha[m'-1], \mathbf{Z} = \mathbf{z}) + \frac{2}{2^n} \\
&\leq \frac{1}{2^\tau} + \frac{2}{2^n}. \qquad (27)
\end{aligned}$$

Case 3-3: $|C'^{\mathsf{c}}[\ell']| = n$.
The finalization tweak is $(N_\alpha, \ell_\alpha, \mathtt{b}_2)$, for both α-th encryption query and the forgery attempt.

We have ${}^\exists C'^{\mathsf{c}}[i] \neq C^{\mathsf{c}}_\alpha[i]$ for some $1 \leq i \leq \ell'$. If $i < \ell'$ $(= \ell_\alpha)$ the case is the same as Case 3-1. Otherwise we have $C'^{\mathsf{c}}[j] = C^{\mathsf{c}}_\alpha[j]$ for all $j = 1, \ldots, \ell'-1$ and $C'^{\mathsf{c}}[\ell'] \neq C^{\mathsf{c}}_\alpha[\ell']$, which implies $C'[m'] \neq C_\alpha[m']$. Since $M^*[m'] = C'[m'] \oplus Z^*$ with $Z^* = \widetilde{\mathsf{R}}^{\langle N_\alpha,\ell_\alpha,\mathtt{f}\rangle}(0^n)$, and $M_\alpha[m'] = C_\alpha[m'] \oplus Z_\alpha$ with $Z_\alpha = Z^*$, $M^*[m']$ is always different from $M_\alpha[m']$. As variables contained in $\varSigma_\alpha$ and $\varSigma^*$ other than $M_\alpha[m_\alpha]$ and $M^*[m']$ are the same, we always have $\varSigma_\alpha \neq \varSigma^*$. Thus $TE^* = \widetilde{\mathsf{R}}^{\langle N_\alpha,\ell_\alpha,\mathtt{b}_2\rangle}(\varSigma^*)$ is random and independent of TE_α, implying $\mathrm{FP}_{\mathbf{z}} \leq 1/2^\tau$. Thus $\mathrm{FP}_{\mathbf{z}}$ is bounded by $1/2^\tau + 2/2^n$.

Case 3-4: $|C'^{\mathsf{c}}[\ell']| < n$.
The finalization tweak is $(N_\alpha, \ell_\alpha, \mathtt{b}_1)$, for both α-th encryption query and the forgery attempt. The analysis is similar to Case 3-3, and we have $\mathrm{FP}_{\mathbf{z}} \leq 1/2^\tau + 2/2^n$.

Case 4: $N' = N_\alpha$, $|C'| \neq |C_\alpha|$ for some $1 \leq \alpha \leq q$.

Case 4-1: $|C^{\mathsf{c}}_\alpha[\ell_\alpha]| = 2n$.
The finalization tweak for the forgery attempt is $(N_\alpha, \ell', \gamma)$ for $\gamma \in \{\mathtt{a}_1, \mathtt{a}_1, \mathtt{b}_1, \mathtt{b}_2\}$, and that for the α-th encryption query is $(N_\alpha, \ell_\alpha, \mathtt{a}_1)$. Note that ℓ' may or may not equal to ℓ_α. As we have $(\ell_\alpha, \mathtt{a}_1) \neq (\ell', \gamma)$ and N_α is unique, the finalization tweak $(N_\alpha, \ell', \gamma)$ is new, i.e., not invoked in encryption queries. Hence $TE^* = \widetilde{\mathsf{R}}^{\langle N_\alpha,\ell',\gamma\rangle}(\varSigma^*)$ is independent and random irrespective of $\varSigma^*$. This implies $\mathrm{FP}_{\mathbf{z}} \leq 1/2^\tau$.

Case 4-2: $n < |C^{\mathsf{c}}_\alpha[\ell_\alpha]| < 2n$.
The finalization tweak for the forgery attempt is $(N_\alpha, \ell', \gamma)$ for $\gamma \in \{\mathtt{a}_1, \mathtt{a}_2, \mathtt{b}_1, \mathtt{b}_2\}$, and that for the α-th encryption query is $(N_\alpha, \ell_\alpha, \mathtt{a}_1)$. If $(\ell_\alpha, \mathtt{a}_1) \neq (\ell', \gamma)$, we have $\mathrm{FP}_{\mathbf{z}} \leq 1/2^\tau$ as with Case 4-1. If $(\ell_\alpha, \mathtt{a}_1) = (\ell', \gamma)$, then we must have $m_\alpha = m'$ and $|C_\alpha[m_\alpha]| \neq |C'[m']|$. This means that $C_\alpha[m_\alpha]10^* \neq C'[m']10^*$, i.e., the inputs to $\widetilde{\mathsf{R}}^{\langle N_\alpha,\ell_\alpha,\mathtt{s}\rangle}$ are different. Defining two bad events, e_1 and e_2, in the same manner to the previous cases, we have $\mathrm{FP}_{\mathbf{z}} \leq 1/2^\tau + 2/2^n$.

Case 4-3: $|C^{\mathsf{c}}_\alpha[\ell_\alpha]| = n$.
The finalization tweak for the forgery attempt is $(N_\alpha, \ell', \gamma)$ for $\gamma \in \{\mathtt{a}_1, \mathtt{a}_2, \mathtt{b}_1, \mathtt{b}_2\}$, and that for the α-th encryption query is $(N_\alpha, \ell_\alpha, \mathtt{b}_2)$. We have $(\ell_\alpha, \mathtt{b}_2) \neq (\ell', \gamma)$, and thus $\mathrm{FP}_{\mathbf{z}} \leq 1/2^\tau$ holds as Case 4-1.

Case 4-4: $|C^{\mathsf{c}}_\alpha[\ell_\alpha]| < n$.
The finalization tweak for the forgery attempt is $(N_\alpha, \ell', \gamma)$ for $\gamma \in \{\mathtt{a}_1, \mathtt{a}_2, \mathtt{b}_1, \mathtt{b}_2\}$, and that for the α-th encryption query is $(N_\alpha, \ell_\alpha, \mathtt{b}_1)$. If $(\ell_\alpha, \mathtt{b}_1) \neq (\ell', \gamma)$, we have $\mathrm{FP}_{\mathbf{z}} \leq 1/2^\tau$ as Case 4-1, and if $(\ell_\alpha, \mathtt{b}_1) \neq (\ell', \gamma)$ and there exists $C'^{\mathsf{c}}[i] \neq C^{\mathsf{c}}_\alpha[i]$ for some $i < \ell'$, the analysis is the same as Case 3-1 to have $\mathrm{FP}_{\mathbf{z}} \leq 1/2^\tau + 2/2^n$.

If $(\ell_\alpha, \mathtt{b}_1) = (\ell', \gamma)$ and $C'^{\mathsf{c}}[i] = C_\alpha^{\mathsf{c}}[i]$ for all $i < \ell'$, we must have $m_\alpha = m'$ and $|C'[m']|, |C_\alpha[m']| < n$ and $|C_\alpha[m_\alpha]| \neq |C'[m']|$. Then we have $M^*[m']10^* \neq M_\alpha[m_\alpha]10^*$. This implies that the difference $\mathit{\Sigma}^* \oplus \mathit{\Sigma}_\alpha$ is $M^*[m']10^* \oplus M_\alpha[m_\alpha]10^* \neq 0$ with probability 1. Therefore, we have $\mathrm{FP}_{\mathbf{z}} \leq 1/2^\tau$.

Summarizing all cases. In all cases, we have $\mathrm{FP}_{\mathbf{z}} \leq 1/2^\tau + 2/2^n$. From Equation (21) this proves

$$\mathtt{Adv}^{\mathtt{auth}}_{\mathbb{OTR}'}(\mathcal{A}) \leq \sum_{\mathbf{z}} \mathrm{FP}_{\mathbf{z}} \cdot \Pr[\mathbf{Z} = \mathbf{z}] \leq \frac{1}{2^\tau} + \frac{2}{2^n} \tag{28}$$

for $\mathcal{A}$ using one verification query. Combining Equation (28) with the result of Bellare, Goldreich and Mityagin [5], we have $\mathtt{Adv}^{\mathtt{auth}}_{\mathbb{OTR}'}(\mathcal{A}) \leq q_v/2^\tau + 2q_v/2^n$ for any $\mathcal{A}$ using $q_v \geq 1$ verification queries. This completes the derivation of AUTH bound.

```
Algorithm $\mathbb{OTR}'\text{-}\mathcal{E}_{\widetilde{\mathsf{R}},\mathsf{R}^{\infty},\tau}(N, A, M)$

$(C, \mathit{TE}) \leftarrow \mathbb{EF}_{\widetilde{\mathsf{R}}}(N, M)$
if $A \neq \varepsilon$ then $\mathit{TA} \leftarrow \mathsf{R}^{\infty}(A)$
else $\mathit{TA} \leftarrow 0^n$
$T \leftarrow \mathrm{msb}_\tau(\mathit{TE} \oplus \mathit{TA})$
return $(C, T)$
```

```
Algorithm $\mathbb{OTR}'\text{-}\mathcal{D}_{\widetilde{\mathsf{R}},\mathsf{R}^{\infty},\tau}(N, C, A, T)$

$(M, \mathit{TE}) \leftarrow \mathbb{DF}_{\widetilde{\mathsf{R}}}(N, C)$
if $A \neq \varepsilon$ then $\mathit{TA} \leftarrow \mathsf{R}^{\infty}(A)$
else $\mathit{TA} \leftarrow 0^n$
$\widehat{T} \leftarrow \mathrm{msb}_\tau(\mathit{TE} \oplus \mathit{TA})$
if $\widehat{T} = T$ return $M$
else return $\perp$
```

```
Algorithm $\mathbb{OTR}\text{-}\mathcal{E}_{\widetilde{\mathsf{R}},\tau}(N, A, M)$

$(C, \mathit{TE}) \leftarrow \mathbb{EF}_{\widetilde{\mathsf{R}}}(N, M)$
if $A \neq \varepsilon$ then $\mathit{TA} \leftarrow \mathbb{AF}_{\widetilde{\mathsf{R}}}(A)$
else $\mathit{TA} \leftarrow 0^n$
$T \leftarrow \mathrm{msb}_\tau(\mathit{TE} \oplus \mathit{TA})$
return $(C, T)$
```

```
Algorithm $\mathbb{OTR}\text{-}\mathcal{D}_{\widetilde{\mathsf{R}},\tau}(N, C, A, T)$

$(M, \mathit{TE}) \leftarrow \mathbb{DF}_{\widetilde{\mathsf{R}}}(N, C)$
if $A \neq \varepsilon$ then $\mathit{TA} \leftarrow \mathbb{AF}_{\widetilde{\mathsf{R}}}(A)$
else $\mathit{TA} \leftarrow 0^n$
$\widehat{T} \leftarrow \mathrm{msb}_\tau(\mathit{TE} \oplus \mathit{TA})$
if $\widehat{T} = T$ return $M$
else return $\perp$
```

```
Algorithm $\mathbb{EF}_{\widetilde{\mathsf{R}}}(N, M)$

$\Sigma \leftarrow 0^n$
$M[1] \| M[2] \| \ldots \| M[m] \xleftarrow{n} M$
$\ell \leftarrow \lfloor m/2 \rfloor$
for $i = 1$ to $\ell - 1$ do
    $C[2i-1] \leftarrow \widetilde{\mathsf{R}}^{\langle N,i,\mathtt{f}\rangle}(M[2i-1]) \oplus M[2i]$
    $C[2i] \leftarrow \widetilde{\mathsf{R}}^{\langle N,i,\mathtt{s}\rangle}(C[2i-1]) \oplus M[2i-1]$
    $\Sigma \leftarrow \Sigma \oplus M[2i]$
if $m$ is even
    $Z \leftarrow \widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{f}\rangle}(M[m-1])$
    $C[m] \leftarrow \mathrm{msb}_{|M[m]|}(Z) \oplus M[m]$
    $C[m-1] \leftarrow \widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{s}\rangle}(C[m]10^*) \oplus M[m-1]$
    $\Sigma \leftarrow \Sigma \oplus Z \oplus C[m]10^*$
    if $|M[m]| \neq n$ then $\mathit{TE} \leftarrow \widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{a}_1\rangle}(\Sigma)$
    else $\mathit{TE} \leftarrow \widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{a}_2\rangle}(\Sigma)$
if $m$ is odd
    $C[m] \leftarrow \mathrm{msb}_{|M[m]|}(\widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{f}\rangle}(0^n)) \oplus M[m]$
    $\Sigma \leftarrow \Sigma \oplus M[m]10^*$
    if $|M[m]| \neq n$ then $\mathit{TE} \leftarrow \widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{b}_1\rangle}(\Sigma)$
    else $\mathit{TE} \leftarrow \widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{b}_2\rangle}(\Sigma)$
$C \leftarrow C[1] \| C[2] \| \ldots \| C[m]$
return $(C, \mathit{TE})$
```

```
Algorithm $\mathbb{DF}_{\widetilde{\mathsf{R}}}(N, C)$

$\Sigma \leftarrow 0^n$
$C[1] \| C[2] \| \ldots \| C[m] \xleftarrow{n} C$
$\ell \leftarrow \lfloor m/2 \rfloor$
for $i = 1$ to $\ell - 1$ do
    $M[2i-1] \leftarrow \widetilde{\mathsf{R}}^{\langle N,i,\mathtt{s}\rangle}(C[2i-1]) \oplus C[2i]$
    $M[2i] \leftarrow \widetilde{\mathsf{R}}^{\langle N,i,\mathtt{f}\rangle}(M[2i-1]) \oplus C[2i-1]$
    $\Sigma \leftarrow \Sigma \oplus M[2i]$
if $m$ is even
    $M[m-1] \leftarrow \widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{s}\rangle}(C[m]10^*) \oplus C[m-1]$
    $Z \leftarrow \widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{f}\rangle}(M[m-1])$
    $M[m] \leftarrow \mathrm{msb}_{|C[m]|}(Z) \oplus C[m]$
    $\Sigma \leftarrow \Sigma \oplus Z \oplus C[m]10^*$
    if $|M[m]| \neq n$ then $\mathit{TE} \leftarrow \widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{a}_1\rangle}(\Sigma)$
    else $\mathit{TE} \leftarrow \widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{a}_2\rangle}(\Sigma)$
if $m$ is odd
    $M[m] \leftarrow \mathrm{msb}_{|C[m]|}(\widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{f}\rangle}(0^n)) \oplus C[m]$
    $\Sigma \leftarrow \Sigma \oplus M[m]10^*$
    if $|C[m]| \neq n$ then $\mathit{TE} \leftarrow \widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{b}_1\rangle}(\Sigma)$
    else $\mathit{TE} \leftarrow \widetilde{\mathsf{R}}^{\langle N,\ell,\mathtt{b}_2\rangle}(\Sigma)$
$M \leftarrow M[1] \| M[2] \| \ldots \| M[m]$
return $(M, \mathit{TE})$
```

Fig. 3. The encryption and decryption algorithms of $\mathbb{OTR}'[\widetilde{\mathsf{R}}, \mathsf{R}^{\infty}, \tau]$ with a tweakable n-bit URF $\widetilde{\mathsf{R}}$ and a VIL-URF, R^{∞}, denoted by $\mathbb{OTR}'\text{-}\mathcal{E}_{\widetilde{\mathsf{R}},\mathsf{R}^{\infty},\tau}$ and $\mathbb{OTR}'\text{-}\mathcal{D}_{\widetilde{\mathsf{R}},\mathsf{R}^{\infty},\tau}$. Using $\mathbb{AF}_{\widetilde{\mathsf{R}}}$ of Fig. 5 instead of R^{∞} yields the encryption and decryption algorithms of $\mathbb{OTR}[\widetilde{\mathsf{R}}, \tau]$, denoted by $\mathbb{OTR}\text{-}\mathcal{E}_{\widetilde{\mathsf{R}},\tau}$ and $\mathbb{OTR}\text{-}\mathcal{D}_{\widetilde{\mathsf{R}},\tau}$.

```
Algorithm $\widetilde{G}[\mathsf{P}]^{\langle N,i,\gamma\rangle}(X)$

Preprocessing: $Q \leftarrow \mathsf{P}(0^n)$, $Q' \leftarrow 4Q$
if $N \neq 0^n$ then $L \leftarrow \mathsf{P}(N10^*)$, $L' \leftarrow 4L$
    Switch $\gamma$
    Case $\mathtt{f}$ : $\Delta \leftarrow 2^{i-1}L'$
    Case $\mathtt{s}$ : $\Delta \leftarrow 2^{i-1}L' \oplus L$
    Case $\mathtt{a}_1$ : $\Delta \leftarrow 3(2^{i-1}L' \oplus L)$
    Case $\mathtt{a}_2$ : $\Delta \leftarrow 3(2^{i-1}L' \oplus L) \oplus L$
    Case $\mathtt{b}_1$ : $\Delta \leftarrow 2^{i-1}3L'$
    Case $\mathtt{b}_2$ : $\Delta \leftarrow 2^{i-1}3L' \oplus L$
Else Switch $\gamma$
    Case $\mathtt{h}$ : $\Delta \leftarrow 2^{i-1}Q'$
    Case $\mathtt{g}_1$ : $\Delta \leftarrow 2^{i-1}Q' \oplus Q$
    Case $\mathtt{g}_2$ : $\Delta \leftarrow 2^{i-1}Q' \oplus 2Q$
$Y \leftarrow \mathsf{P}(\Delta \oplus X)$
return $Y$
```

Fig. 4. Tweakable URP.

```
Algorithm $\mathbb{AF}_{\widetilde{\mathsf{R}}}(A)$

$\Xi \leftarrow 0^n$
$A[1] \| A[2] \| \ldots \| A[a] \xleftarrow{n} A$
for $i = 1$ to $a - 1$ do
    $\Xi \leftarrow \Xi \oplus \widetilde{\mathsf{R}}^{\langle 0^n, i, \mathtt{h}\rangle}(A[i])$
    $Q' \leftarrow 2Q'$
$\Xi \leftarrow \Xi \oplus A[a]10^*$
if $|A[a]| \neq n$ then $TA \leftarrow \widetilde{\mathsf{R}}^{\langle 0^n, a, \mathtt{g}_1\rangle}(\Xi)$
else $TA \leftarrow \widetilde{\mathsf{R}}^{\langle 0^n, a, \mathtt{g}_2\rangle}(\Xi)$
return $TA$
```

Fig. 5. Authentication function using $\widetilde{\mathsf{R}}$.

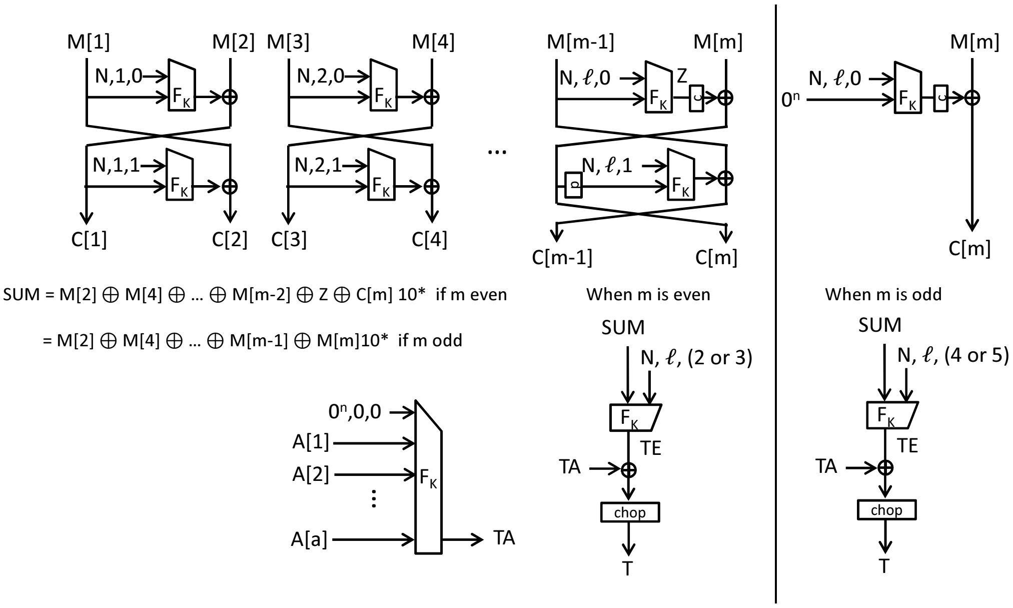

Fig. 6. An instantiation of $\mathbb{OTR}'$ using VIL-PRF.