IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2020, No. 3, pp. 87–118. DOI:10.13154/tosc.v2020.i3.87-118

Fast Decryption: a New Feature of Misuse-Resistant AE

Kazuhiko Minematsu

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

Abstract. Misuse-resistant AE (MRAE) is a class of authenticated encryption (AE) that has a resistance against a potential misuse (repeat) of nonce. MRAE has received significant attention from the initial proposal by Rogaway and Shrimpton. They showed a generic MRAE construction called SIV. SIV becomes a de-facto scheme for MRAE, however, one notable drawback is its two-pass operation for both encryption and decryption. This implies that MRAE built on SIV is slower than the integrated nonce-based AE schemes, such as OCB.
In this paper, we propose a new method to improve this situation. Particularly, our MRAE proposal (decryption-fast SIV or DFV) allows to decrypt as fast as a plain decryption, hence theoretically doubles its speed from the original SIV, while keeping the encryption speed equivalent to SIV. We present several generic compositions for DFV and their instantiations.

Keywords: Authenticated Encryption · Nonce Misuse · MRAE · Decryption · SIV · OCB · Provable Security

1 Introduction

Authenticated encryption (AE) is a symmetric-key cryptographic function for simultaneously providing confidentiality and integrity of plaintexts/ciphertexts. Many popular AE algorithms, such as GCM [MV04], OCB [RBBK01,Rog04,KR11], and ChaCha20-Poly1305 [NL18] are nonce-based AE (NAE), where a nonce is a value that never repeats at encryptions. The security of NAE crucially relies on the uniqueness of nonce. In principle, the nonce uniqueness is easy to maintain, say by using a counter, nonce may repeat in practice due to various reasons, *e.g.*, by misconfiguration of software or low-entropy random source for nonce derivation. The problem of repeating nonce is often called *nonce misuse*. Recently, it received significant attentions from the research community, and it is becoming a real concern as shown by [BZD$^+$16].

Nonce misuse attacks against NAE can be devastating. Most notably, GCM reveals its authentication key only with a single nonce misuse [Jou06], which implies universal forgery attacks. Although these attacks do not invalidate the original security proofs assuming a nonce-respecting adversary, they are extensively studied for various NAE algorithms due to its practical relevance [HP08,PSWZ15,SW14,ADL17]. Vaudenay and Vizár [VV18] showed a thorough study of *robustness* of the 3rd-round candidates of CAESAR competition [CAE14], including the security against nonce misuse.

To overcome this weakness of NAE, Rogaway and Shrimpton [RS06] introduced the notion of Misuse-resistant AE (MRAE) and proposed a generic MRAE scheme called SIV. Specifically, to encrypt a plaintext M with an associated data (AD) A, SIV first derives a initialization vector (IV), T_{iv}, by applying a pseudorandom function (PRF) to the tuple (A, M), and encrypts M by an IV-based encryption scheme taking T_{iv} as IV. To decrypt,

Received: 2020-03-02 Revised: 2020-06-01 Accepted: 2020-08-01 Published: 2020-09-28

SIV first decrypts the ciphertext C with IV T_{iv}, and checks if the received IV is identical to the one computed from the decrypted plaintext and the received AD. See Figure 1 on page 89. Here, A may contain a nonce N but N is not necessarily unique for each encryption. This structure achieves the best-possible security against nonce misuse, namely the confidentiality up to the whole input repetition and the unforgeability [RS06]. Since the proposal, SIV has been extensively studied and it has many concrete instantiations and variants. For example, Deoxys-II [JNPS14], one of the winners of CAESAR competition, adopts a variant of SIV called SCT proposed by Peyrin and Seurin [PS16]. SCT is a TBC-based design, and Iwata *et al.* [IMPS17a] proposed another TBC-based one, ZAE, that has a higher efficiency and security than SCT. GCM-SIV and AES-GCM-SIV are MRAE schemes reusing the components of GCM proposed by Gueron and Lindell [GL15] and Gueron *et al.* [GLL19]. RIV [AFL$^+$16] is a security-enhanced variant of SIV. NIST Lightweight Cryptography, which is a national standardization project for lightweight AE, received 57 submissions [NIS19]. The current second-round candidates include multiple MRAE schemes, ESTATE [CDJ$^+$19], SUNDAE-GIFT [BBP$^+$19, BBLT18], and Romulus-M [IKMP20].

SIV offers a strong defense in depth. That is, it hides the plaintext up to a repetition of whole input and protects the integrity of the ciphertext, even if nonce repeats. However, it comes with a drawback in its computation because it needs two passes over the input for both encryption and decryption. This implies roughly twice more computation than the efficient one-pass, rate-1[1] NAE schemes, such as OCB, whose complexity is almost as small as plain, unauthenticated encryption schemes (*e.g.*, counter mode). The block cipher-based instantiation of SIV shown in [RS06] is indeed rate-1/2. Known variants of SIV also share this property, that is, it is not possible to achieve rate 1[2]. This increased computation has been considered as the price we have to pay for MRAE.

In this paper, we propose a way to improve the situation by reducing the computation cost for decryption. More precisely, our core proposal is a new generic scheme of MRAE whose decryption is one-pass. The encryption is two-pass as for SIV, which is inevitable, because the security requirements of MRAE require that any ciphertext bit must depend on the whole input. To our knowledge, all previous MRAE schemes are variants of SIV or encode-then-encipher scheme [BR00] (such as AEZ [HKR15]), which is even costlier than SIV. Our proposal is the first kind of MRAE that has a smaller decryption cost than SIV. Essentially it allows to reduce the decryption cost to that of an unauthenticated decryption, thus rate-1 decryption. It is observed that the two-pass encryption structure is unavoidable for any MRAE, since every ciphertext bit must depend on the whole input (A, M) to ensure confidentiality. This implies that, our proposal achieves the *best-possible total efficiency* of MRAE in the sense that its computation cost cannot be substantially improved for both directions. Moreover, fast decryption is a desirable feature in practice. It is possible to think of applications that require a fast operation for (authenticated) decryption but not for encryption, due to the asymmetry of the protocol or the computing devices between the sender and the receiver. Storage encryption is one example, where encryption can be done in the background, however decryption should be close to real-time as it affects the latency of read operations.

Our idea is basically simple. While SIV composes a PRF and an IV-based encryption, we compose a PRF and an NAE. The nonce (or random IV[3]) for NAE, V, is derived from the input tuple (A, M) for an AD A and a plaintext M using one or two calls of the PRF. The NAE taking the nonce V encrypts M to generate a ciphertext C and a tag T. This process is similar to the IV-derivation of SIV, however, since we use an NAE, the output of encryption process is a tuple (V, A, C, T) instead of (T_{iv}, A, C) as SIV did. The bandwidth

[1] Rate is the number of input blocks per one primitive call. See Section 2 for details.

[2] The maximum achievable rate of SIV and its variants may be larger than 1/2 as it depends on the input size of the primitive, as shown by Table 1.

[3] We interchangeably call V a nonce or a random IV, assuming its generation process is clear from the context and thus no confusion is possible.

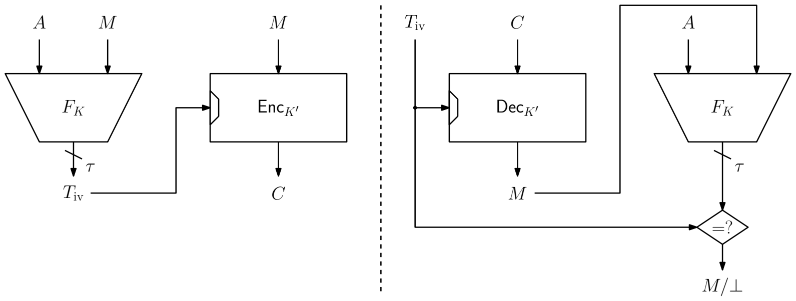

Figure 1: SIV.

overhead will be increased from $|T_{\mathrm{iv}}|$ to $|V| + |T|$, but it is basically the same as that of NAE. The decryption is nothing but a decryption of the NAE scheme we use. Assuming this NAE scheme is efficient (one-pass and rate-1), the decryption is as efficient as a plain, unauthenticated decryption. We name this scheme *decryption-fast* SIV, or DFV.

Although the above idea may sound trivial, the realization of it needs some non-trivial considerations. In particular, a straightforward approach incurs an increased computation for encryption, and a naive attempt to remove it can lead to insecure schemes (See Section 5).

We show three generic compositions for DFV, called DFV1, DFV2 and DFV3, and prove their security bounds that are comparable to the original SIV. It turns out that building an efficient DFV is closely related to the conversion of an AD-less NAE to an NAE with AD (NAEAD), and we adopt the known generic conversions proposed by Rogaway [Rog02]. To demonstrate the effectiveness of DFV, we also present two concrete instantiations of DFV. The first one (OCB-DFV) is block cipher-based and uses OCB and PMAC [Rog04]. The second one (ΘCB-DFV) is based on a tweakable block cipher [LRW02] and uses ΘCB3 [KR11] and ZMAC [IMPS17a]. The former gives the classical up-to-birthday-bound (upBB) security as for (the original block cipher-based instantiation of) SIV, while the latter gives a higher security, namely beyond-the-birthday-bound (BBB) security. See Table 1 for the comparison with some known MRAE schemes. We also provide a brief consideration on permutation-based instantiations at Section 6.

A possible high-level question is when our proposals improve on SIV. Assuming the use of identical primitive, DFV improves SIV, in the sense that it enables a higher decryption rate while keeping the encryption rate unchanged, when the underlying NAE for DFV is rate-1 for both encryption and decryption. This holds for DFV2 and DFV3, though DFV1 has some limitations. See Appendix A for a more detailed comparison.

Related Work on Generic Composition. Generic composition was studied by the seminal work of Bellare and Namprempre [BN00] and Krawczyk [Kra01]. Namprempre *et al.* [NRS14] extended [BN00] and showed a number of compositions for both NAE and MRAE, where the latter generalizes SIV. Sarkar [Sar14] studied compositions using a stream cipher. Berti *et al.* [BPP18] extended [NRS14] and Imamura *et al.* [IMI16] showed a refined analysis of [Sar14]. The problem of converting an AD-less NAE into an NAEAD was first studied by Rogaway [Rog02]. A variant was proposed by Sarkar [Sar10].

2 Preliminaries

Let $\{0,1\}^*$ be the set of all finite bit strings. For $X \in \{0,1\}^*$, $|X|$ is its length in bits. The empty string is denoted by ε and $|\varepsilon| = 0$. For an integer $i \geq 0$, $\{0,1\}^i$ is the set of all bit strings of i bits, and $\{0,1\}^{\leq i}$ is the set of all bit strings of at most i bits, including ε.

Table 1: Comparison of MRAE schemes. BC(n) (TBC(t,n)) denotes n-bit block cipher (t-bit effective tweak and n-bit block TBC). **ERate** (**DRate**) denotes the encryption (decryption) rate for messages. **Bandwidth OH** denotes the overhead of bandwidth in bits. **Security** denotes the MRAE advantage in bits. For OCB-DFV we assume $n/2$-bit tag.

Scheme	Primitive	ERate	DRate	Security	Bandwidth OH	Ref
SIV	BC(n)	1/2	1/2	$n/2$	n	[RS06]
SUNDAE	BC(n)	1/2	1/2	$n/2$	n	[BBLT18]
OCB-DFV	BC(n)	1/2	1	$n/2$	$1.5n$	This work
SCT	TBC(n,n)	1/2	1/2	$n/2$	n	[PS16]
ZAE	TBC(n,n)	2/3	2/3	n	$2n$	[IMPS17a]
Romulus-M	TBC($2n,n$)	2/3	2/3	$n/2$	n	[IKMP20]
ΘCB-DFV	TBC(n,n)	2/3	1	n	$3n$	This work

For an integer $\ell \geq 1$, $|X|_\ell$ is the length of $X \in \{0,1\}^*$ in ℓ-bit blocks, which is defined as $|X|_\ell = \lceil |X|/\ell \rceil$ if $X \neq \varepsilon$, and $|X|_\ell = 1$ if $X = \varepsilon$. For two bit strings X and Y, $X \,\|\, Y$ is their concatenation. We also write this as XY if it is clear from the context. Let 0^i be the string of i zero bits, and for instance we write 10^i for $1 \,\|\, 0^i$. For $X \in \{0,1\}^*$ with $|X| \geq i$, $\mathtt{msb}_i(X)$ is the first (left) i bits of X, and $\mathtt{lsb}_i(X)$ is the last (right) i bits of X. If X is uniformly chosen from the set $\mathcal{X}$, we write $X \overset{\$}{\leftarrow} \mathcal{X}$.

For any $x \in \{0,1\}^{\leq n}$, $\mathtt{pad}(x)$ denotes a so-called one-zero (non-injective) padding: $\mathtt{pad}(x) = x \| 10^{n-|x|-1}$ when $|x| < n$ and $\mathtt{pad}(x) = x$ when $|x| = n$. We also define a simple, non-injective zero padding $\mathtt{pad}_0(*)$ as $\mathtt{pad}_0(x) = x \,\|\, 0^{n-|x|}$. For a positive integer i, let $[i] \coloneqq \{1,2,\ldots,i\}$ and $[\![i]\!] \coloneqq \{0,1,\ldots,i\}$.

In the pseudocodes, [X **if** E **else** Y] is a shorthand for [**if** E **then** X **else** Y].

2.1 Cryptographic Primitives

(Tweakable) Block Ciphers. A tweakable block cipher (TBC) [LRW02] is a keyed function $\widetilde{E} : \mathcal{K} \times \mathcal{TW} \times \mathcal{M} \to \mathcal{M}$ such that for each $(K,T) \in \mathcal{K} \times \mathcal{TW}$, $\widetilde{E}(K,T,\cdot)$ is a permutation over $\mathcal{M}$. Here, K is a key and T is a public value called tweak. The encryption of a plaintext $M \in \mathcal{M}$ with a key $K \in \mathcal{K}$ and a tweak $T \in \mathcal{TW}$ is a ciphertext $C = \widetilde{E}(K,T,M)$. It is also written as $\widetilde{E}_K(T,X)$ or $\widetilde{E}_K^T(X)$. Similarly, the decryption is written as $M = \widetilde{E}^{-1}(K,T,C)$ or $\widetilde{E}_K^{-1}(T,C)$ or $(\widetilde{E}_K^T)^{-1}(C)$. When $\mathcal{TW}$ is written as $\mathcal{TW}' \times \mathcal{D}$ for a set of integers $\mathcal{D}$, we call $\mathcal{TW}'$ the *effective* tweak space of $\widetilde{E}$. The set $\mathcal{D}$ is typically for domain separation, that is, a small set to generate a number of distinct TBC instances. Note that a conventional block cipher $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ is equivalent to a TBC with $|\mathcal{TW}| = 1$. We write $E_K^{-1}(*)$ to denote the decryption function.

Random Primitives. Let $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{TW}$ be non-empty finite sets. Let $\mathrm{Func}(\mathcal{X},\mathcal{Y})$ be the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$, and let $\mathrm{Perm}(\mathcal{X})$ be the set of all permutations over $\mathcal{X}$. Moreover, let $\mathrm{Perm}(\mathcal{TW},\mathcal{X})$ be the set of all functions $f : \mathcal{TW} \times \mathcal{X} \to \mathcal{X}$ such that for any $T \in \mathcal{TW}$, $f(T,\cdot)$ is a permutation over $\mathcal{X}$.

A uniform random function (URF) with a domain $\mathcal{X}$ and a range $\mathcal{Y}$, $\mathsf{R} : \mathcal{X} \to \mathcal{Y}$, is a random function with uniform distribution over $\mathrm{Func}(\mathcal{X},\mathcal{Y})$. Similarly, a uniform random permutation (URP) over $\mathcal{X}$, $\mathsf{P} : \mathcal{X} \to \mathcal{X}$, is a random permutation with uniform distribution over $\mathrm{Perm}(\mathcal{X})$. An n-bit URP is a URP over $\{0,1\}^n$. A tweakable URP (TURP) with a tweak space $\mathcal{TW}$ and a message space $\mathcal{X}$, $\widetilde{\mathsf{P}} : \mathcal{TW} \times \mathcal{X} \to \mathcal{X}$, is a random tweakable permutation with uniform distribution over $\mathrm{Perm}(\mathcal{TW},\mathcal{X})$. The decryption is written as $\mathsf{P}^{-1}(*)$ for URP and $(\widetilde{\mathsf{P}}^{-1})^T(*)$ for TURP given tweak T.

Efficiency Measure. Rate is a popular efficiency measure for (tweakable) block cipher modes. It is the number of input message blocks per cryptographic primitive call, assuming the message block size and the input and output sizes of the primitive are fixed and understood. Throughout the paper, we use n to denote the input block size and the block size of a block cipher. In case of tweakable block cipher, we assume the input block size is n and the tweak size is a function of n, say $2n$. Typically, constant primitive calls that are needed for any inputs, irrespective of the possibility of pre-computation, are ignored. Therefore, when a mode with an n-bit block cipher needs $x\ell + c$ block cipher calls for an ℓ-block ($n\ell$-bit) message for some non-negative constant c, its rate is said to be $1/x$. For example, (any version of) OCB has rate 1 for both encryption and decryption.

We stress that, although it is convenient, the notion of rate is not universal. First of all, comparing rates of different modes using different primitives is basically pointless. Moreover, there is no ultimate consensus when a mode uses multiple primitives, *e.g.*, when an n-bit block cipher and an n-bit full multiplier over $\mathrm{GF}(2^n)$ are combined (say for GCM), or when a reduced-round block cipher is used together with a full-round one, *etc.* None of the schemes discussed in this paper will deal with such cases.

2.2 Authenticated Encryption

We describe the syntax of NAE and MRAE schemes.

NAE. Let $\mathsf{NAE} = (\mathsf{NAE.Enc}, \mathsf{NAE.Dec})$ be an NAE scheme. The (deterministic) encryption algorithm $\mathsf{NAE.Enc}$ takes a key $K \in \mathcal{K}$ and a tuple (N, A, M) of a nonce $N \in \mathcal{N}$, an AD $A \in \mathcal{A}$, and a plaintext $M \in \mathcal{M}$ as input, and returns a ciphertext $C \in \mathcal{M}$ and a tag $T \in \mathcal{T}$. Typically, $\mathcal{M} = \{0,1\}^*$ and $\mathcal{T} = \{0,1\}^\tau$ for a fixed, small τ. The tuple (N, A, C, T) will be sent to the receiver. The (deterministic) decryption algorithm $\mathsf{NAE.Dec}$ takes $K \in \mathcal{K}$ and the tuple (N, A, C, T) as input, and returns $M \in \mathcal{M}$ or the reject symbol $\perp$.

Our definition of NAE covers both cases of $\mathcal{A} \neq \emptyset$ and $\mathcal{A} = \emptyset$, where the former is often called AE with AD (AEAD) [Rog02]. To avoid confusion, if we explicitly mean an NAE scheme without AD ($\mathcal{A} = \emptyset$), we will call it a *plain* NAE (pNAE). For a pNAE scheme, we omit the notation of $\mathcal{A}$ from its syntax. Note that an pNAE is trivially derived from any non-plain NAE by dropping AD from the syntax.

MRAE. Let $\mathsf{MRAE} = (\mathsf{MRAE.Enc}, \mathsf{MRAE.Dec})$ be an MRAE scheme. The encryption $\mathsf{MRAE.Enc}$ takes key $K \in \mathcal{K}$, AD $A \in \mathcal{A}$, and plaintext $M \in \mathcal{M}$ and outputs a ciphertext $C \in \mathcal{M}$ and a tag $T \in \mathcal{T}$. Nonce N is absent (in which case it is also called a deterministic AE or DAE) or exists but may repeat at encryption. For the latter case, N is typically assumed to be a part of AD A. The tuple (A, C, T) will be sent to the receiver. The decryption $\mathsf{MRAE.Dec}$ takes K and the tuple (A, C, T) and returns a plaintext M or the reject symbol $\perp$.

Unfinished Decryption. In our second scheme, we assume that the decryption of the underlying pNAE scheme can be decomposed into two routines: *unfinished decryption* $\mathsf{UDec} : \mathcal{K} \times \mathcal{N} \times \mathcal{M} \to \mathcal{M} \times \mathcal{T}$ and comparison $\mathsf{Cmp} : \mathcal{M} \times \mathcal{T} \times \mathcal{T} \to \mathcal{M} \cup \{\perp\}$, where the latter is defined as $\mathsf{Cmp}(M, T, T') = M$ iff $T = T'$ and otherwise $\perp$. On receiving a tuple $(N, C, T) \in \mathcal{N} \times \mathcal{M} \times \mathcal{T}$, the pNAE decryption first performs $(M', T') \leftarrow \mathsf{UDec}_K(N, C)$ and the final output is $\mathsf{Cmp}(M', T', T)$. Here, M' in the above corresponds to the *unverified plaintext* introduced by Andreeva *et al.*. [ABL+14a], and T' is the tag value that is locally computed. Note that UDec itself is not an *unverified decryption* routine, which returns M' for input (N, C), although it is simply derived by discarding the second output argument of UDec. In any case, our assumption holds for many known (p)NAE schemes, such as

GCM. Some exceptions exist, for example an encode-then-encipher scheme [BR00, HKR15], since its verification is different – it compares a part of large block cipher's decryption result with a fixed value.

2.3 Security Notions

Let A be an adversary that queries an oracle $\mathcal{O}$. We say A is a distinguisher if it outputs $x \in \{0,1\}$ as a final outcome. If the final outcome is 1, we write $\mathsf{A}^{\mathcal{O}} = 1$ to denote this event. It is a probabilistic event whose randomness comes from those of A and $\mathcal{O}$. Queries of A may be adaptive unless otherwise specified. If there are multiple oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$, $\mathsf{A}^{\mathcal{O}_1, \mathcal{O}_2, \ldots}$ means that A can query any oracle in an arbitrary order.

Let $\mathcal{O}$ and $\mathcal{O}'$ be the oracles. For an adversary A who is a distinguisher for $\mathcal{O}$ and $\mathcal{O}'$ using adaptive queries, we define the indistinguishability as

$$\mathbf{Adv}^{\mathtt{ind}}_{\mathcal{O},\mathcal{O}'}(\mathsf{A}) := |\Pr[\mathsf{A}^{\mathcal{O}} = 1] - \Pr[\mathsf{A}^{\mathcal{O}'} = 1]|.$$

For two tuples of oracles, $\mathbf{O} = (\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_s)$ and $\mathbf{O}' = (\mathcal{O}'_1, \mathcal{O}'_2, \ldots, \mathcal{O}'_s)$, $\mathbf{Adv}^{\mathtt{ind}}_{\mathbf{O},\mathbf{O}'}(\mathsf{A})$ is defined as $|\Pr[\mathsf{A}^{\mathcal{O}_1,\mathcal{O}_2,\ldots,\mathcal{O}_s} = 1] - \Pr[\mathsf{A}^{\mathcal{O}'_1,\mathcal{O}'_2,\ldots,\mathcal{O}'_s} = 1]|$. Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a keyed function. The PRF-advantage of F_K against A is defined as

$$\mathbf{Adv}^{\mathtt{prf}}_{F}(\mathsf{A}) := \mathbf{Adv}^{\mathtt{ind}}_{F,\mathsf{R}_F}(\mathsf{A}) = |\Pr[K \xleftarrow{\$} \mathcal{K} : \mathsf{A}^{F_K} = 1] - \Pr[\mathsf{A}^{\mathsf{R}_F} = 1]|, \qquad (1)$$

where $\mathsf{R}_F : \mathcal{X} \to \mathcal{Y}$ is a URF. If (1) is negligibly small for any A of a practical amount of complexity, we say F_K is a pseudorandom function (PRF). The domain of F_K may consist of multiple sets, such as $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ for $\mathcal{X}_i = \{0,1\}^*$ for $i = 1,2$, where we assume $(X_1, X_2) \in \mathcal{X}$ and $(X'_1, X'_2) \in \mathcal{X}$ are different inputs to F_K iff $(X_1, X_2) \neq (X'_1, X'_2)$ (even if $X_1 \,\|\, X_2 = X'_1 \,\|\, X'_2$ holds). In this case, F_K is called a *vector-input* PRF.

For any NAE NAE_K with a nonce $N \in \mathcal{N}$, let $\mathsf{NAE}^{\$}_K$ be the random IV-based counterpart. That is, N is chosen randomly by the encryption oracle instead of the value chosen by the adversary. Specifically, $\mathsf{NAE}^{\$}.\mathsf{Enc}_K$ is queried with a plaintext $M \in \mathcal{M}$ and returns (N, C, T), where $N \xleftarrow{\$} \mathcal{N}$ and $(C, T) \leftarrow \mathsf{NAE}.\mathsf{Enc}_K(N, M)$. The decryption $\mathsf{NAE}^{\$}.\mathsf{Dec}_K$ is identical to $\mathsf{NAE}.\mathsf{Dec}_K$. The same applies to pNAE schemes.

We define the following security notions for NAE and MRAE. They are all-in-one security notions that capture both privacy (confidentiality) and authenticity (integrity).

Definition 1. Let $\mathsf{MRAE} = (\mathsf{MRAE.Enc}, \mathsf{MRAE.Dec})$ and $\mathsf{NAE} = (\mathsf{NAE.Enc}, \mathsf{NAE.Dec})$ be an MRAE scheme and an NAE scheme respectively. Suppose both have a τ-bit tag, and NAE_K has an n-bit nonce. We define

$$\mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{MRAE}}(\mathsf{A}_1) := |\Pr[K \xleftarrow{\$} \mathcal{K} : \mathsf{A}_1^{\mathsf{MRAE.Enc}_K, \mathsf{MRAE.Dec}_K} = 1] - \Pr[\mathsf{A}_1^{\$,\perp} = 1]|, \qquad (2)$$

$$\mathbf{Adv}^{\mathtt{nae}}_{\mathsf{NAE}}(\mathsf{A}_2) := |\Pr[K \xleftarrow{\$} \mathcal{K} : \mathsf{A}_2^{\mathsf{NAE.Enc}_K, \mathsf{NAE.Dec}_K} = 1] - \Pr[\mathsf{A}_2^{\$,\perp} = 1]|, \qquad (3)$$

$$\mathbf{Adv}^{\mathtt{nae\$}}_{\mathsf{NAE}}(\mathsf{A}_3) := |\Pr[K \xleftarrow{\$} \mathcal{K} : \mathsf{A}_3^{\mathsf{NAE}^{\$}.\mathsf{Enc}_K, \mathsf{NAE}^{\$}.\mathsf{Dec}_K} = 1] - \Pr[\mathsf{A}_3^{\$,\perp} = 1]|, \qquad (4)$$

where \$ (for (2) and (3)) is a random-bit oracle that returns a uniformly random string of $|M| + \tau$ bits for any query containing a plaintext M. If MRAE is written as $\mathsf{MRAE}[F, G]$, that means F and G are the components, we assume their key spaces are understood and independently randomly sampled in the game. For (4), the \$ oracle returns $|M| + n + \tau$ random bits for any query (A, M). We require A_1 not to repeat encryption queries, and A_2 to be nonce-respecting, *i.e.*, using unique nonce for each encryption, however no requirement for those used by decryption. For all notions, any decryption query must be non-trivial, *e.g.*, for (3), a decryption query (N, A, C, T) is not allowed when A_2 previously issued an encryption query (N, A, M) and received (C, T). Similar conditions apply for (2) and (4).

For an adversary A against NAE/MRAE, let q_e denote the number of encryption queries, q_d denote the number of decryption queries. When an i-th encryption query (decryption query) is denoted by (N_i, A_i, M_i) $((N'_i, A'_i, C'_i, T'_i))$, we define $\sigma_e := \sum_{i\in[q_e]} |A_i|_n + |M_i|_n$ and $\sigma_d := \sum_{j\in[q_d]} |A'_j|_n + |C'_j|_n$, and $\sigma := \sigma_e + \sigma_d$. The time complexity of A is denoted by t.

For a list of adversarial parameters θ and a security notion $\mathtt{sec}$, we write θ-SEC adversary to mean an adversary that can be an argument of $\mathbf{Adv}^{\mathtt{sec}}_{\Pi}(*)$ for any scheme Π compliant to the security notion. In addition, we may write $\mathbf{Adv}^{\mathtt{sec}}_{\Pi}(\theta)$ to mean $\max_{\mathsf{A}} \mathbf{Adv}^{\mathtt{sec}}_{\Pi}(\mathsf{A})$ where the maximum is taken for all θ-sec adversaries. To avoid confusion, we typically do this for the case that θ is a singleton (say the number of queries) and does not contain the time complexity, which is considered as infinity. For time complexity t and data complexity σ, let $\tilde{t}(\sigma)$ be the shorthand for $t + O(\sigma)$.

Privacy and Authenticity Notions. In our analysis, we also need the following standard individual notions for privacy and authenticity [BN00].

Definition 2. Let $\mathsf{NAE} = (\mathsf{NAE.Enc}, \mathsf{NAE.Dec})$ be an NAE scheme.

$$\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{NAE}}(\mathsf{A}_1) := |\Pr[K \xleftarrow{\$} \mathcal{K} : \mathsf{A}_1^{\mathsf{NAE.Enc}_K} = 1] - \Pr[\mathsf{A}_1^{\$} = 1]|,$$
$$\mathbf{Adv}^{\mathtt{auth}}_{\mathsf{NAE}}(\mathsf{A}_2) := |\Pr[K \xleftarrow{\$} \mathcal{K} : \mathsf{A}_1^{\mathsf{NAE.Enc}_K, \mathsf{NAE.Dec}_K} \text{ forges }],$$

where \$ oracle is as defined as Definition 1. The adversary in the privacy notion is nonce-respecting, and $[\mathsf{A}_2^{\mathsf{NAE.Enc}_K, \mathsf{NAE.Dec}_K} \text{ forges }]$ means that A_2 receives $M \neq \perp$ from $\mathsf{NAE.Dec}_K$ by a non-trivial query (N, A, C, T).

When the authenticity adversary A has single decryption query, we may use $\mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\mathsf{NAE}}(\mathsf{A})$ instead of $\mathbf{Adv}^{\mathtt{auth}}_{\mathsf{NAE}}(\mathsf{A})$ to emphasize that the number of decryption query is one. A (q_e, q_d, σ, t)-AUTH adversary uses q_e encryption queries and q_d decryption queries, with σ total blocks for all queries, and time complexity t. A (q_e, σ, t)-AUTH-1 adversary is defined similarly, with single decryption query.

Proposition 1. *Let NAE be an NAE scheme of ν-bit nonce. For any (q_e, q_v, σ, t)-NAE\$ adversary A, we have $\mathbf{Adv}^{\mathtt{nae\$}}_{\mathsf{NAE}}(\mathsf{A}) \leq \mathbf{Adv}^{\mathtt{nae}}_{\mathsf{NAE}}(\mathsf{A}') + q_e^2/2^{\nu+1}$ for some $(q_e, q_v, \sigma, \tilde{t}(\sigma))$-NAE adversary A'.*

The proof is easily obtained by relaxing the winning condition of $\mathsf{nae\$}$ notion so that a repeat of random nonces in two encryption queries immediately gives a win, and considering a bad event of nonce repeat in encryption queries. The probability of bad event is at most $q_e^2/2^{\nu+1}$, and unless the bad event occurs, the advantage is bounded by the NAE advantage of Definition 1.

For (tweakable) block ciphers, we define SPRP (strong pseudorandom permutation) and TSPRP (tweakable SPRP) advantages as their chosen-ciphertext security, that is,

$$\mathbf{Adv}^{\mathtt{sprp}}_{E}(\mathsf{A}_1) := \mathbf{Adv}^{\mathtt{ind}}_{(E,E^{-1}),(\mathsf{P},\mathsf{P}^{-1})}(\mathsf{A}_1),$$
$$\mathbf{Adv}^{\mathtt{tsprp}}_{\widetilde{E}}(\mathsf{A}_2) := \mathbf{Adv}^{\mathtt{ind}}_{(\widetilde{E},\widetilde{E}^{-1}),(\widetilde{\mathsf{P}},\widetilde{\mathsf{P}}^{-1})}(\mathsf{A}_2).$$

Here, A_2 can arbitrarily choose a tweak as it is a part of a query.

3 Decryption-Fast SIV

We present MRAE schemes that achieve our goal: a faster decryption than SIV. Specifically, we propose three generic compositions of an NAE and PRFs, assuming their keys are

```
Algorithm
$\mathsf{DFV1}[F_K, \mathsf{NAE}_{K'}].\mathsf{Enc}(A, M)$
    $V \leftarrow F_K(A, M)$
    $(C, T) \leftarrow \mathsf{NAE.Enc}_{K'}(V, A, M)$
    return $(V, C, T)$
```

```
Algorithm
$\mathsf{DFV1}[F_K, \mathsf{NAE}_{K'}].\mathsf{Dec}(V, A, C, T)$
    $Y \leftarrow \mathsf{NAE.Dec}_{K'}(V, A, C, T)$
    return $Y$
```

Figure 2: A generic construction DFV1. The function $F : \mathcal{K} \times \mathcal{A} \times \mathcal{M} \to \mathcal{V}$ is a vector-input PRF, NAE = (NAE.Enc, NAE.Dec) is a non-plain NAE scheme with a key space $\mathcal{K}'$, a nonce space $\mathcal{V} = \{0,1\}^{\nu}$, an AD space $\mathcal{A}$, a message space $\mathcal{M}$ and a tag space $\mathcal{T}$.

independent. The decryption algorithm is essentially that of the NAE we use. Later, using known rate-1 NAE schemes, we show several concrete instantiations achieving rate-1 decryption, and thus are faster than SIV using the same primitives. To be precise, these instantiations are not directly derived from our generic compositions. This is to achieve several efficiency properties, such as the use of single key.

DFV1. The first scheme, which we call DFV1, is the most basic one to realize our idea. It is depicted in the top of Figure 5 on page 96, and its pseudocode is shown in Figure 2. DFV1 uses a vector-input PRF $F : \mathcal{K} \times \mathcal{A} \times \mathcal{M} \to \mathcal{V}$ and a non-plain NAE NAE with a nonce space $\mathcal{V}$, where $\mathcal{V} = \{0,1\}^{\nu}$ for some $\nu > 0$. For encryption, it takes (A, M), derives $V \leftarrow F_K(A, M)$ and $(C, T) \leftarrow \mathsf{NAE.Enc}_{K'}(V, A, M)$. After receiving the tuple (V, A, C, T), the decryption of DFV1 is simply a decryption of $\mathsf{NAE}_{K'}$ taking this tuple. The derived V corresponds to the (synthetic) IV of SIV, however, its role is different in the decryption. DFV1 is simple and intuitive. A similar structure was informally described in [BNT19], however for a different purpose. PRF-to-IV proposed by [ABL+14a] is also closely related to DFV1. However, its authenticated decryption involves the decryption of the underlying NAE *and* the PRF to process the whole input. The purpose of PRF-to-IV is different from ours and it does not capture what we want to achieve. The security proof of DFV1 is rather obvious. It contains a hidden inefficiency for encryption because it processes AD twice while SIV processes AD only once.

DFV2 and DFV3. Our second and third schemes, DFV2 and DFV3, aim at removing the aforementioned inefficiency of DFV1. For generic composition, this inevitably needs a pNAE scheme as a component. The middle and the bottom of Figure 5 on page 96 depict DFV2 and DFV3. Their pseudocodes are shown in Figures 3 and 4 on page 95. They use two PRFs, $F : \mathcal{K} \times \mathcal{M} \to \mathcal{S}$ and $G : \mathcal{K}' \times \mathcal{S} \times \mathcal{M} \to \mathcal{V}$, $\mathcal{S} = \{0,1\}^{n'}$ and $\mathcal{V} = \{0,1\}^{\nu}$, and a pNAE scheme $\mathsf{pNAE}_{K''} = (\mathsf{pNAE.Enc}_{K''}, \mathsf{pNAE.Dec}_{K''})$. The nonce space of pNAE is $\mathcal{V}$ for DFV2 and $\mathcal{W} = \mathcal{V} \times \mathcal{S} = \{0,1\}^{\nu+n'}$ for DFV3. The tag length of pNAE is τ bits. DFV2 requires $n' \geq \tau$.

For DFV2, we use the unfinished decryption $\mathsf{pNAE.UDec}_{K''}$ (see Section 2.2) instead of $\mathsf{pNAE.Dec}_{K''}$. DFV3 does not need the unfinished decryption, but requires a larger nonce space for pNAE. For both schemes, an encryption expands the output by $\nu + \tau$ bits. As mentioned in Section 1, the idea of these schemes are closely related to the problem of adding an AD to a pNAE scheme, which was studied by Rogaway [Rog02]. In fact, our schemes are based on his two proposals, more specifically *ciphertext translation* (CT) for DFV2 and *nonce stealing* (NS) for DFV3. See Section 4 for more details.

```
Algorithm
DFV2[$F_K, G_{K'}, \mathsf{pNAE}_{K''}$].Enc($A, M$)
    $S \leftarrow F_K(A)$
    $V \leftarrow G_{K'}(S, M)$
    $(C, U) \leftarrow \mathsf{pNAE.Enc}_{K''}(V, M)$
    $T \leftarrow U \oplus \mathtt{msb}_\tau(S)$
    return $(V, C, T)$
```

```
Algorithm
DFV2[$F_K, G_{K'}, \mathsf{pNAE}_{K''}$].Dec($V, A, C, T$)
    $S' \leftarrow F_K(A)$
    $(M', U') \leftarrow \mathsf{pNAE.UDec}_{K''}(V, C)$
    $T' \leftarrow U' \oplus \mathtt{msb}_\tau(S')$
    if $T \neq T'$ then return $\perp$
    else return $M'$
```

Figure 3: A Ciphertext-translation-based DFV (DFV2). The functions $F : \mathcal{K} \times \mathcal{A} \to \mathcal{S}$ and $G : \mathcal{K}' \times \mathcal{S} \times \mathcal{M} \to \mathcal{V}$ are PRFs with $\mathcal{S} = \{0,1\}^{n'}$ and $\mathcal{V} = \{0,1\}^{\nu}$, and pNAE is a pNAE having a ν-bit nonce space. Tag T is τ bits satisfying $n' \geq \tau$.

```
Algorithm
DFV3[$F_K, G_{K'}, \mathsf{pNAE}_{K''}$].Enc($A, M$)
    $S \leftarrow F_K(A)$
    $V \leftarrow G_{K'}(S, M)$
    $W \leftarrow V \,\|\, S$
    $(C, T) \leftarrow \mathsf{pNAE.Enc}_{K''}(W, M)$
    return $(V, C, T)$
```

```
Algorithm
DFV3[$F_K, G_{K'}, \mathsf{pNAE}_{K''}$].Dec($V, A, C, T$)
    $S' \leftarrow F_K(A)$
    $W' \leftarrow V \,\|\, S'$
    $Y \leftarrow \mathsf{pNAE.Dec}_{K''}(W', C)$
    return $Y$
```

Figure 4: A nonce-stealing-based DFV (DFV3). The functions $F : \mathcal{K} \times \mathcal{A} \to \mathcal{S}$ and $G : \mathcal{K}' \times \mathcal{S} \times \mathcal{M} \to \mathcal{V}$ are PRFs, and $\mathsf{pNAE}_{K''}$ is a pNAE with a nonce space $\mathcal{W} = \{0,1\}^{\nu+n'}$, where $|V| = \nu$ and $|S| = n'$. Tag T is τ bits.

4 Security of DFV

4.1 Security of DFV1

Theorem 1. *For (q_e, q_d, σ, t)-MRAE adversary* A*,*

$$\mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV1}[F,\mathsf{NAE}]}(\mathsf{A}) \leq \mathbf{Adv}^{\mathtt{nae}}_{\mathsf{NAE}}(\mathsf{A}') + \mathbf{Adv}^{\mathtt{prf}}_{F}(\mathsf{B}) + \frac{q_e^2}{2^{\nu+1}},$$

where A′ *is a $(q_e, q_d, \sigma, \tilde{t}(\sigma))$-NAE adversary and* B *is a $(q_e + q_d, \sigma, \tilde{t}(\sigma))$-PRF adversary.*

Proof. We observe

$$\mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV1}[F,\mathsf{NAE}]}(\mathsf{A}) \leq \mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV1}[\mathsf{R},\mathsf{NAE}]}(\mathsf{A}) + \mathbf{Adv}^{\mathtt{prf}}_{F}(\mathsf{B}) \tag{5}$$

from the hybrid argument involving a URF $\mathsf{R} : \mathcal{A} \times \mathcal{M} \to \mathcal{V}$. Since A never repeats queries, V given to $\mathsf{DFV1}[\mathsf{R}, \mathsf{NAE}_{K'}]$.Enc is always independent and random. Combining this fact with Proposition 1, we have

$$\mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV1}[\mathsf{R},\mathsf{NAE}]}(\mathsf{A}) = \mathbf{Adv}^{\mathtt{nae\$}}_{\mathsf{NAE}}(\mathsf{A}') \leq \mathbf{Adv}^{\mathtt{nae}}_{\mathsf{NAE}}(\mathsf{A}'') + \frac{q_e^2}{2^{\nu+1}}, \tag{6}$$

for some $(q_e, q_d, \sigma, \tilde{t}(\sigma))$-NAE($) adversaries A′ and A″. Combining (5) and (6) concludes the proof. □

4.2 Security of DFV2

Theorem 2. *For (q_e, q_d, σ, t)-MRAE adversary* A*,*

$$\begin{aligned} &\mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV2}[F,G,\mathsf{pNAE}]}(\mathsf{A}) \\ &\leq \mathbf{Adv}^{\mathtt{prf}}_{F}(\mathsf{B}) + \mathbf{Adv}^{\mathtt{prf}}_{G}(\mathsf{B}') \\ &+ 2q_d\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{pNAE}}(\mathsf{A}') + q_d\mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\mathsf{pNAE}}(\mathsf{A}'') + \frac{(q_e+q_d)^2}{2^{n'+1}} + \frac{q_e^2}{2^{\nu+1}} + \frac{q_d}{2^{\tau}}. \end{aligned}$$

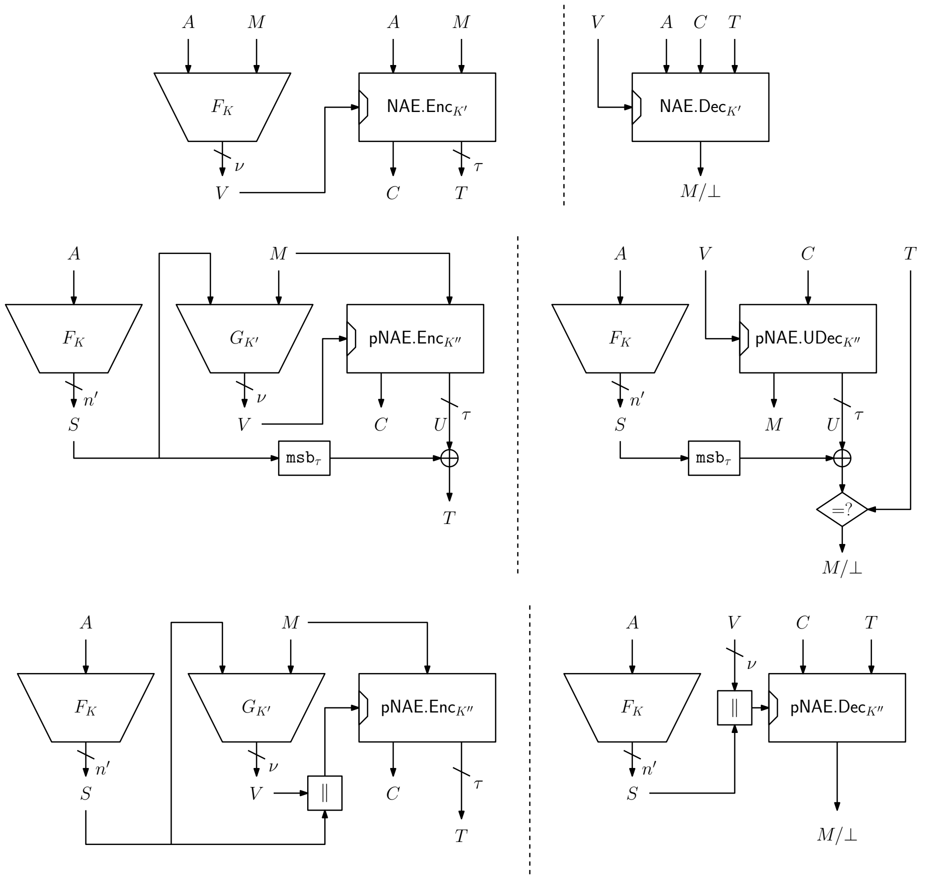

Figure 5: Schemes of DFV. (Top) a general scheme DFV1, (Middle) a ciphertext translation-based DFV2, and (Bottom) a nonce-stealing-based DFV3. For each scheme, the encryption (decryption) routine is shown in the left (right).

for some $(q_e + q_d, \sigma, \tilde{t}(\sigma))$*-PRF adversary* B*, and* $(q_e, \sigma, \tilde{t}(\sigma))$*-PRF adversary* B'*, and* $(q_e, \sigma, \tilde{t}(\sigma))$*-PRIV adversary* A' *and* $(q_e, \sigma, \tilde{t}(\sigma))$*-AUTH-1 adversary* A''*.*

Proof. Let $\mathsf{R}_F : \mathcal{A} \to \mathcal{S}$ and $\mathsf{R}_G : \mathcal{M} \times \mathcal{S} \to \mathcal{V}$ be URFs, and let $\mathsf{DFV2}_\mathsf{R}$ denote $\mathsf{DFV2}[\mathsf{R}_F, \mathsf{R}_G, \mathsf{pNAE}_{K''}]$. Let $\mathsf{DFV2}^*_\mathsf{R}$ be a variant of $\mathsf{DFV2}_\mathsf{R}$ that uses URF $\mathsf{R}^*_G : \mathcal{A} \times \mathcal{M} \to \mathcal{V}$ instead of R_G and derives as $S \leftarrow \mathsf{R}_F(A)$ and $V \leftarrow \mathsf{R}^*_G(A, M)$. Thus, S is not involved in the computation of V for $\mathsf{DFV2}^*_\mathsf{R}$. Let $f[\mathsf{R}_F, \mathsf{R}_G](A, M) = (S, V)$, where $S = \mathsf{R}_F(A)$ and $V = \mathsf{R}_G(S, M)$. Similarly, let $f^*[\mathsf{R}_F, \mathsf{R}^*_G](A, M) = (S, V)$, where $S = \mathsf{R}_F(A)$ and $V = \mathsf{R}^*_G(A, M)$. Note that $f[\mathsf{R}_F, \mathsf{R}^*_G]$ ($f^*[\mathsf{R}_F, \mathsf{R}^*_G]$) is a part of the algorithm of $\mathsf{DFV2}_\mathsf{R}$ ($\mathsf{DFV2}^*_\mathsf{R}$), and this is the sole difference between them. If q is the number of queries to $f[\mathsf{R}_F, \mathsf{R}_G]$ or $f^*[\mathsf{R}_F, \mathsf{R}^*_G]$,

$$\mathbf{Adv}^{\mathtt{ind}}_{f[\mathsf{R}_F, \mathsf{R}_G], f^*[\mathsf{R}_F, \mathsf{R}^*_G]}(q) \leq \frac{q^2}{2^{n'+1}} \tag{7}$$

holds from the standard collision analysis on S.

We observe that $\mathsf{DFV2}^*_\mathsf{R}$ can be interpreted as an instance of $\mathsf{DFV1}$ with NAE derived by the ciphertext translation (CT) [Rog02] applied to $\mathsf{pNAE}_{K''}$. See Figure 6 on page 99 for the definition of CT. Here, CT turns a pNAE (pNAE) into a non-plain NAE (NAE) by using a keyed function F. The tag of NAE is a sum of the tag of pNAE and $\mathtt{msb}_\tau(F(A))$. We note that the original definition [Rog02] is more general than Figure 6, in that the ciphertext and the tag are not explicitly separated, which we call the unified ciphertext (for both NAE and pNAE). To help understanding, we elaborate a bit on the original. For $X, Y \in \{0,1\}^*$, let $X \hat{\oplus} Y$ be the XOR of them by prepending zeros to the shorter one (*e.g.*, $0101001 \hat{\oplus} 111 = 0101110$). When CT is applied, the NAE encryption for (N, A, M) for a non-empty A is done by first computing the unified ciphertext $\mathcal{C}_p$ of pNAE encryption taking (N, M), and the unified ciphertext of CT is derived as $\mathcal{C}_p \hat{\oplus} F_{K'}(A)$ for some keyed function $F_{K'} : \{0,1\}^* \to \{0,1\}^\tau$. The NAE decryption for $(N, A, \mathcal{C})$ is simply the pNAE decryption of $(N, A, \mathcal{C} \hat{\oplus} F_{K'}(A))$. When A is an empty string, the NAE encryption and decryption are identical to the pNAE encryption and decryption ignoring A.

Figure 6 is essentially an instance of this general CT by specifying the unified ciphertext as a concatenation of a ciphertext and a tag, and assuming AD is never empty, say with some encoding. The keyed hashing function to A corresponds to the composition of $\mathtt{msb}_\tau(*)$ and $\mathsf{R}_F(*)$. Decryption is also compliant with the original CT since checking $U' \oplus \mathtt{msb}_\tau(S') = T$ (in Figure 6) is equivalent to checking $T \oplus \mathtt{msb}_\tau(S') = U'$. In the form of the original CT, the latter is the verification procedure for the decryption of pNAE consisting of $\mathsf{pNAE.UDec}$ and Cmp (see Section 2.2) if $T = \mathtt{lsb}_\tau(\mathcal{C})$ for the unified ciphertext $\mathcal{C}$.

From Figure 6, we observe the equivalence

$$\mathsf{DFV2}^*_{\mathsf{R}} \equiv \mathsf{DFV1}[\mathsf{R}^*_G, \mathsf{CT}[\mathsf{pNAE}, \mathsf{R}_F]].$$

From the security proof of CT [Rog02, Theorem 2], for a (q_e, σ, t)-PRIV adversary A_1 and a (q_e, σ, t)-AUTH-1 adversary A_2, we have

$$\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{CT}[\mathsf{pNAE},\mathsf{R}_F]}(\mathsf{A}_1) \le \mathbf{Adv}^{\mathtt{priv}}_{\mathsf{pNAE}}(\mathsf{A}'_1), \tag{8}$$

$$\mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\mathsf{CT}[\mathsf{pNAE},\mathsf{R}_F]}(\mathsf{A}_2) \le \mathbf{Adv}^{\mathtt{priv}}_{\mathsf{pNAE}}(\mathsf{A}'_2) + \mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\mathsf{pNAE}}(\mathsf{A}''_2) + \frac{1}{2^\tau}, \tag{9}$$

for some $(q_e, \sigma, \tilde{t}(\sigma))$-PRIV adversaries A'_1 and A'_2, and a $(q_e, \sigma, \tilde{t}(\sigma))$-AUTH-1 adversary A''_2.

Therefore, we have

$$\begin{aligned} \mathbf{Adv}^{\mathtt{nae}}_{\mathsf{CT}[\mathsf{pNAE},\mathsf{R}_F]}(\mathsf{A}) &\le \mathbf{Adv}^{\mathtt{priv}}_{\mathsf{CT}[\mathsf{pNAE},\mathsf{R}_F]}(\mathsf{A}_p) + q_d \mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\mathsf{CT}[\mathsf{pNAE},\mathsf{R}_F]}(\mathsf{A}_a) \\ &\le (q_d + 1)\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{pNAE}}(\mathsf{A}'_p) + q_d \mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\mathsf{pNAE}}(\mathsf{A}'_a) + \frac{q_d}{2^\tau}, \end{aligned} \tag{10}$$

for some $(q_e, \sigma, \tilde{t}(\sigma))$-PRIV adversary A_p, $(q_e, \sigma, \tilde{t}(\sigma))$-PRIV adversary A'_p, and $(q_e, \sigma, \tilde{t}(\sigma))$-AUTH-1 adversaries A_a and A'_a. Here, the first inequality follows from Rogaway and Shrimpton [RS06, Proposition 8] and the second follows from (8) and (9).

Thus, we have

$$\begin{aligned} \mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV2}_\mathsf{R}}(\mathsf{A}) &\le \mathbf{Adv}^{\mathtt{ind}}_{\mathsf{DFV2}_\mathsf{R},\mathsf{DFV2}^*_\mathsf{R}}(\mathsf{A}) + \mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV2}^*_\mathsf{R}}(\mathsf{A}) \\ &\le \mathbf{Adv}^{\mathtt{ind}}_{f[\mathsf{R}_F,\mathsf{R}_G],f^*[\mathsf{R}_F,\mathsf{R}^*_G]}(q_e + q_d) + \mathbf{Adv}^{\mathtt{nae}}_{\mathsf{CT}[\mathsf{pNAE},\mathsf{R}_F]}(\mathsf{A}') + \frac{q_e^2}{2^{\nu+1}} \\ &\le (q_d + 1)\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{pNAE}}(\mathsf{A}'_p) + q_d \mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\mathsf{pNAE}}(\mathsf{A}'_a) + \frac{(q_e + q_d)^2}{2^{n'+1}} + \frac{q_e^2}{2^{\nu+1}} + \frac{q_d}{2^\tau} \end{aligned}$$

for some (q_e, q_d, σ, t)-NAE adversary A' and $(q_e, \sigma, \tilde{t}(\sigma))$-PRIV adversary A'_p and $(q_e, \sigma, \tilde{t}(\sigma))$-AUTH-1 adversary A'_a. The second inequality follows from Theorem 1, and the third follows from (7) and (10). This completes the proof. □

4.3 Security of DFV3

Theorem 3. *For a (q_e, q_d, σ, t)-MRAE adversary* A,

$$\begin{aligned}
&\mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV3}[F,G,\mathsf{pNAE}]}(\mathsf{A}) \\
&\leq \mathbf{Adv}^{\mathtt{prf}}_{F}(\mathsf{B}) + \mathbf{Adv}^{\mathtt{prf}}_{G}(\mathsf{B}') + \mathbf{Adv}^{\mathtt{priv}}_{\mathsf{pNAE}}(\mathsf{A}') + q_d\left(\mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\mathsf{pNAE}}(\mathsf{A}'') + \frac{1}{2^{n'}}\right) \\
&+ \frac{(q_e+q_d)^2}{2^{n'+1}} + \frac{q_e^2}{2^{\nu+1}}
\end{aligned}$$

holds for some $(q_e + q_d, \sigma, \tilde{t}(\sigma))$-PRF adversary B, *and $(q_e, \sigma, \tilde{t}(\sigma))$-PRF adversary* B', *and $(q_e, \sigma, \tilde{t}(\sigma))$-PRIV adversary* A' *and $(q_e, \sigma, \tilde{t}(\sigma))$-AUTH-1 adversary* A''.

We note that the term $q_d/2^{\tau}$, which should be included for any AE of τ-bit tag, is implicitly brought by $q_d\mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\mathsf{pNAE}}(\mathsf{A}'')$.

Proof. As mentioned, DFV3 is based on the (generalized) nonce stealing [Rog02], NS, that converts a pNAE to an NAE. See Figure 7 on page 99 for the definition of NS. The original form of nonce stealing adopts the unified ciphertext as well as the case of ciphertext translation. NS composes $\mathsf{pNAE}_{K''}$ with $\nu + n'$-bit nonce and a PRF $F_K : \mathcal{A} \to \mathcal{S}$ with $\mathcal{S} = \{0,1\}^{n'}$. The resulting NAE is $\mathsf{NS}[\mathsf{pNAE}_{K''}, F_K]$ which has a ν-bit nonce.

Let $\mathsf{DFV3_R}$ denote $\mathsf{DFV3}[\mathsf{R}_F, \mathsf{R}_G, \mathsf{pNAE}_{K''}]$. As before, the security of $\mathsf{DFV3}[F_K, G_{K'}, \mathsf{pNAE}_{K''}]$ is a sum of MRAE advantage of $\mathsf{DFV3_R}$ and the PRF advantages of F_K and $G_{K'}$. Let us focus on the former. Let $\mathsf{DFV3^*_R}$ denote a variant that uses $V \leftarrow \mathsf{R}^*_G(A, M)$ instead of $V \leftarrow \mathsf{R}_G(S, M)$ in $\mathsf{DFV3_R}$. We observe that $\mathsf{DFV3^*_R}$ is an NS-based instance of DFV1 as

$$\mathsf{DFV3^*_R} \equiv \mathsf{DFV1}[\mathsf{R}^*_G, \mathsf{NS}[\mathsf{pNAE}_{K''}, \mathsf{R}_F]].$$

In our proof, N in Figure 7 corresponds to V. We need the privacy and authenticity bounds of $\mathsf{NS}[\mathsf{pNAE}_{K''}, \mathsf{R}_F]$. They are as follows.

Lemma 1. *For a (q_e, σ, t)-PRIV adversary* A_1 *and a (q_e, σ, t)-AUTH-1 adversary* A_2,

$$\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{NS}[\mathsf{pNAE},\mathsf{R}_F]}(\mathsf{A}_1) \leq \mathbf{Adv}^{\mathtt{priv}}_{\mathsf{pNAE}}(\mathsf{A}'_1) \tag{11}$$

$$\mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\mathsf{NS}[\mathsf{pNAE},\mathsf{R}_F]}(\mathsf{A}_2) \leq \mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\mathsf{pNAE}}(\mathsf{A}'_2) + \frac{1}{2^{n'}}, \tag{12}$$

for some $(q_e, \sigma, \tilde{t}(\sigma))$-PRIV adversary A'_1 *and $(q_e, \sigma, \tilde{t}(\sigma))$-AUTH-1 adversary* A'_2.

Proof of Lemma 1. To prove (11), we simply observe that A'_1 can simulate $\mathsf{R}_F(A) \to S$.

To prove (12), we use a similar argument[4] as the authenticity proof of CT [Rog02]. Let A be an AUTH-1 adversary against $\mathsf{NS}[\mathsf{pNAE}_{K''}, \mathsf{R}_F]$ and let B be an AUTH-1 adversary against $\mathsf{pNAE}_{K''}$. For convenience, we allow both adversaries to make trivial decryption queries (*i.e.*, for A, a decryption query (V, A, C, T) is trivial when an encryption query (V, A, M) has already been made and (C, T) has returned) and the adversary is said to win iff it receives a non-$\bot$ response from the decryption oracle from a non-trivial decryption query. Here, B uses A and R_F as internal routines, and records all the transcript generated by them. When an encryption query (V, A, M) is made by A, B invokes $\mathsf{R}_F(A) \to S$ and makes an encryption query (W, M) where $W = V \,\|\, S$. For this query, B receives (C, T) from $\mathsf{pNAE.Enc}_{K''}$ and it will be given to A. When a decryption query (V', A', C', T') is made by A, B invokes $\mathsf{R}_F(A') \to S'$ and makes a decryption query (W', C') where $W' = V' \,\|\, S'$, and passes the return value from $\mathsf{pNAE.Dec}_{K''}$ to A.

[4] The original proof of nonce-stealing does not need it and is much simpler because it does not compress AD.

```
Algorithm
CT[pNAE_K, F_K'].Enc(N, A, M)
    $S \leftarrow F_{K'}(A)$
    $(C, U) \leftarrow \mathsf{pNAE.Enc}_K(N, M)$
    $T \leftarrow U \oplus \mathtt{msb}_\tau(S)$
    return $(C, T)$
```

```
Algorithm
CT[pNAE_K, F_K'].Dec(N, A, C, T)
    $S' \leftarrow F_{K'}(A)$
    $(M', U') \leftarrow \mathsf{pNAE.UDec}_K(N, C)$
    $T' \leftarrow U' \oplus \mathtt{msb}_\tau(S')$
    if $T \neq T'$ then return $\perp$
    else return $M'$
```

Figure 6: Ciphertext Translation.

```
Algorithm
NS[pNAE_K, F_K'].Enc(N, A, M)
    $S \leftarrow F_{K'}(A)$
    $W \leftarrow N \,\|\, S$
    $(C, T) \leftarrow \mathsf{pNAE.Enc}_K(W, M)$
    return $(C, T)$
```

```
Algorithm
NS[pNAE_K, F_K'].Dec(N, A, C, T)
    $S' \leftarrow F_{K'}(A)$
    $W' \leftarrow N \,\|\, S'$
    $Y \leftarrow \mathsf{pNAE.Dec}_K(W', C)$
    return $Y$
```

Figure 7: Generalized Nonce Stealing.

The event that A forges (against $\mathsf{NS}[\mathsf{pNAE}_{K''}, \mathsf{R}_F]$) happens iff (1) B forges or (2) B's decryption query is trivial but A's is not. Let $(V_i, S_i, A_i, M_i, C_i, T_i)$ be the tuple generated by the i-th encryption query from A and the response for $\mathsf{pNAE}_{K''}$ and R_F. We observe that, (2) implies that there exists an i-th encryption query from A, and $(V', S', C', T') = (V_i, S_i, C_i, T_i)$ (which is needed for B to make $(W' = V' \,\|\, S', C', T')$ a trivial query), and $A' \neq A_i$ (which is needed for A to make it non-trivial). If we write this event as Coll, we have

$$\Pr[\mathsf{A}^{\mathsf{NS}[\mathsf{pNAE}_{K''}, \mathsf{R}_F]} \text{ forges }] \leq \Pr[\mathsf{B}^{\mathsf{pNAE}_{K''}} \text{ forges }] + \Pr[\mathsf{B}^{\mathsf{pNAE}_{K''}} \text{ invokes } \mathsf{Coll}],$$

where each probability term assumes uniform K'', and observe that the last term is at most $1/2^{n'}$ since this is a non-trivial collision of $S' = \mathsf{R}_F(A')$ and $S_i = \mathsf{R}_F(A_i)$ for this target i-th query. Note that $V' = V_i$ implies such target exists at most once among q_e encryption queries. This proves (12). We conclude the proof of Lemma 1.

Thus we have

$$\begin{aligned}
\mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV3_R}}(\mathsf{A}) &\leq \mathbf{Adv}^{\mathtt{ind}}_{\mathsf{DFV3_R}, \mathsf{DFV3^*_R}}(\mathsf{A}) + \mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV3^*_R}}(\mathsf{A}) \\
&\leq \mathbf{Adv}^{\mathtt{ind}}_{f[\mathsf{R}_F, \mathsf{R}_G], f^*[\mathsf{R}_F, \mathsf{R}^*_G]}(\mathsf{B}) + \mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV3^*_R}}(\mathsf{A}) \\
&\leq \frac{(q_e + q_d)^2}{2^{n'+1}} + \mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV3^*_R}}(\mathsf{A}) \\
&\leq \frac{(q_e + q_d)^2}{2^{n'+1}} + \frac{q_e^2}{2^{\nu+1}} + \mathbf{Adv}^{\mathtt{nae}}_{\mathsf{NS}[\mathsf{pNAE}, \mathsf{R}_F]}(\mathsf{A}') \\
&\leq \frac{(q_e + q_d)^2}{2^{n'+1}} + \frac{q_e^2}{2^{\nu+1}} + \mathbf{Adv}^{\mathtt{priv}}_{\mathsf{NS}[\mathsf{pNAE}, \mathsf{R}_F]}(\mathsf{A}'_p) + q_d \mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\mathsf{NS}[\mathsf{pNAE}, \mathsf{R}_F]}(\mathsf{A}'_a) \\
&\leq \frac{(q_e + q_d)^2}{2^{n'+1}} + \frac{q_e^2}{2^{\nu+1}} + \mathbf{Adv}^{\mathtt{priv}}_{\mathsf{pNAE}}(\mathsf{A}''_p) + q_d \mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\mathsf{pNAE}}(\mathsf{A}''_a) + \frac{q_d}{2^{n'}},
\end{aligned}$$

for some $(q_e + q_d, \tilde{t}(\sigma))$-PRF adversary B, and $(q_e, q_d, \sigma, \tilde{t}(\sigma))$-NAE adversary A', and $(q_e, \sigma, \tilde{t}(\sigma))$-PRIV adversaries A'_p and A''_p, and $(q_e, \sigma, \tilde{t}(\sigma))$-AUTH-1 adversaries A'_a and A''_a. Here, the third inequality follows from (7), the fourth follows from Theorem 1, the fifth follows from [RS06, Proposition 8] and the last follows from Lemma 1. This concludes the proof. □

5 Discussions

5.1 Tightness of the Bounds, and Comparison with SIV

The bounds of DFV2 and DFV3 do not seem to be tight, in particular with respect to the computational terms regarding pNAE. The term q_d is multiplied by the PRIV term and AUTH-1 term for DFV2. If we simply instantiate pNAE by (any version of) OCB in DFV2, its PRIV bound is $O(\sigma^2/2^n)$ and its AUTH-1 bound is $O(\sigma^2/2^n + 1/2^\tau)$ for σ queried blocks. Thus, assuming $n = \nu = n'$ the bound for DFV2 becomes $O(q_d\sigma^2/2^n + q_d/2^\tau)$. This cubic degradation is worse than the quadratic bound of a birthday-secure, block cipher-based instantiation of SIV, whose MRAE bound is $O(\sigma^2/2^n + q/2^n)$ for q queries and σ blocks (from [RS06, Theorem 2], assuming birthday-secure components). In case of DFV3, q_d is only multiplied by AUTH-1 term of pNAE, however we basically obtain the same bound as DFV2 if pNAE is OCB. If we use OCB3, a more rigorous comparison could be possible with an improved bound of OCB3 [BN17], however DFV3 is still inferior to SIV as long as we use Theorem 3 in a black-box way.

This cubic degradation is undesirable, and because it is already present in the bounds of the original CT and NS transformations, it seems not easy to avoid. In general, concrete instantiations of DFV with dedicated analysis can give better bounds than using the black-box application of the above bounds. In fact, this happens for our instantiations at Section 6.

5.2 Wrong Variation of DFV1

While our core idea is rather simple, the realization of it needed some cares. In fact, some variations/optimizations of DFV1 or DFV2 or DFV3 can easily go wrong. For example, consider a variation of DFV1 that uses $V = F_K(M) \,\|\, A$ as a (variable-length) nonce. It might save the total computation depending on the difference in efficiencies of F and NAE, and looks secure as it involves all the input information to derive V. However, this is insecure since it leaks the coincidence of plaintexts via V, even if ADs are different. This weakness is related (but not identical) to a problematic NAE scheme discussed by Bellare *et al.* [BNT19] that derives a nonce from a plaintext, say by taking a (key-less) hash of it. This easily leaks the plaintext by an off-line attack if it has a low entropy.

5.3 Intractability of Refined Variants of DFV2 and DFV3

One might want to optimize DFV2 and DFV3 in terms of computation. For example, by XORing S to (the first n' bits of) M will reduce the input length of G hence its computation cost. However, such seemingly minor improvement makes the security proof intractable. To see this, consider a variant of DFV2 that uses $M \oplus (S \,\|\, 0^{|M|-n'})$ for the input of $G_{K'}$, which we call DFV2a. For simplicity, we assume $|M| = n'$. Then, DFV2a generates V as $V \leftarrow G_{K'}(M \oplus S)$ and $S \leftarrow F_K(A)$. If we are to prove the security of DFV2a, it requires to bound the MRAE advantage of $\mathsf{DFV2a}_{\mathsf{R}} \coloneqq \mathsf{DFV2a}[\mathsf{R}_F, \mathsf{R}_G, \mathsf{pNAE}_{K''}]$. In a similar manner to the proof of DFV2, we introduce $\mathsf{DFV2a}^*_{\mathsf{R}}$ that generates $V \leftarrow \mathsf{R}^*_G(A, M)$ (while $S \leftarrow \mathsf{R}_F(A)$). The difference between $\mathsf{DFV2a}_{\mathsf{R}}$ and $\mathsf{DFV2a}^*_{\mathsf{R}}$ is their processes to derive (S, V), namely

$$f_a[\mathsf{R}_F, \mathsf{R}_G] = (S, V), \text{ where } S = \mathsf{R}_F(A), V = \mathsf{R}_G(M \oplus S) \text{ for } \mathsf{DFV2a}_{\mathsf{R}},$$
$$f^*_a[\mathsf{R}_F, \mathsf{R}^*_G] = (S, V), \text{ where } S = \mathsf{R}_F(A), V = \mathsf{R}^*_G(A, M) \text{ for } \mathsf{DFV2a}^*_{\mathsf{R}}.$$

However, unlike the case of DFV2, we cannot reduce the indistinguishability between $\mathsf{DFV2a}_{\mathsf{R}}$ and $\mathsf{DFV2a}^*_{\mathsf{R}}$ (Game2a) to that between $f_a[\mathsf{R}_F, \mathsf{R}_G]$ and $f^*_a[\mathsf{R}_F, \mathsf{R}^*_G]$, as the latter is easy to distinguish by finding a collision on inputs to G. The point is that, in the case of DFV2, the reduction works as the indistinguishability between $\mathsf{DFV2}_{\mathsf{R}}$ and $\mathsf{DFV2}^*_{\mathsf{R}}$

does not collapse even if S is given to the adversary. On the other hand, if we give S to the adversary in $\mathsf{Game2a}$, she wins with a high probability, as she can easily invoke an input collision of G when querying $\mathsf{DFV2a}_{\mathsf{R}}$, hence, by checking the collision of V, she can distinguish the two oracles. Note that how S is hidden from the adversary is not quantifiable as the underlying pNAE has only a computational security guarantee.

One might think of another proof path that first applies the hybrid argument involving $\mathsf{pNAE}_{K''}$ and the ideal pNAE scheme ($\$(*, *)$), keeping R_F and R_G. This also faces a similar problem; the confidentiality of S cannot be ensured in $\mathsf{DFV2}$, and the adversary who learns S can easily distinguish the two worlds, as she can invoke a collision on V for a pair of different ADs.

In other words, the problem is that V must be unique to ensure the confidentiality of S (equivalent to the randomness of NAE outputs on distinct nonces), while the confidentiality of S is needed to ensure the distinctness of V. This poses a kind of chicken-or-the-egg dilemma. We remark that this problem is not present in [Rog02] because the uniqueness of nonce for encryption queries is guaranteed everywhere in the game.

Another potential optimization option for $\mathsf{DFV2}$ is the use of an almost XOR-universal hash function [WC81] for $G_{K'}$ instead of a PRF. This works for pNAE-to-non-plain NAE conversion [Rog02]. For $\mathsf{DFV3}$, an option is to take an XOR of V and S instead of $V \,\|\, S$ to derive W to reduce the required nonce length for pNAE we use. However, as far as we tried, proving/disproving their security is hard for the same reason as above.

Perhaps this problem is related to the nature of generic composition and an interesting future topic. We think that these optimizations are generally possible for the specific constructions, by considering the ad-hoc security proofs.

5.4 RUP Security

One security feature that has not discussed so far is a security under a release of unverified plaintext (RUP) introduced by Andreeva *et al.* [ABL+14a]. They consider the situation where the decryption oracle leaks the unverified plaintext irrespective of the result of verification, thus allows the adversary to access the unverified decryption oracle. As mentioned at Section 2.2, an unverified decryption oracle is trivially obtained by the unfinished decryption routine (if exists). For an MRAE scheme MRAE, its unverified decryption is written as $\mathsf{MRAE.UvDec}$. It takes (A, C) and returns an unverified plaintext M'. We also consider a verification oracle $\mathsf{MRAE.Verf}$, which takes (A, C, T) and returns $\top$ if $\mathsf{MRAE.Dec}(A, C, T) \neq \bot$ (*i.e.*, authentication is successful), and $\bot$ otherwise.

We here briefly discuss RUP security of DFV, particularly focusing on authenticity. The relevant notion is INT-RUP (for INTegrity under RUP). The INT-RUP notion for an MRAE scheme MRAE is defined as follows.

$$\mathbf{Adv}^{\texttt{int-rup}}_{\mathsf{MRAE}}(\mathsf{A}) := \Pr[K \xleftarrow{\$} \mathcal{K} : \mathsf{A}^{\mathsf{MRAE.Enc}_K, \mathsf{MRAE.UvDec}_K, \mathsf{MRAE.Verf}_K} \text{ forges }],$$

which means the probability of receiving $\top$ from $\mathsf{MRAE.Verf}_K$ without making a query leading to a trivial win [ABL+14a].

We show that $\mathsf{DFV1}$ is INT-RUP secure if the underlying NAE is. More formally, let A be a $(q_e, q_d, q_v, \sigma, t)$-INT-RUP adversary, which uses q_e encryption queries, q_d unverified decryption queries, and q_v verification queries, with total queried blocks σ, and time complexity t. Then we have

$$\mathbf{Adv}^{\texttt{int-rup}}_{\mathsf{DFV1}[F,\mathsf{NAE}]}(\mathsf{A}) \leq \mathbf{Adv}^{\texttt{int-rup}}_{\mathsf{NAE}}(\mathsf{A}') + \mathbf{Adv}^{\texttt{prf}}_{F}(\mathsf{B}) + \frac{q_e^2}{2^{\nu+1}}, \tag{13}$$

for some $(q_e, q_d, q_v, \sigma, \tilde{t}(\sigma))$-int-rup adversary A' and $(q_e + q_d + q_v, \sigma, \tilde{t}(\sigma))$-PRF adversary B. The inequality follows from the fact that INT-RUP adversary against $\mathsf{DFV1}[\mathsf{R}_F, \mathsf{NAE}_{K'}]$ is simulatable by another INT-RUP adversary against $\mathsf{NAE}_{K'}$ that uses random V instead of chosen V. We note that this result is not implied by the INT-RUP analysis of PRF-to-IV [ABL+14b, Proposition 10] due to the difference in the decryption.

This implies that we can use an INT-RUP-secure NAE scheme for DFV1 to ensure the whole INT-RUP security. For block cipher-based schemes, some examples are CCM [ABL+14a], SILC [IMG+17] [IMG+14], ΘCBt [HSY17], OCB-IC [ZWH+17], GCM-RUP [ADL17], and mCPFB [CDN16], while the only the last one[5] is known to have a rate large than 1/2.

The bound is tight when the INT-RUP security of $\mathsf{NAE}_{K'}$ does not depend on the choice of V at encryption. For example, when $\mathsf{NAE}_{K'}$ is OCB, the INT-RUP attack just needs one encryption query of any nonce and an unverified decryption query, both having about n blocks [ABL+14a]. In this case the both sides of (13) is close to one with two queries. Conversely, if $\mathsf{NAE}_{K'}$ is broken by INT-RUP attack but only with a small subset of possible nonce values at encryption, it might be the case that DFV1 has a non-negligible INT-RUP security. Similar analysis is applicable to DFV2 and DFV3, though it seems more involved and will need more study. Analysis of security under a refined/combined RUP security notion [CDD+19] is also an interesting direction.

Fall-back to SIV. A useful feature of DFV is that it simply enables a fall-back to SIV by treating pNAE/NAE as an IV-based encryption scheme. Taking DFV2 as an example, this can be done by changing the algorithms as follows. For encryption, we do not send T[6]. For decryption receiving (V, A, C), we follow SIV : perform an integrity check at V by computing $V' = G_{K'}(S', M')$ for $S' = F_K(A)$ and $M' = \mathsf{pNAE.UvDec}_{K''}(V, C)$, thus assuming the existence of unverified decryption. It is not hard to see that this is a secure instantiation of SIV, because $(\mathsf{pNAE.Enc}, \mathsf{pNAE.UvDec})$ works as a secure IV-based encryption[7] from the privacy condition of pNAE, and the derivation of V from (A, M) is a PRF up to a collision on S.

This feature is beneficial when the necessity of RUP security is not determined at the deployment. Note that SIV is INT-RUP secure [ABL+14a, ABL+14b], that is, it has authenticity under RUP. For privacy/confidentiality under RUP, it meets a weak notion called PA1, and to meet a stronger notion, PA2, a further enhancement on SIV is needed [ABL+14a]. Roughly, these notions denote the indistinguishability of $(\mathsf{Enc}_K, \mathsf{Dec}_K)$ from $(\mathsf{Enc}_K, \mathsf{Ext})$ for some extractor Ext which does not possess K, either with access to query history (PA1) or without it (PA2).

6 Instantiations

6.1 OCB-DFV : Birthday-Secure Parallel MRAE

We present an instantiation of DFV2 using OCB, which we call OCB-DFV. While DFV2 is a generic composition needing three keys, the TBC XEX inside OCB allows to effectively reduce the number of keys to single block cipher key. We try to keep the original algorithms, and at the same time avoid a key derivation function (KDF) applied on the top. This is because the use of KDF imposes a non-trivial computational overhead, and it may require another primitive such as a hash function.

Figure 8 on page 107 shows OCB-DFV. Following [Rog04], by writing $2a$ for $a \in \{0,1\}^n$, we mean a constant $\mathrm{GF}(2^n)$ multiplication by $\mathtt{x}$, also called a doubling. Similarly, $3a$ means $2a \oplus a$. See Appendix B for more details. For the sake of simplicity of pseudocode,

[5] We assume the polynomial hash in GCM-RUP is equally costly to block cipher-based MACs. For ΘCBt, it only fulfills a relaxed security notion from INT-RUP.

[6] Alternatively, we send T and require the decryption to ignore it, though it needs to modify the security goal as a forgery on T would not be caught.

[7] We need the length-preserving property for security, as mentioned by [RS06].

OCB-DFV is based on OCB2 and PMAC [Rog04]. OCB3 could be used as well. Since OCB2 has been shown to be broken [IIMP19], the algorithm is a fixed version (OCB2f) shown in [IIMP19]. Figure 10 in Appendix E shows the pseudocodes of the necessary components of OCB2f for OCB-DFV, namely the encryption algorithm and the unfinished decryption algorithm for the empty AD. The algorithm of PMAC in the left of Figure 8 is a generalization from those defined at [Rog04, IIMP19], where the input $\mathsf{c} \in \{0,1\}^n$ was originally fixed to 0^n. We also removed the constant multiplication by 3^2 at line 2 of PMAC as this is needed only if nonce can be 0^n for OCB2f in the game, which is not our case; V is $n-2$ bits and padded with 10 when given to OCB2f (line 3 of the right of Figure 8).

OCB-DFV is the first block cipher MRAE mode that achieves rate-1 decryption and rate-1/2 encryption. It is fully parallelizable. The security is $n/2$-bit as with many block cipher modes. See also Table 1.

Security Bound of OCB-DFV.

Theorem 4. *Let $n = 128$. For (q_e, q_d, σ, t)-MRAE adversary* A*,*

$$\mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{OCB\text{-}DFV}[E]}(\mathsf{A}) \le \mathbf{Adv}^{\mathtt{sprp}}_{E}(\mathsf{B}) + \frac{13\sigma^2}{2^n} + \frac{5q_d}{2^\tau}$$

holds for some $(\sigma, \tilde{t}(\sigma))$-PRF adversary B.

The proof specifies $n = 128$, however it works for any n as long as the instance of XEX shown in (14) is secure (see [Rog04]).

Proof. Following [Rog04, IIMP19], we can represent OCB-DFV as a mode of TBC called XEX[8]. Let $\widetilde{\mathsf{P}} : \mathcal{TW} \times \mathcal{M} \to \mathcal{M}$ be a TURP, where $\mathcal{TW} = \{0,1\}^n \times \mathcal{I} \times \mathcal{D}$, $\mathcal{M} = \{0,1\}^n$, $\mathcal{I} = [2^n]$, and $\mathcal{D} = \{0,1,2\}$. We introduce $\mathsf{iOCB\text{-}DFV}[\widetilde{\mathsf{P}}]$ in Figure 11 in Appendix E as an ideal variant[9] of $\mathsf{OCB\text{-}DFV}[\mathsf{P}]$ using $\widetilde{\mathsf{P}}$. It consists of $\mathsf{iPMAC}[\widetilde{\mathsf{P}}]$ shown in the left of Figure 11 and $\Theta\mathsf{CB2f}[\widetilde{\mathsf{P}}]$ shown in the bottom of Figure 10. Using a URP $\mathsf{P} : \mathcal{M} \to \mathcal{M}$, our instantiation of XEX is

$$\mathsf{XEX}[\mathsf{P}](T_W, X) = \begin{cases} 2^i 3^j L \oplus \mathsf{P}(2^i 3^j L \oplus X) & \text{if } j = 0 \\ \mathsf{P}(2^i 3^j L \oplus X) & \text{if } j \in \{1,2\} \end{cases} \tag{14}$$

where $T_W = (N, i, j) \in \mathcal{TW}$ and $L = \mathsf{P}(N)$. We observe that

$$\mathsf{iOCB\text{-}DFV}[\mathsf{XEX}[\mathsf{P}]] \equiv \mathsf{OCB\text{-}DFV}[\mathsf{P}]. \tag{15}$$

Let iPMAC_F and iPMAC_G denote $\mathsf{iPMAC}[\widetilde{\mathsf{P}}](0^n, *)$ and $\mathsf{iPMAC}[\widetilde{\mathsf{P}}](0^{n-1}1, *)$. We write R_F and R_G to denote the independent URFs : $\{0,1\}^* \to \{0,1\}^n$. Let $\mathsf{DFV2}_\Theta$ denote $\mathsf{DFV2}[\mathsf{iPMAC}_F, \mathsf{iPMAC}_G, \Theta\mathsf{CB2f}[\widetilde{\mathsf{P}}]]$ and $\mathsf{DFV2}^*_\Theta$ denote $\mathsf{DFV2}[\mathsf{R}_F, \mathsf{R}_G, \Theta\mathsf{CB2f}[\widetilde{\mathsf{P}}]]$. Note that the first element of the tweak used by $\Theta\mathsf{CB2f}[\widetilde{\mathsf{P}}]$ is $V \,\|\, 10$, hence never collides with those used by iPMAC_F and iPMAC_G. Thus $\Theta\mathsf{CB2f}[\widetilde{\mathsf{P}}]$ is independent from them, and thus

$$\mathsf{DFV2}_\Theta \equiv \mathsf{iOCB\text{-}DFV}[\widetilde{\mathsf{P}}]. \tag{16}$$

From (15), $\mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{OCB\text{-}DFV}[\mathsf{P}]}(\mathsf{A}) = \mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{iOCB\text{-}DFV}[\mathsf{XEX}[\mathsf{P}]]}(\mathsf{A})$ holds, which is bounded as

$$\mathbf{Adv}^{\mathtt{ind}}_{\mathsf{iOCB\text{-}DFV}[\mathsf{XEX}[\mathsf{P}]], \mathsf{DFV2}_\Theta}(\mathsf{A}) + \mathbf{Adv}^{\mathtt{ind}}_{\mathsf{DFV2}_\Theta, \mathsf{DFV2}^*_\Theta}(\mathsf{A}) + \mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV2}^*_\Theta}(\mathsf{A}). \tag{17}$$

[8] More precisely, a combination of XE and XEX called XEX*.

[9] We remark that iOCB-DFV is different from ΘCB-DFV in the next section.

From (16), the first term of (17) is bounded by $\mathbf{Adv}^{\mathtt{tsprp}}_{\mathsf{XEX[P]}}(\sigma)$, which is at most $9.5\sigma^2/2^n$ from the security proof of XEX [Rog04, Theorem 3]. The second term is at most $(q_e+q_d)^2/2^n$ which is derived by a simple analysis similar to the proof of PMAC [Rog04].

The last term of the above is bounded by applying Theorem 2 ;

$$\begin{aligned}\mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{DFV2}^*_\Theta}(\mathsf{A}) &\leq 2q_d\mathbf{Adv}^{\mathtt{priv}}_{\Theta\mathsf{CB2f}[\widetilde{\mathsf{P}}]}(\mathsf{A}') + q_d\mathbf{Adv}^{\mathtt{auth\text{-}1}}_{\Theta\mathsf{CB2f}[\widetilde{\mathsf{P}}]}(\mathsf{A}'') + \frac{(q_e+q_d)^2}{2^{n'+1}} + \frac{q_e^2}{2^{\nu+1}} + \frac{q_d}{2^\tau} \\ &\leq 0 + q_d\left(\frac{2}{2^n} + \frac{2}{2^\tau}\right) + \frac{(q_e+q_d)^2}{2^{n+1}} + \frac{q_e^2}{2^{n-1}} + \frac{q_d}{2^\tau}. \qquad (18)\end{aligned}$$

where the second inequality follows from the bounds of ΘCB2f [IIMP19] showing the privacy bound being 0 and the authenticity bound being $2/2^n + 2/2^\tau$ for single verification query, and the fact that $\nu = n-2$ and $n' = n$ from the specification. Combining (17) and (18) and the fact $\sigma \geq q_e + q_d$ and $\tau \leq n$, the total bound is at most

$$\frac{9.5\sigma^2}{2^n} + \frac{(q_e+q_d)^2}{2^n} + q_d\left(\frac{2}{2^n} + \frac{2}{2^\tau}\right) + \frac{(q_e+q_d)^2}{2^{n+1}} + \frac{q_e^2}{2^{n-1}} + \frac{q_d}{2^\tau} \leq \frac{13\sigma^2}{2^n} + \frac{5q_d}{2^\tau},$$

which completes the proof. □

Other block cipher-Based Instantiations. We suggest some other options for block cipher-based instantiations of DFV. In principle, any combination of a fast (ideally rate-1) NAE mode and a fast MAC mode should provide a good instantiation. If we use OTR [Min14] instead of OCB, the resulting scheme has the same rate as OCB-DFV and is inverse-free, that is, there is no need to implement the block cipher decryption function. If we consider to use DFV on constrained devices, the memory size is often a concern. In this case, a small-state MAC mode, say GCBC [Nan09], and a small-state rate-1 NAE mode, such as COFB [CIMN17]. SAEB [NS19] could be used as well. It has a smaller state than COFB, though its rate is below 1.

In choosing an NAE mode, the supported nonce length needs to be checked as it limits the achievable security. For example, the original CHES version of COFB has a $n/2$-bit nonce (thus a straightforward use of it implies $n/4$-bit security), while a version of COFB submitted to NIST Lightweight Cryptography [BCI+19] supports an n-bit nonce.

Getting Birthday Bounds. To help understanding when we can obtain a birthday-secure instantiation of DFV, let us describe the general proof idea of OCB-DFV. First we form a sequence of games that reaches to $\mathsf{DFV2}^*_\Theta$, where every intermediate game is reduced to indistinguishability of the underlying TBC built on a block cipher. Assuming the TBC is birthday-secure such as XEX, the intermediate games preserve birthday security. At the end of game sequence we evaluate MRAE advantage of $\mathsf{DFV2}^*_\Theta$ by using the generic result (Theorem 2). Since $\mathsf{DFV2}^*_\Theta$ contains a pNAE scheme that has perfect PRIV and almost perfect AUTH-1 bounds (where perfect PRIV bound is zero and perfect AUTH-1 bound is $1/2^\tau$), applying the generic result does not degrade the total security.

6.2 ΘCB-DFV : Beyond-Birthday-Bound Secure Parallel MRAE

We present another instantiation of DFV using a dedicated TBC. Our main goal is to achieve a stronger security than the $n/2$-bit security achieved by OCB-DFV, namely so-called beyond-birthday-bound (BBB) security. It is possible to use an n-bit block cipher to build a BBB-secure instance of DFV, by combing a BBB-secure PRF [Yas11] and a BBB-secure NAE [Iwa08], however, the resulting scheme is generally costly[10]. Instead, we employ

[10] When we rely on the ideal-cipher model, there are efficient block cipher modes to implement a BBB-secure TBC [Men15, WGZ+16, JLM+17]. It could be used as a primitive for ΘCB-DFV.

a dedicated TBC, such as SKINNY [BJK+16], QARMA [Ava17], and CRAFT [BLMR19], since it enables efficient BBB-secure constructions in general.

Our scheme, ΘCB-DFV, is roughly an instantiation of DFV2 using ZMAC [IMPS17a] as a PRF and ΘCB3 [KR11] as a pNAE. ZMAC+ [LN17] could be used as well. More precisely, F_K and $G_{K'}$ are instantiated by ZMAC with a simple domain separation, and pNAE is instantiated by a variant of ΘCB3, which we call ΘCBL (for Long nonce). While the original ΘCB3 has an n-bit nonce, ΘCBL has a $2n$-bit nonce to make the whole scheme n-bit secure.

Figure 9 in Appendix E shows the pseudocodes of ΘCB-DFV and ΘCBL. Also in Appendix E, Figure 12 shows ZMAC, for which we apply a minor modification on its domain separation, and set the tweak length as n (originally, $t \in [n]$).

Efficiency. The encryption rate (for message) of ΘCB-DFV is 2/3 as two blocks need one TBC call for ZMAC and 2 calls for ΘCBL. The decryption rate is 1, and ΘCB-DFV is the first TBC-based MRAE that achieves it (Table 1). It is fully parallelizable. Besides, AD is processed even faster, as two AD blocks are processed by one TBC call.

We explain the structure of ΘCB-DFV. It is based on a TBC of n-bit block and n-bit effective tweak. More specifically, the tweak space is $\mathcal{TW} = \{0,1\}^n \times \mathcal{D}$ where $\mathcal{D} = [\![15]\!]$. This setting is for simplicity, however, it is possible to modify so that $|\mathcal{TW}| \le 2^n$ by making the effective tweak length slightly shorter. Such a modification is required when we use an existing n-bit block TBC supporting an n-bit tweak. We also apply XTX tweak extension scheme [MI15] to the TBC. This requires two additional TBC calls to generate two n-bit values, which are used as masks to the block and tweak input to the TBC. It is described as follows.

Definition 3. Let $\widetilde{E} : \mathcal{K} \times \mathcal{TW} \times \mathcal{M} \to \mathcal{M}$ be a TBC with a message space $\mathcal{M} = \{0,1\}^n$ and a tweak space $\mathcal{TW} = \{0,1\}^n \times \mathcal{D}$, $\mathcal{D} = [\![15]\!]$. Our XTX turns $\widetilde{E}$ into another TBC $\mathsf{XTX}[\widetilde{E}] : \mathcal{K} \times \mathcal{TW}_{\mathsf{XTX}} \times \mathcal{M} \to \mathcal{M}$, where $\mathcal{TW}_{\mathsf{XTX}} = \{0,1\}^{2n} \times \mathcal{I} \times \mathcal{D}$ for $\mathcal{I} = [2^n]$. For a tweak $T_W = (N, i, d) \in \mathcal{TW}_{\mathsf{XTX}}$ and a plaintext $M \in \mathcal{M}$, the encryption of $\mathsf{XTX}[\widetilde{E}]$ is

$$\mathsf{XTX}[\widetilde{E}](K, M) = L \oplus \widetilde{E}_K^{W \oplus i, d}(L \oplus M),$$

where $L = \widetilde{E}_K^{N_1,0}(N_2)$ and $W = \widetilde{E}_K^{N_1,1}(N_2)$ with $N_1 = \mathtt{msb}_n(N)$ and $N_2 = \mathtt{lsb}_n(N)$. The decryption is analogously defined.

Lemma 2. *Let* $\mathsf{XTX}[\widetilde{\mathsf{P}}]$ *be as defined at Definition 3, based on a TURP* $\widetilde{\mathsf{P}} : \mathcal{TW} \times \mathcal{M} \to \mathcal{M}$*. Then we have*

$$\mathbf{Adv}^{\mathtt{tsprp}}_{\mathsf{XTX}[\widetilde{\mathsf{P}}]}(q) \le \frac{4q^2}{2^{2n}}.$$

The proof of Lemma 2 is in Appendix C.

Security Bound of ΘCB-DFV.

Theorem 5. *For* (q_e, q_d, σ, t)*-MRAE adversary* A,

$$\mathbf{Adv}^{\mathtt{mrae}}_{\Theta\mathsf{CB\text{-}DFV}[\widetilde{E}]}(\mathsf{A}) \le \mathbf{Adv}^{\mathtt{tsprp}}_{\widetilde{E}}(\mathsf{B}) + \frac{27\sigma^2}{2^{2n}} + 4\left(\frac{2q_e + q_d}{2^n}\right)^{3/2} + \frac{3q_d}{2^\tau}$$

for some $(6\sigma, \tilde{t}(\sigma))$*-TSPRP adversary* B.

Theorem 5 shows the n-bit security of ΘCB-DFV. The proof is a simple combination of Lemma 2 and the bounds of ZMAC and ΘCB3 and is shown in Appendix D.

Comparison of ΘCBL and ΘCB3. It would be worthwhile to compare ΘCBL and ΘCB3. To support n-bit nonce and the maximum input of $\approx 2^n$ blocks, ΘCB3 needs a TBC of $2n$-bit effective tweak. ΘCBL also supports these parameters with a TBC of a shorter, n-bit effective tweak. Since a TBC with a longer tweak is expected to be costlier than that of a shorter tweak, ΘCBL can be seen as an efficiency improvement of ΘCB3 in addition with a support of a longer nonce. This was possible with our XTX of Definition 3. Other tweak extension schemes employed by ZMAC and ZOCB [BGIM19] could be used as well, though they need constant multiplications over a field for every tweak update. It seems that ours is minimal to achieve our goal. To measure the concrete performance difference in practice, however, we need to consider other factors those specific to platforms or ciphers.

Other Instantiations. As well as Section 6.1, when memory size is our concern rather than the parallelizability, we could use small-state, serial alternatives to ZMAC and ΘCBL, such as Grochow *et al.* [GLN19] for PRF, and PFB [NS19] or Romulus-N [IKMP20] for pNAE. The latter would also need a proper mechanism to support $2n$-bit nonce. Thanks to the n-bit BBB-security of the components, the resulting instantiation will maintain the n-bit MRAE security as well. The actual contribution in memory reduction depends on the specification and needs a further investigation.

6.3 Permutation-Based Instantiations

We briefly discuss about possible permutation-based instantiations. For PRF, we can use a full-absorption sponge, *e.g.*, FKS [MRV15]. For (p)NAE, we can use OPP [GJMN16] or Duplex [BDPV12] or Beetle [CDNY18] or their variants. In case we compose FKS and OPP using a b-bit permutation following DFV2, the security is roughly $b/2$ bits from their bounds. If we use (serial) Duplex instead of (parallel) OPP, the resulting scheme is structurally close to a permutation-based MRAE scheme called MRS [GJMN16]. However, MRS is an instantiation of SIV, hence it does not output a tag at the end of the encryption part, and our permutation-based DFV does not need a PRF for ciphertext when decryption. Consequently, a permutation-based DFV roughly halves the decryption cost from MRS as it is the principle of DFV (the exact gain will depend on the specification).

While the composition of FKS and OPP is more or less similar to OCB-DFV, the composition of FKS and Duplex needs a dedicated, more involved analysis, in particular if we use a single key for both components. We leave the concrete specification and its analysis as a future topic.

7 Conclusions

In this article, we have described a new generic construction of MRAE. While SIV is the most popular choice for MRAE, it incurs an increased computation from the efficient rate-1 NAE schemes such as OCB. Our proposal, DFV, reduces the decryption cost to that of the rate-1 NAEs, and at the same time keeps the encryption cost. This implies a certain optimality of DFV as its total efficiency cannot be further improved. We think our work to fill the efficiency gap between MRAE and integrated NAE schemes.

Several future directions can be considered, such as studying the problem shown in Section 5, or designing variations of DFV that enjoy graceful security degradation [PS16] (which is ongoing). Software and hardware implementations of our instantiations are also interesting to evaluate their real advantage.

```
Algorithm PMAC[$E_K$](c, $A$)
    $S \leftarrow 0^n$
    $V \leftarrow E_K(\mathtt{c})$
    $(A[1], \ldots, A[a]) \xleftarrow{n} A$
    for $i = 1$ to $a - 1$
        $S \leftarrow S \oplus E_K(2^i V \oplus A[i])$
    $S \leftarrow S \oplus \mathtt{pad}(A[a])$
    if $|A[a]| = n$
        $Q \leftarrow E_K(2^a 3V \oplus S)$
    else  $Q \leftarrow E_K(2^a 3^2 V \oplus S)$
    return $Q$
```

```
Algorithm OCB-DFV[$E_K$].Enc($A, M$)
    $S \leftarrow \mathsf{PMAC}[E_K](0^n, A)$
    $V \leftarrow \mathsf{PMAC}[E_K](0^{n-1}1, M \,\|\, S)$
    $V \leftarrow \mathtt{msb}_{n-2}(V)$
    $(C, T) \leftarrow \mathsf{OCB2f}[E_K].\mathsf{Enc}(V \,\|\, 10, M)$
    $T \leftarrow T \oplus \mathtt{msb}_\tau(S)$
    return $(V, C, T)$

Algorithm OCB-DFV[$E_K$].Dec($V, A, C, T$)
    $S' \leftarrow \mathsf{PMAC}[E_K](0^n, A)$
    $(M', U') \leftarrow \mathsf{OCB2f}[E_K].\mathsf{UDec}(V, C)$
    $T' \leftarrow U' \oplus \mathtt{msb}_\tau(S')$
    if $T \neq T'$ then return $\perp$
    else return $M'$
```

Figure 8: OCB-DFV, a birthday-secure, block cipher-based DFV. OCB2f is from [IIMP19] and shown in Figure 10 (Appendix E).

Acknowledgements

The author would like to thank the anonymous referees for their insightful comments and suggestions, and Stefan Lucks for shepherding. The author also thanks Tetsu Iwata, Mustafa Khairallah and Thomas Peyrin for initial discussions.

References

[ABL$^+$14a] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. How to securely release unverified plaintext in authenticated encryption. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 105–125. Springer, Heidelberg, December 2014.

[ABL$^+$14b] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. How to securely release unverified plaintext in authenticated encryption. Cryptology ePrint Archive, Report 2014/144, 2014. http://eprint.iacr.org/2014/144.

[ADL17] Tomer Ashur, Orr Dunkelman, and Atul Luykx. Boosting authenticated encryption robustness with minimal modifications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 3–33. Springer, Heidelberg, August 2017.

[AFL$^+$16] Farzaneh Abed, Christian Forler, Eik List, Stefan Lucks, and Jakob Wenzel. RIV for robust authenticated encryption. In Thomas Peyrin, editor, *FSE 2016*, volume 9783 of *LNCS*, pages 23–42. Springer, Heidelberg, March 2016.

[Ava17] Roberto Avanzi. The QARMA block cipher family. *IACR Trans. Symm. Cryptol.*, 2017(1):4–44, 2017.

[BBLT18] Subhadeep Banik, Andrey Bogdanov, Atul Luykx, and Elmar Tischhauser. SUNDAE: Small universal deterministic authenticated encryption for the internet of things. *IACR Trans. Symm. Cryptol.*, 2018(3):1–35, 2018.

[BBP+19] Subhadeep Banik, Andrey Bogdanov, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, Elmar Tischhauser, and Yosuke Todo. SUNDAE-GIFT. Submission to NIST Lightweight Cryptography, 2019.

[BCI+19] Subhadeep Banik, Avik Chakraborti, Tetsu Iwata, Kazuhiko Minematsu, Mridul Nandi, Thomas Peyrin, Yu Sasaki adn Siang Meng Sim, and Yosuke Todo. GIFT-COFB. Submission to NIST Lightweight Cryptography, 2019.

[BDPV12] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In Ali Miri and Serge Vaudenay, editors, *SAC 2011*, volume 7118 of *LNCS*, pages 320–337. Springer, Heidelberg, August 2012.

[BGIM19] Zhenzhen Bao, Jian Guo, Tetsu Iwata, and Kazuhiko Minematsu. ZOCB and ZOTR: Tweakable blockcipher modes for authenticated encryption with full absorption. *IACR Trans. Symm. Cryptol.*, 2019(2):1–54, 2019.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 123–153. Springer, Heidelberg, August 2016.

[BLMR19] Christof Beierle, Gregor Leander, Amir Moradi, and Shahram Rasoolzadeh. CRAFT: Lightweight tweakable block cipher with efficient protection against DFA attacks. *IACR Trans. Symm. Cryptol.*, 2019(1):5–45, 2019.

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Tatsuaki Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 531–545. Springer, Heidelberg, December 2000.

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

[BNT19] Mihir Bellare, Ruth Ng, and Björn Tackmann. Nonces are noticed: AEAD revisited. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 235–265. Springer, Heidelberg, August 2019.

[BPP18] Francesco Berti, Olivier Pereira, and Thomas Peters. Reconsidering generic composition: The tag-then-encrypt case. In Debrup Chakraborty and Tetsu Iwata, editors, *INDOCRYPT 2018*, volume 11356 of *LNCS*, pages 70–90. Springer, Heidelberg, December 2018.

[BR00] Mihir Bellare and Phillip Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In Tatsuaki Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 317–330. Springer, Heidelberg, December 2000.

[BZD+16] Hanno Böck, Aaron Zauner, Sean Devlin, Juraj Somorovsky, and Philipp Jovanovic. Nonce-Disrespecting Adversaries: Practical Forgery Attacks on GCM in TLS. In *WOOT*. USENIX Association, 2016.

[CAE14] CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness, 2014. http://competitions.cr.yp.to/caesar.html/.

[CDD$^+$19] Donghoon Chang, Nilanjan Datta, Avijit Dutta, Bart Mennink, Mridul Nandi, Somitra Sanadhya, and Ferdinand Sibleyras. Release of unverified plaintext: Tight unified model and application to ANYDAE. *IACR Trans. Symmetric Cryptol.*, 2019(4):119–146, 2019.

[CDJ$^+$19] Avik Chakraborti, Nilanjan Datta, Ashwin Jha, Cuauhtemoc Mancillas Lopez, Mridul Nandi, and Yu Sasaki. ESTATE. Submission to NIST Lightweight Cryptography, 2019.

[CDN16] Avik Chakraborti, Nilanjan Datta, and Mridul Nandi. INT-RUP analysis of block-cipher based authenticated encryption schemes. In Kazue Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 39–54. Springer, Heidelberg, February / March 2016.

[CDNY18] Avik Chakraborti, Nilanjan Datta, Mridul Nandi, and Kan Yasuda. Beetle family of lightweight and secure authenticated encryption ciphers. *IACR TCHES*, 2018(2):218–241, 2018. https://tches.iacr.org/index.php/TCHES/article/view/881.

[CIMN17] Avik Chakraborti, Tetsu Iwata, Kazuhiko Minematsu, and Mridul Nandi. Blockcipher-based authenticated encryption: How small can we go? In Wieland Fischer and Naofumi Homma, editors, *CHES 2017*, volume 10529 of *LNCS*, pages 277–298. Springer, Heidelberg, September 2017.

[GJMN16] Robert Granger, Philipp Jovanovic, Bart Mennink, and Samuel Neves. Improved masking for tweakable blockciphers with applications to authenticated encryption. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 263–293. Springer, Heidelberg, May 2016.

[GL15] Shay Gueron and Yehuda Lindell. GCM-SIV: Full nonce misuse-resistant authenticated encryption at under one cycle per byte. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 2015*, pages 109–119. ACM Press, October 2015.

[GLL19] Shay Gueron, Adam Langley, and Yehuda Lindell. AES-GCM-SIV: Nonce Misuse-Resistant Authenticated Encryption. RFC 8452, 2019.

[GLN19] Tony Grochow, Eik List, and Mridul Nandi. DoveMAC: A TBC-based PRF with smaller state, full security, and high rate. *IACR Trans. Symm. Cryptol.*, 2019(3):43–80, 2019.

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 15–44. Springer, Heidelberg, April 2015.

[HP08] Helena Handschuh and Bart Preneel. Key-recovery attacks on universal hash function based MAC algorithms. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 144–161. Springer, Heidelberg, August 2008.

[HSY17] Shoichi Hirose, Yu Sasaki, and Kan Yasuda. Rate-one AE with security under RUP. In Phong Q. Nguyen and Jianying Zhou, editors, *ISC 2017*, volume 10599 of *LNCS*, pages 3–20. Springer, Heidelberg, November 2017.

[IIMP19] Akiko Inoue, Tetsu Iwata, Kazuhiko Minematsu, and Bertram Poettering. Cryptanalysis of OCB2: Attacks on authenticity and confidentiality. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 3–31. Springer, Heidelberg, August 2019.

[IKMP20] Tetsu Iwata, Mustafa Khairallah, Kazuhiko Minematsu, and Thomas Peyrin. Duel of the titans: The romulus and remus families of lightweight AEAD algorithms. *IACR Trans. Symmetric Cryptol.*, 2020, 2020. (to appear).

[IMG$^+$14] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka, and Eita Kobayashi. SILC. Submission to CAESAR competition, 2014.

[IMG$^+$17] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka, and Eita Kobayashi. SILC is INT-RUP Secure. Early Symmetric Crypto (ESC), 2017.

[IMI16] Kazuya Imamura, Kazuhiko Minematsu, and Tetsu Iwata. Integrity analysis of authenticated encryption based on stream ciphers. In Liqun Chen and Jinguang Han, editors, *ProvSec 2016*, volume 10005 of *LNCS*, pages 257–276. Springer, Heidelberg, November 2016.

[IMPS17a] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A fast tweakable block cipher mode for highly secure message authentication. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 34–65. Springer, Heidelberg, August 2017.

[IMPS17b] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A fast tweakable block cipher mode for highly secure message authentication. Cryptology ePrint Archive, Report 2017/535, 2017. http://eprint.iacr.org/2017/535.

[Iwa08] Tetsu Iwata. Authenticated encryption mode for beyond the birthday bound security. In Serge Vaudenay, editor, *AFRICACRYPT 08*, volume 5023 of *LNCS*, pages 125–142. Springer, Heidelberg, June 2008.

[JLM$^+$17] Ashwin Jha, Eik List, Kazuhiko Minematsu, Sweta Mishra, and Mridul Nandi. XHX - A framework for optimally secure tweakable block ciphers from classical block ciphers and universal hashing. In Tanja Lange and Orr Dunkelman, editors, *LATINCRYPT 2017*, volume 11368 of *LNCS*, pages 207–227. Springer, Heidelberg, September 2017.

[JNPS14] Jérémy Jean, Ivica Nikolic, Thomas Peyrin, and Yannick Seurin. Deoxys. Submission to CAESAR competition, 2014.

[Jou06] Antoine Joux. Authentication Failures in NIST Version of GCM. Comments on the Draft GCM Specification, 2006. https://csrc.nist.gov/CSRC/media/Projects/Block-Cipher-Techniques/documents/BCM/Comments/800-38-series-drafts/GCM/Joux_comments.pdf/.

[KR11] Ted Krovetz and Phillip Rogaway. The software performance of authenticated-encryption modes. In Antoine Joux, editor, *FSE 2011*, volume 6733 of *LNCS*, pages 306–327. Springer, Heidelberg, February 2011.

[Kra01] Hugo Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is SSL?). In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 310–331. Springer, Heidelberg, August 2001.

[LN17] Eik List and Mridul Nandi. ZMAC$^+$ – an efficient variable-output-length variant of ZMAC. *IACR Trans. Symm. Cryptol.*, 2017(4):306–325, 2017.

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

[Men15] Bart Mennink. Optimally secure tweakable blockciphers. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 428–448. Springer, Heidelberg, March 2015.

[MI15] Kazuhiko Minematsu and Tetsu Iwata. Tweak-length extension for tweakable blockciphers. In Jens Groth, editor, *15th IMA International Conference on Cryptography and Coding*, volume 9496 of *LNCS*, pages 77–93. Springer, Heidelberg, December 2015.

[Min14] Kazuhiko Minematsu. Parallelizable rate-1 authenticated encryption from pseudorandom functions. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 275–292. Springer, Heidelberg, May 2014.

[MRV15] Bart Mennink, Reza Reyhanitabar, and Damian Vizár. Security of full-state keyed sponge and duplex: Applications to authenticated encryption. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 465–489. Springer, Heidelberg, November / December 2015.

[MV04] David A. McGrew and John Viega. The security and performance of the Galois/counter mode (GCM) of operation. In Anne Canteaut and Kapalee Viswanathan, editors, *INDOCRYPT 2004*, volume 3348 of *LNCS*, pages 343–355. Springer, Heidelberg, December 2004.

[Nan09] Mridul Nandi. Fast and secure CBC-type MAC algorithms. In Orr Dunkelman, editor, *FSE 2009*, volume 5665 of *LNCS*, pages 375–393. Springer, Heidelberg, February 2009.

[NIS19] NIST Lightweight Cryptography Standardization, 2019. https://csrc.nist.gov/Projects/Lightweight-Cryptography.

[NL18] Yoav Nir and Adam Langley. ChaCha20 and Poly1305 for IETF Protocols. RFC 8439, 2018.

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

[NS19] Yusuke Naito and Takeshi Sugawara. Lightweight authenticated encryption mode of operation for tweakable block ciphers. *IACR TCHES*, 2020(1):66–94, 2019. https://tches.iacr.org/index.php/TCHES/article/view/8393.

[PS16] Thomas Peyrin and Yannick Seurin. Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 33–63. Springer, Heidelberg, August 2016.

[PSWZ15] Thomas Peyrin, Siang Meng Sim, Lei Wang, and Guoyan Zhang. Cryptanalysis of JAMBU. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 264–281. Springer, Heidelberg, March 2015.

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

[Rog02] Phillip Rogaway. Authenticated-encryption with associated-data. In Vijayalakshmi Atluri, editor, *ACM CCS 2002*, pages 98–107. ACM Press, November 2002.

[Rog04] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 16–31. Springer, Heidelberg, December 2004.

[RS06] Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 373–390. Springer, Heidelberg, May / June 2006.

[Sar10] Palash Sarkar. A Simple and Generic Construction of Authenticated Encryption with Associated Data. *ACM Trans. Inf. Syst. Secur.*, 13(4):33:1–33:16, 2010.

[Sar14] Palash Sarkar. Modes of operations for encryption and authentication using stream ciphers supporting an initialisation vector. *Cryptography and Communications*, 6(3):189–231, 2014.

[SW14] Yu Sasaki and Lei Wang. Message extension attack against authenticated encryptions: Application to PANDA. In Dimitris Gritzalis, Aggelos Kiayias, and Ioannis G. Askoxylakis, editors, *CANS 14*, volume 8813 of *LNCS*, pages 82–97. Springer, Heidelberg, October 2014.

[VV18] Serge Vaudenay and Damian Vizár. Can caesar beat galois? - Robustness of CAESAR candidates against nonce reusing and high data complexity attacks. In Bart Preneel and Frederik Vercauteren, editors, *ACNS 18*, volume 10892 of *LNCS*, pages 476–494. Springer, Heidelberg, July 2018.

[WC81] Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22:265–279, 1981.

[WGZ$^+$16] Lei Wang, Jian Guo, Guoyan Zhang, Jingyuan Zhao, and Dawu Gu. How to build fully secure tweakable blockciphers from classical blockciphers. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 455–483. Springer, Heidelberg, December 2016.

[Yas11] Kan Yasuda. A new variant of PMAC: Beyond the birthday bound. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 596–609. Springer, Heidelberg, August 2011.

[ZWH$^+$17] Ping Zhang, Peng Wang, Honggang Hu, Changsong Cheng, and Wenke Kuai. INT-RUP security of checksum-based authenticated encryption. In Tatsuaki Okamoto, Yong Yu, Man Ho Au, and Yannan Li, editors, *ProvSec 2017*, volume 10592 of *LNCS*, pages 147–166. Springer, Heidelberg, October 2017.

A A Detailed Comparison with SIV

Let n denote the block size, and let us assume a "standard" instantiation of SIV with an n-bit block cipher. It uses a rate-1 block cipher-based PRF (*e.g.*, CMAC) and a rate-1 block cipher-based IV-based encryption (*e.g.*, CTR mode) as originally specified in [RS06]. For a-block AD and m-block message, it needs $a+2m$ block cipher calls for both encryption and decryption, where we ignore constants. For the three variants of DFV, we assume the use of identical PRF as SIV, using the identical block cipher. For the underlying NAE scheme, we assume that it needs $a'_e + m'_e$ calls for encryption of a AD blocks and m plaintext blocks, and $a'_d + m'_d$ calls for decryption of a AD blocks and m ciphertext blocks. When AD is not present (thus pNAE), it needs m'_e calls for encryption and m'_d calls for decryption. With these settings, Table 2 shows the numbers of block cipher calls needed by SIV, DFV1, DFV2, and DFV3. This shows that, when $a'_e = a'_d = a$ and $m'_e = m'_d = m$ (as achieved by OCB), all DFV schemes have a faster decryption than SIV, and DFV2 and DFV3 keep the same encryption cost. In contrast, DFV1 has a slower encryption than SIV unless AD is empty.

Table 2: Detailed Comparison with SIV.

Scheme	Encryption	Decryption
SIV	$a + 2m$	$a + 2m$
DFV1	$a + m + a'_e + m'_e$	$a'_d + m'_d$
DFV2	$a + m + m'_e$	$a + m'_d$
DFV3	$a + m + m'_e$	$a + m'_d$

B Field Multiplication

An element a in the Galois field GF(2^n) will be interchangeably represented as an n-bit string $a_{n-1} \ldots a_1 a_0$, a formal polynomial $a_{n-1}\mathtt{x}^{n-1} + \cdots + a_1\mathtt{x} + a_0$, or an integer $\sum_{i=0}^{n-1} a_i 2^i$. Hence, by writing $2 \cdot a$ or $2a$ when no confusion is possible, we mean the multiplication of a by $2 = \mathtt{x}$. This operation is called *doubling*. The doubling is quite simple. For example, when $n = 128$, we define the field GF(2^n) (as is standard) by the primitive polynomial $\mathtt{x}^{128} + \mathtt{x}^7 + \mathtt{x}^2 + \mathtt{x} + 1$. The doubling $2a$ over this field is $(a \ll 1)$ if $\mathtt{msb}_1(a) = 0$ and $(a \ll 1) \oplus (0^{120}10000111)$ if $\mathtt{msb}_1(a) = 1$, where $(a \ll 1)$ denotes the left-shift of a by one bit. In the same manner, we define $3 \cdot a$ or $3a$ as $2 \cdot a + a$. An expression $2^i a$ means i doublings.

C Proof of Lemma 2

The proof is obtained by combining the original security proof of XTX [MI15, Theorem 1] and the following analysis on the mask generation function. Let $F_{\widetilde{\mathsf{P}}}$ be the mask generation function inside XTX[$\widetilde{\mathsf{P}}$], that is,

$$F_{\widetilde{\mathsf{P}}}(X) = (L, J),$$

where $X = (N, i, d) \in \mathcal{TW}$, $J = (W \oplus i, d)$, $L = \widetilde{\mathsf{P}}^{N_1,0}(N_2)$, $W = \widetilde{\mathsf{P}}^{N_1,1}(N_2)$, $N_1 = \mathtt{msb}_n(N)$ and $N_2 = \mathtt{lsb}_n(N)$. Here, L is given to the block input of $\widetilde{\mathsf{P}}$, and J is given to the tweak input of $\widetilde{\mathsf{P}}$.

Let $X = (N, i, d)$ and $X' = (N', i', d')$ be inputs to $F_{\widetilde{\mathsf{P}}}$, and $(L, J = (W, d)) = F_{\widetilde{\mathsf{P}}}(X)$ and $(L', J' = (W', d')) = F_{\widetilde{\mathsf{P}}}(X')$. For any $X \neq X'$, we have

$$p(X, X') := \max_{\delta \in \{0,1\}^n} \Pr_{\widetilde{\mathsf{P}}}[L \oplus L' = \delta, J = J'] \leq \frac{4}{2^{2n}} \tag{19}$$

from the following case analysis.

- **Case 1**: $N \neq N'$. We have

  $$p(X, X') \leq \Pr_{\widetilde{\mathsf{P}}}[L \oplus L' = \delta] \cdot \Pr_{\widetilde{\mathsf{P}}}[W = W'] \leq \left(\frac{1}{2^n - 1}\right)^2 \leq \frac{4}{2^{2n}},$$

  since (L, L') and (W, W') are independent (due to the domain separation) and $L \oplus L'$ and $W \oplus W'$ has the maximum point probability $1/(2^n - 1)$.

- **Case 2**: $N = N'$, $i \neq i'$. Then $W = W'$ and thus $W \oplus i \neq W' \oplus i'$ holds, therefore $p(X, X') = 0$.

- **Case 3**: $N = N'$, $i = i'$, $d \neq d'$. Then $p(X, X') = 0$ as J directly contains d.

Equation (19) implies that $F_{\widetilde{\mathsf{P}}}$ is $4/2^{2n}$-partial AXU [MI15]. Combining this fact and [MI15, Theorem 1], we conclude the proof.

D Proof of Theorem 5

We derive a bound for $\Theta\mathsf{CB\text{-}DFV}[\widetilde{\mathsf{P}}]$. The computational analogue is trivial. Let $\mathsf{i}\Theta\mathsf{CB\text{-}DFV}$ be an idealized $\Theta\mathsf{CB\text{-}DFV}[\widetilde{\mathsf{P}}]$ using a URF $\mathsf{R} : \{0,1\}^* \to \{0,1\}^{2n}$ and an idealized version of $\Theta\mathsf{CBL}$ called $\mathsf{i}\Theta\mathsf{CBL}$ (Figure 13 in Appendix E). Here, $\mathsf{i}\Theta\mathsf{CBL}$ uses a TURP $\widetilde{\mathsf{P}}_{\mathsf{XTX}} : \mathcal{TW}_{\mathsf{XTX}} \times \mathcal{M} \to \mathcal{M}$. Then we have

$$\mathbf{Adv}^{\mathtt{mrae}}_{\Theta\mathsf{CB\text{-}DFV}[\widetilde{\mathsf{P}}]}(\mathsf{A}) \leq \mathbf{Adv}^{\mathtt{prf}}_{\mathsf{ZMAC}[\widetilde{\mathsf{P}}]}(\mathsf{B}) + \mathbf{Adv}^{\mathtt{tsprp}}_{\mathsf{XTX}[\widetilde{\mathsf{P}}]}(\sigma_2) + \mathbf{Adv}^{\mathtt{mrae}}_{\mathsf{i}\Theta\mathsf{CB\text{-}DFV}}(\mathsf{A}) \tag{20}$$

for some $(2q_e + q_d, \sigma + q_e)$-PRF adversary B because an encryption of $\Theta\mathsf{CB\text{-}DFV}$ requires two calls of ZMAC, and an encryption query (N, A, M) generates two inputs to ZMAC with total n-bit length $|A|_n + |M|_n + |S|_n = |A|_n + |M|_n + 2$. From the PRF bound of ZMAC [IMPS17b, Theorem 1], the first term of the right hand side of (20) is bounded by

$$\frac{2.5(\sigma_e + q_e + \sigma_d)}{2^{2n}} + 4\left(\frac{2q_e + q_d}{2^n}\right)^{3/2} \leq \frac{10\sigma^2}{2^{2n}} + 4\left(\frac{2q_e + q_d}{2^n}\right)^{3/2}.$$

The second term of $\mathbf{Adv}^{\mathtt{tsprp}}_{\mathsf{XTX}[\widetilde{\mathsf{P}}]}(\sigma_2)$ is bounded by $16\sigma^2/2^{2n}$ from Lemma 2 with the fact that $\Theta\mathsf{CB\text{-}DFV}[\widetilde{\mathsf{P}}]$ invokes $\mathsf{XTX}[\widetilde{\mathsf{P}}]$ (for $\Theta\mathsf{CBL}$) for at most $\sigma_e + q_e + \sigma_d + q_d \leq 2\sigma$ blocks. To derive the bound of the last term, we observe that $\mathsf{i}\Theta\mathsf{CBL}$ is essentially an instance of $\Theta\mathsf{CB3}$ with $2n$-bit nonce, and thus its $\mathbf{Adv}^{\mathtt{priv}}$ bound is 0 and $\mathbf{Adv}^{\mathtt{auth\text{-}1}}$ bound is $2^{n-\tau}/(2^n - 1) < 2/2^\tau$ [KR11]. We combine this observation with Theorem 2 and that $n' = \nu = 2n$, and obtain the bound $2q_d \cdot 0 + 2q_d/2^\tau + (q_e + q_d)^2/2^{2n+1} + q_e^2/2^{2n+1} + q_d/2^\tau$, which is at most $\sigma^2/2^{2n} + 3q_d/2^\tau$. Therefore, (20) is bounded by

$$\frac{10\sigma^2}{2^{2n}} + 4\left(\frac{2q_e + q_d}{2^n}\right)^{3/2} + \frac{16\sigma^2}{2^{2n}} + \frac{\sigma^2}{2^{2n}} + \frac{3q_d}{2^\tau} \leq \frac{27\sigma^2}{2^{2n}} + 4\left(\frac{2q_e + q_d}{2^n}\right)^{3/2} + \frac{3q_d}{2^\tau}.$$

This concludes the proof.

E Left-out Figures

```
Algorithm ΘCB-DFV[$\widetilde{E}_K$].Enc($A, M$)
    $S \leftarrow$ ZMAC[$\widetilde{E}_K$]($0 \,\|\, A$)
    $V \leftarrow$ ZMAC[$\widetilde{E}_K$]($1 \,\|\, M \,\|\, S$)
    $(C, T) \leftarrow$ ΘCBL[$\widetilde{E}_K$].Enc($V, M$)
    return $(V, C, T)$
```

```
Algorithm ΘCB-DFV[$\widetilde{E}_K$].Dec($V, A, C, T$)
    $S' \leftarrow$ ZMAC[$\widetilde{E}_K$]($0 \,\|\, A$)
    $(M', U') \leftarrow$ ΘCBL[$\widetilde{E}_K$].UDec($V, C$)
    $T' \leftarrow U' \oplus \mathtt{msb}_\tau(S')$
    if $T \neq T'$ then return $\perp$
    else return $M'$
```

```
Algorithm ΘCBL[$\widetilde{E}_K$].Enc($N, M$)
    $S \leftarrow 0^n$
    $(N_1, N_2) \xleftarrow{n} N$
    $L \leftarrow \widetilde{E}_K^{N_1,0}(N_2)$
    $W \leftarrow \widetilde{E}_K^{N_1,1}(N_2)$
    $(M[1], \ldots, M[m]) \xleftarrow{n} M$
    for $i = 1$ to $m - 1$
        $C[i] \leftarrow L \oplus \widetilde{E}_K^{W \oplus i,2}(L \oplus M[i])$
        $S \leftarrow S \oplus M[i]$
    $S \leftarrow S \oplus \mathtt{pad}(M[m])$
    $Z \leftarrow L \oplus \widetilde{E}_K^{W \oplus m,3}(L)$
    $C[m] \leftarrow \mathtt{msb}_{|M[m]|}(Z) \oplus M[m]$
    $w \leftarrow 4$ if $|M[m]| \neq n$ else $5$
    $T \leftarrow L \oplus \widetilde{E}_K^{W \oplus m,w}(L \oplus S)$
    $T \leftarrow \mathtt{msb}_\tau(T)$
    return $(C, T)$
```

```
Algorithm ΘCBL[$\widetilde{E}_K$].UDec($N, C$)
    $S \leftarrow 0^n$
    $(N_1, N_2) \xleftarrow{n} N$
    $L \leftarrow \widetilde{E}_K^{N_1,0}(N_2)$
    $W \leftarrow \widetilde{E}_K^{N_1,1}(N_2)$
    $(C[1], \ldots, C[m]) \xleftarrow{n} C$
    for $i = 1$ to $m - 1$
        $M'[i] \leftarrow L \oplus (\widetilde{E}_K^{-1})^{W \oplus i,2}(L \oplus C[i])$
        $S \leftarrow S \oplus M'[i]$
    $S \leftarrow S \oplus \mathtt{pad}(M'[m])$
    $Z \leftarrow L \oplus \widetilde{E}_K^{W \oplus m,3}(L)$
    $M'[m] \leftarrow \mathtt{msb}_{|M'[m]|}(Z) \oplus C[m]$
    $w \leftarrow 4$ if $|C[m]| \neq n$ else $5$
    $U' \leftarrow L \oplus \widetilde{E}_K^{W \oplus m,w}(L \oplus S)$
    return $(M', U')$
```

```
Algorithm ΘCBL[$\widetilde{E}_K$].Dec($N, C, T$)
    $(M', U') \leftarrow$ ΘCBL[$\widetilde{E}_K$].UDec($N, C$)
    if $\mathtt{msb}_\tau(U') = T$ then return $M'$
    else return $\perp$
```

Figure 9: (Top) ΘCB-DFV, a TBC-based, full-n-bit-secure DFV. (Bottom) ΘCBL, an pNAE component of ΘCB-DFV with $2n$-bit nonce. ZMAC is in Figure 12.

```
Algorithm $\mathsf{OCB2f}[E_K].\mathsf{Enc}(N, M)$
  $L \leftarrow E_K(N)$
  $(M[1], \ldots, M[m]) \overset{n}{\leftarrow} M$
  for $i = 1$ to $m - 1$
      $C[i] \leftarrow 2^i L \oplus E_K(2^i L \oplus M[i])$
  $Z \leftarrow 2^m L \oplus E_K(2^m L \oplus \mathtt{len}(M[m]))$
  $C[m] \leftarrow M[m] \oplus \mathtt{msb}_{|M[m]|}(Z)$
  $\Sigma \leftarrow \mathtt{pad}_0(C[m]) \oplus Z$
  $\Sigma \leftarrow M[1] \oplus \cdots \oplus M[m-1] \oplus \Sigma$
  $T \leftarrow E_K(2^m 3L \oplus \Sigma)$
  $T \leftarrow \mathtt{msb}_\tau(T)$
  return $(C, T)$
```

```
Algorithm $\mathsf{OCB2f}[E_K].\mathsf{UDec}(N, C)$
  $L \leftarrow E_K(N)$
  $(C[1], \ldots, C[m]) \overset{n}{\leftarrow} C$
  for $i = 1$ to $m - 1$
      $M'[i] \leftarrow 2^i L \oplus E_K^{-1}(2^i L \oplus C[i])$
  $Z \leftarrow 2^m L \oplus E_K(2^m L \oplus \mathtt{len}(C[m]))$
  $M'[m] \leftarrow C[m] \oplus \mathtt{msb}_{|C[m]|}(Z)$
  $\Sigma \leftarrow \mathtt{pad}_0(C[m]) \oplus Z$
  $\Sigma \leftarrow M'[1] \oplus \cdots \oplus M'[m-1] \oplus \Sigma$
  $U' \leftarrow E_K(2^m 3L \oplus \Sigma)$
  return $(M', U')$
```

```
Algorithm $\Theta\mathsf{CB2f}[\widetilde{\mathsf{P}}].\mathsf{Enc}(N, M)$
  $(M[1], \ldots, M[m]) \overset{n}{\leftarrow} M$
  for $i = 1$ to $m - 1$
      $C[i] \leftarrow \widetilde{\mathsf{P}}^{N,i,0}(M[i])$
  $Z \leftarrow \widetilde{\mathsf{P}}^{N,m,0}(\mathtt{len}(M[m]))$
  $C[m] \leftarrow M[m] \oplus \mathtt{msb}_{|M[m]|}(Z)$
  $\Sigma \leftarrow \mathtt{pad}_0(C[m]) \oplus Z$
  $\Sigma \leftarrow M[1] \oplus \cdots \oplus M[m-1] \oplus \Sigma$
  $T \leftarrow \widetilde{\mathsf{P}}^{N,m,1}(\Sigma)$
  $T \leftarrow \mathtt{msb}_\tau(T)$
  return $(C, T)$
```

```
Algorithm $\Theta\mathsf{CB2f}[\widetilde{\mathsf{P}}].\mathsf{UDec}(N, C)$
  $(C[1], \ldots, C[m]) \overset{n}{\leftarrow} C$
  for $i = 1$ to $m - 1$
      $M'[i] \leftarrow (\widetilde{\mathsf{P}}^{-1})^{N,i,0}(C[i])$
  $Z \leftarrow \widetilde{\mathsf{P}}^{N,m,0}(\mathtt{len}(C[m]))$
  $M'[m] \leftarrow C[m] \oplus \mathtt{msb}_{|C[m]|}(Z)$
  $\Sigma \leftarrow \mathtt{pad}_0(C[m]) \oplus Z$
  $\Sigma \leftarrow M'[1] \oplus \cdots \oplus M'[m-1] \oplus \Sigma$
  $U' \leftarrow \widetilde{\mathsf{P}}^{N,m,1}(\Sigma)$
  return $(M', U')$
```

Figure 10: (Top) Encryption and Unfinished Decryption of $\mathsf{OCB2f}$ for empty AD. $\mathtt{len} : \{0,1\}^{\leq n} \to \{0,1\}^n$ is an injective length-encoding function. (Bottom) TURP-based idealization of $\mathsf{OCB2f}$, $\Theta\mathsf{CB2f}$.

```
Algorithm $\mathsf{iPMAC}[\widetilde{\mathsf{P}}](\mathsf{c}, A)$
  $S \leftarrow 0^n$
  $(A[1], \ldots, A[a]) \overset{n}{\leftarrow} A$
  for $i = 1$ to $a - 1$
      $S \leftarrow S \oplus \widetilde{\mathsf{P}}^{\mathsf{c},i,0}(A[i])$
  $S \leftarrow S \oplus \mathtt{pad}(A[a])$
  if $|A[a]| = n$
      $Q \leftarrow \widetilde{\mathsf{P}}^{\mathsf{c},a,1}(S)$
  else $Q \leftarrow \widetilde{\mathsf{P}}^{\mathsf{c},a,2}(S)$
  return $Q$
```

```
Algorithm $\mathsf{iOCB\text{-}DFV.Enc}[\widetilde{\mathsf{P}}](A, M)$
  $S \leftarrow \mathsf{iPMAC}[\widetilde{\mathsf{P}}](0^n, A)$
  $V \leftarrow \mathsf{iPMAC}[\widetilde{\mathsf{P}}](0^{n-1}1, M \,\|\, S)$
  $V \leftarrow \mathtt{msb}_{n-2}(V)$
  $(C, T) \leftarrow \Theta\mathsf{CB2f}[\widetilde{\mathsf{P}}].\mathsf{Enc}(V \,\|\, 10, M)$
  $T \leftarrow T \oplus \mathtt{msb}_\tau(S)$
  return $(V, C, T)$
```

```
Algorithm $\mathsf{iOCB\text{-}DFV}[\widetilde{\mathsf{P}}].\mathsf{Dec}(V, A, C, T)$
  $S' \leftarrow \mathsf{iPMAC}[\widetilde{\mathsf{P}}](0^n, A)$
  $(M', U') \leftarrow \Theta\mathsf{CB2f}[\widetilde{\mathsf{P}}].\mathsf{UDec}(V \,\|\, 10, C)$
  $T' \leftarrow U' \oplus \mathtt{msb}_\tau(S')$
  if $T \neq T'$ then return $\perp$
  else return $M'$
```

Figure 11: TURP-based idealization of $\mathsf{OCB\text{-}DFV}[\mathsf{P}]$, $\mathsf{iOCB\text{-}DFV}[\widetilde{\mathsf{P}}]$.

```
**Algorithm** ZHASH$[\widetilde{E}_K](X)$
  $U \leftarrow 0^n$, $V \leftarrow 0^n$
  $L_\ell \leftarrow \widetilde{E}_K^{0^n,6}(0^n)$
  $L_r \leftarrow \widetilde{E}_K^{0^{n-1}1,6}(0^n)$
  $(X[1], \ldots, X[m]) \xleftarrow{2n} X$
  **for** $i = 1$ **to** $m$ **do**
    $X_\ell \leftarrow \mathtt{msb}_n(X[i])$
    $X_r \leftarrow \mathtt{lsb}_n(X[i])$
    $S_\ell \leftarrow L_\ell \oplus X_\ell$
    $S_r \leftarrow L_r \oplus X_r$
    $C_\ell \leftarrow \widetilde{E}_K^{S_r,7}(S_\ell)$
    $C_r \leftarrow C_\ell \oplus X_r$
    $U \leftarrow 2(U \oplus C_\ell)$
    $V \leftarrow V \oplus C_r$
    $(L_\ell, L_r) \leftarrow (2L_\ell, 2L_r)$
  **return** $(U, V)$
```

```
**Algorithm** ZFIN$[\widetilde{E}_K](i, U, V)$
  $Y[1] \leftarrow \widetilde{E}_K^{V,i}(U) \oplus \widetilde{E}_K^{V,i+1}(U)$
  $Y[2] \leftarrow \widetilde{E}_K^{V,i+2}(U) \oplus \widetilde{E}_K^{V,i+3}(U)$
  $Y \leftarrow Y[1] \,\|\, Y[2]$
  **return** $Y$
```

```
**Algorithm** ZMAC$[\widetilde{E}_K](M)$
  $X \leftarrow \mathtt{pad}_{\mathtt{zmac}}(M)$
  $(U, V) \leftarrow$ ZHASH$[\widetilde{E}_K](X)$
  **if** $|M| \bmod 2n = 0$ **and** $|M| > 0$
    $Y \leftarrow$ ZFIN$[\widetilde{E}_K](8, U, V)$
  **else**
    $Y \leftarrow$ ZFIN$[\widetilde{E}_K](12, U, V)$
  **return** $Y$
```

Figure 12: Specification of ZMAC, with a minor modification on domain separation from the original. The padding function $\mathtt{pad}_{\mathtt{zmac}}$ is a non-injective padding defined over $2n$-bit blocks: $\mathtt{pad}_{\mathtt{zmac}}(X) = X$ when $|X|$ is a positive multiple of $2n$ and $\mathtt{pad}_{\mathtt{zmac}}(X) = X \,\|\, 10^{2n-|X|-1}$ otherwise.

```
Algorithm $\mathsf{i\Theta CB\text{-}DFV.Enc}(A, M)$
    $S \leftarrow \mathsf{R}(0 \,\|\, A)$
    $V \leftarrow \mathsf{R}(1 \,\|\, M \,\|\, S)$
    $(C, T) \leftarrow \mathsf{i\Theta CBL}[\widetilde{\mathsf{P}}_{\mathsf{XTX}}].\mathsf{Enc}(V, M)$
    return $(V, C, T)$
```

```
Algorithm $\mathsf{i\Theta CB\text{-}DFV.Dec}(V, A, C, T)$
    $S' \leftarrow \mathsf{R}(0 \,\|\, A)$
    $(M', U') \leftarrow \mathsf{i\Theta CBL}[\widetilde{\mathsf{P}}_{\mathsf{XTX}}].\mathsf{UDec}(V, C)$
    $T' \leftarrow U' \oplus \mathtt{msb}_{\tau}(S')$
    if $T \neq T'$ then return $\perp$
    else return $M'$
```

```
Algorithm $\mathsf{i\Theta CBL}[\widetilde{\mathsf{P}}_{\mathsf{XTX}}].\mathsf{Enc}(N, M)$
    $S \leftarrow 0^n$
    $(M[1], \ldots, M[m]) \xleftarrow{n} M$
    for $i = 1$ to $m - 1$
        $C[i] \leftarrow \widetilde{\mathsf{P}}_{\mathsf{XTX}}^{N,i,2}(M[i])$
        $S \leftarrow S \oplus M[i]$
    $S \leftarrow S \oplus \mathtt{pad}(M[m])$
    $Z \leftarrow \widetilde{\mathsf{P}}_{\mathsf{XTX}}^{N,m,3}(0^n)$
    $C[m] \leftarrow \mathtt{msb}_{|M[m]|}(Z) \oplus M[m]$
    $w \leftarrow 4$ if $|M[m]| \neq n$ else $5$
    $T \leftarrow \widetilde{\mathsf{P}}_{\mathsf{XTX}}^{N,m,w}(S)$
    $T \leftarrow \mathtt{msb}_{\tau}(T)$
    return $(C, T)$
```

```
Algorithm $\mathsf{i\Theta CBL}[\widetilde{\mathsf{P}}_{\mathsf{XTX}}].\mathsf{UDec}(N, C)$
    $S \leftarrow 0^n$
    $(C[1], \ldots, C[m]) \xleftarrow{n} C$
    for $i = 1$ to $m - 1$
        $M'[i] \leftarrow (\widetilde{\mathsf{P}}_{\mathsf{XTX}}^{-1})^{N,i,2}(C[i])$
        $S \leftarrow S \oplus M'[i]$
    $S \leftarrow S \oplus \mathtt{pad}(M'[m])$
    $Z \leftarrow \widetilde{\mathsf{P}}_{\mathsf{XTX}}^{N,m,3}(0^n)$
    $M'[m] \leftarrow \mathtt{msb}_{|C[m]|}(Z) \oplus C[m]$
    $w \leftarrow 4$ if $|C[m]| \neq n$ else $5$
    $U' \leftarrow \widetilde{\mathsf{P}}_{\mathsf{XTX}}^{N,m,w}(S)$
    return $(M', U')$
```

```
Algorithm $\mathsf{i\Theta CBL}[\widetilde{\mathsf{P}}_{\mathsf{XTX}}].\mathsf{Dec}(N, C, T)$
    $(M', U') \leftarrow \mathsf{i\Theta CBL}[\widetilde{\mathsf{P}}_{\mathsf{XTX}}].\mathsf{UDec}(N, C)$
    if $\mathtt{msb}_{\tau}(U') \neq T$ then return $\perp$
    else return $M'$
```

Figure 13: (Top) $\mathsf{i\Theta CB\text{-}DFV}$ using a URF $\mathsf{R} : \{0,1\}^* \to \{0,1\}^{2n}$ and an idealized variant of $\mathsf{\Theta CBL}$, $\mathsf{i\Theta CBL}$. (Bottom) $\mathsf{i\Theta CBL}$. It is based on a TURP $\widetilde{\mathsf{P}}_{\mathsf{XTX}}$ of $3n$-bit effective tweak.