DNDK:
Combining Nonce and Key Derivation for Fast and Scalable AEAD

Shay Gueron
University of Haifa and Meta

Thomas Ristenpart
Cornell Tech

Abstract

Authenticated encryption with associated data (AEAD) schemes are responsible for securing increasingly critical digital infrastructure. Unfortunately, current widely deployed schemes suffer from various limitations that make them difficult to use securely in practice. Popular schemes like AES-GCM limit the amount of data that can be encrypted with a single key, preventing secure scaling to modern workloads. At the same time, practitioners may not be able to move away from the use of AES-GCM due to mature and widely deployed implementations, legacy constraints, and compliance.

In this paper, we provide approaches to improve the secure scaling of AEAD schemes via what we call derived-nonce, derived-key (DNDK) transforms. At a high level, such transforms use a root key to derive a nonce and key for use with an underlying scheme. The challenge is doing so in a way that introduces as little overhead as possible, while relying on a small number of assumptions on the used primitives. We provide some general results about secure scaling transforms and a concrete design for AES-GCM that is called DNDK-GCM. It requires as little as three additional AES calls to enable use of the same key to encrypt up to 2^{64} bytes of data, even when using random nonces. We also provide a detailed performance analysis. DNDK-GCM is now a draft IETF standard, and is already deployed at cloud scale by companies including Meta.

1 Introduction

Symmetric encryption is the workhorse of modern cryptography, responsible for protecting the majority of web traffic, data stored at rest in databases, laptops, mobile devices, and much more. Today, it is mainly realized via authenticated encryption with associated data (AEAD) schemes. Arguably the most important example is AES-GCM [47], whose high performance, security, and widespread implementation support (both in hardware and software) has made it one of the most widely used cryptographic schemes.

However, recent results and cloud-scale workloads illustrate some challenges with using the current generation of AEAD schemes:

(1) The scale of modern encryption-using systems exceeds current design capacities.

(2) Damaging key commitment attacks affect the most widely used AEAD schemes.

We discuss these challenges below.

Encryption at scale. AEAD encryption takes as input a key, nonce, associated data, and message. The fastest schemes, such as AES-GCM, OCB [44, 55], and ChaCha20/Poly1305 [52], provide security only when encryption never reuses a nonce—also referred to as an IV—with the same key. Reusing a nonce leads to compromise of both confidentiality and integrity. Applications can enforce nonce uniqueness by constructing nonces with a counter. However, in many real-world scenarios, maintaining state to track counters is costly and prone to errors.

System designers generally prefer to generate a random nonce for every message under the assumption that the probability of nonce collisions during the lifetime of the key is sufficiently small. In this context, NIST's Special Publication 800-38D [25, Section 8] requires that: "The probability that the authenticated encryption function will ever be invoked with the same IV and the key on two (or more) distinct sets of input data shall be no greater than 2^{-32}." Here AES-GCM is limited because for random nonces security after q encryption calls is upper bounded by $q^2/2^{97}$; this bound is also essentially tight. Satisfying the NIST security requirement therefore limits the lifetime of a key to encrypting at most $2^{32.5}$ messages.

Modern workloads can require encrypting such an amount of data within just a few seconds [41], imposing a too-frequent key rotation rate. This complicates implementations, raising their expense to build and maintain, while also raising the likelihood of security problems due to implementation bugs. For further discussion by cryptographic engineers about these complexities, see, e.g., [15, 30, 41].

A second scaling challenge is that, independently of the

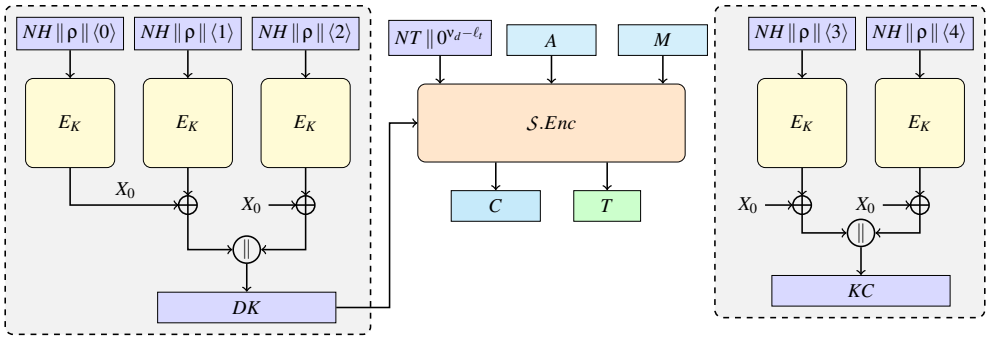

Figure 1: Diagram of DNDKv2 encryption using using a block cipher E and AEAD $\mathcal{S}$, and using a configuration specified by ρ, a 5-bit parameter string indicating whether to use key commitment and which of 16 different nonce lengths to use. Upon input root key K, nonce $NH \| NT$, associated data A, and message M, DNDKv2 generates a derived key DK, applies $\mathcal{S}$'s encryption using DK and the nonce tail NT appropriately padded, and outputs a ciphertext and tag pair. Here $\|$ denotes string concatenation and $\langle x \rangle$ denotes encoding an integer $x < 5$ as a 3-bit string. DNDKv2 supports an optional key commitment KC (box on the right). Decryption re-derives DK, executes $\mathcal{S}$'s decryption and, when using key commitment, re-computes KC to check it.

collision rate in random nonces, the fact that AES-GCM uses a 128-bit block cipher (AES) enforces a limit on the maximum number of encrypted blocks under a given key. Due to the birthday bound, an adversary that makes q encryption queries with total length at most σ blocks has distinguishing advantage at most $(\sigma + q + 1)^2/2^{129}$ [38]. Keeping this advantage below 2^{-32}, implies, at least, a limit of $2^{48.5}$ blocks (2^{52} bytes) under a given key.

Commitment issues. A second issue with existing schemes is that they are not key committing: an adversary can efficiently construct a nonce N, associated data A, ciphertext and tag pair (C,T), and keys K,K' such that $Dec(K,N,A,C,T)$ and $Dec(K',N,A,C,T)$ both succeed. This was first observed for AES-GCM in [27], and its impact on security demonstrated by a sequence of results showing how to exploit lack of commitment in various important settings [1, 23, 45]. Subsequent work explored a strengthening of key commitment to a notion now called context commitment [8].

The consensus is now that general-purpose schemes should provide at least key commitment, and ideally, context commitment (c.f., [5, 7, 9, 19, 40, 41, 45, 46]).

Transform-based approaches. One approach to the above problems is clean-slate: build new cryptographic schemes that achieve secure scaling and commitment (e.g., [20, 48]). But legacy, compliance, and performance constraints will require continued use of AES-GCM for the foreseeable future.

Another approach is to seek lightweight transforms that convert an existing AEAD into one that achieves better secure scaling, as first explored for AES-GCM by Gueron and Lindell [33], standardized in the form of AES-GCM-SIV [32], analyzed subsequently in several works [14, 39], and used in a variety of practical settings [16]. These do not target, nor provide, key commitment. We similarly target lightweight transforms, building off this line of work.

Our results. We explore a family of transforms that we call derived-nonce derived-key (DNDK) transforms. One instance of our proposed transform, called DNDK-GCM, extends the security of AES256-GCM. See Fig. 1. DNDK-GCM is already a draft RFC [31] and in use in production at Meta.

While prior transforms have focused on using a key derivation function applied to a root key and nonce to generate a one-time-use derived key [33], we instead propose generating both a derived nonce and derived key, hence the name, to achieve secure scaling with very lightweight derivation. Our preferred transform builds off Iwata's XORP [36] to build a derivation function that requires as few as three AES calls (beyond any used by the underlying AEAD). We also show how one can easily extend our derivation function to provide a commitment to root secret key at the cost of just two additional AES calls. We establish security via a proof that XORP is key committing, a result that may be useful elsewhere. We caveat that we do not provide context commitment, necessitating additional defenses (c.f., [8, 17]) when adversaries can force decryption under an adversarially known key and adversarially chosen nonces or associated data.

DNDK-GCM is fast. We report on timings from an already open source implementation in the already released Haberdashery [35] library. Tiny messages incur the highest, but still modest, overhead: $\sim 3\times$ over plain AES256-GCM. For moderate size messages of 16 KB the overhead is less than $1.02\times$ AES256-GCM, and for larger 1 MB messages $1.0005\times$. See Sec. 9 for performance details.

We provide formal analyses of DNDK transforms' confidentiality and authenticity that reduce to the security of the underlying derivation function and AEAD. These carefully account for adversarial resources and, in particular, how the pattern of nonces used in DNDK's encryption translate to the pattern of keys used with the underlying AEAD. We show how to combine these analyses with standard-model multi-

user security bounds for AES-GCM that provide good bounds by taking advantage of the limitations on that pattern of key use. Ultimately this gives full standard-model reductions; we also give corollaries in the ideal cipher model in order to enable deriving security for all parameters.

Our main proofs of security for DNDK rely on existing analyses or on standard techniques, and assume only standard assumptions about AES. We view this as a benefit of our approach, since it should aid in verifying proofs and, in turn, improved confidence in the security of DNDK-GCM, which is already being used to protect sensitive information in practice.

History and impact. DNDK has evolved over time, with two versions developed in public with engagement with the community via talks by the first author [28, 30] and incorporation of feedback. Version one used a different KDF from version two. The first version is more expensive and not modularly analyzable as a (strong) PRF, as pointed out to the first author by Degabriele [21]. This issue does not seem to lead to an attack. Version two (DNDKv2, our focus here) is faster and enjoys a modular security analysis as shown in the body. DNDKv2 is now used at scale at Meta as the default encryption method. Moreover, it can be a drop-in replacement for ad hoc nonce extension implementations (such as WhatsApp's Labyrinth message storage protocol [49]).

DNDK has already had academic impact as well, motivating a number of follow-up works including [11, 13, 42, 58], as we discuss in Sec. 8.

This is an updated version of this work, which was first posted in Spring of 2025. Besides editorial changes, this version provides alternative proofs for XORP and CTR key commitment, initial analyses for configuration commitment, and includes more discussion about DNDK and alternative instantiations.

2 Preliminaries

Our schemes will often be parameterized by a block length n; most often we focus on $n = 128$ which matches AES. For a bit string X, its size in n-bit blocks is $\lceil |X|/n \rceil$. We denote sampling uniformly from a set S by $X \leftarrow\!\$\ S$. For two strings X, Y we denote their concatenation as $X \,\|\, Y$. We let $\{0,1\}^\ell$ be the set of all bit strings of length ℓ bits, and 0^ℓ be the bit string of ℓ zero bits. For sets Y_1, Y_2, Dom let $F : Dom \to Y_1 \times Y_2$ be a function that outputs pairs. We define the right projection $F[2]$ of F as $F[2](X) = y_2$ if and only if $F(X) = (y_1, y_2)$ for some $y_1 \in Y_1$.

Block cipher security. A block cipher $E : \{0,1\}^{k_d} \times \{0,1\}^n \to \{0,1\}^n$ for key length k_d and block length n is a family of permutations, one per key. The mu-PRP1_E game starts by running an adversary $\mathcal{A}$, which can make queries to a key generation oracle and an enciphering oracle. The former takes no input, increments a key instance counter μ, and generates a new key instance $K_\mu \leftarrow\!\$\ \{0,1\}^{k_d}$. The encipher oracle takes a key instance identifier $i \le \mu$ and an input $X \in \{0,1\}^n$ and returns $E(K_i, X)$. The adversary eventually halts with output a single bit, which is also the games output. Game mu-PRP0_E is the same except that key generation picks a new random permutation $\pi_\mu \leftarrow\!\$\ \mathrm{Perm}(n)$ where $\mathrm{Perm}(n)$ is the set of all permutations on n bits. Also, enciphering queries (i, X) return $\pi_i(X)$. The advantage of an mu-PRP$_E$ adversary is

$$\mathsf{Adv}^{\text{mu-prp}}_E(\mathcal{B}) = \\ \left|\Pr\left[\text{mu-PRP1}_E(\mathcal{A}) \Rightarrow 1\right] - \Pr\left[\text{mu-PRP0}_E(\mathcal{A}) \Rightarrow 1\right]\right|$$

where the events denote running the game with adversary $\mathcal{A}$ and that game outputting one.

An ideal cipher IC is a family of random permutations, one per key input. We denote sampling an ideal cipher via $IC \leftarrow\!\$\ \mathrm{Perm}(k_d, n)$. Here $\mathrm{Perm}(k_d, n)$ is the set of all functions $\{0,1\}^{k_d} \times \{0,1\}^n \to \{0,1\}^n$ that are permutations for fixed first input. We write IC^{-1} to denote the inverse of IC, i.e., $IC^{-1}(K, IC(K, X)) = X$. In the ideal cipher model, we replace E with a randomly sampled IC in the games above, and give the adversary oracle access to both IC and IC^{-1} on keys of their choosing. The following was shown in [50].

Theorem 1. *Let $E : \{0,1\}^{k_d} \times \{0,1\}^n \to \{0,1\}^n$ be an ideal cipher and $\mathcal{A}$ be an* mu-PRP$_E$ *adversary making at most p queries to IC or IC^{-1}, at most κ key generation queries, and at most q enciphering queries. Then*

$$\mathsf{Adv}^{\text{mu-prp}}_E(\mathcal{A}) \le \frac{\kappa \cdot p}{2^{k_d}} + \frac{\kappa(\kappa - 1)}{2^{k_d}}$$

If $\mathcal{A}$ makes a single key generation query, the bound is $p/2^{k_d}$.

AEAD schemes. An AEAD scheme $\mathcal{S} = (Kg, Enc, Dec)$ is a triple of algorithms. Key generation $\mathcal{S}.Kg$ takes no input and outputs a secret key K of length $\kappa \ge 0$ bits. Encryption $\mathcal{S}.Enc(K, N, A, M)$ takes as input a key K, nonce N, associated data A, and message M and outputs a ciphertext, tag pair (C, T). We let $\mathcal{S}.\tau$ be the tag length of the scheme, assuming that $|T| = \mathcal{S}.\tau$ always. Decryption $\mathcal{S}.Dec(K, N, A, C, T)$ takes as input a key, nonce, associated data, ciphertext, and tag, and outputs a message or distinguished error symbol $\perp$. Associated data and messages are bit strings of some length up to maximums that we will specify later for particular schemes. We will require that our AEAD schemes be tidy [51]. We will consider schemes implicitly parameterized by a block length n (due to underlying use of a block cipher).

A relevant example scheme is AES-GCM, which we define for the case of nonces being 96 bits. Key generation chooses a key K uniformly from $\{0,1\}^k$ for $k = 128$ for AES128-GCM and $k = 256$ for AES256-GCM. We will use AES-GCM when we consider either key length, and otherwise specify the length. Encryption derives a hash key $L = E_K(0^n)$, and encryption proceeds by computing $C = M \oplus P$ for $P =$

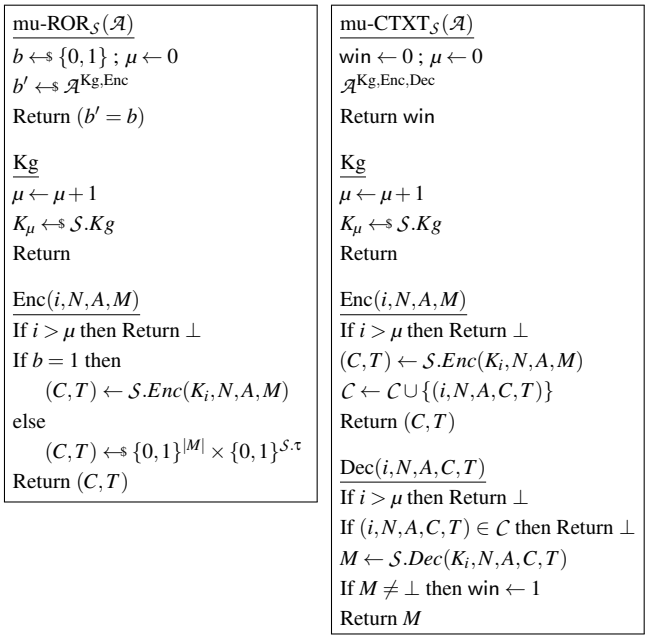

Figure 2: Multi-user AEAD security games for **(left)** confidentiality and **(right)** ciphertext integrity.

$E_K(N \| \langle 2 \rangle) \| \cdots \| E_K(N \| \langle \lceil |M|/128 \rceil + 2 \rangle)$ where counters are encoded in 32 bits. Thus $|M|$ in blocks can be at most $2^{32} - 2$. Then the tag is computed as $T = \mathrm{GHASH}(L,C) \oplus E_K(N \| \langle 0 \rangle)$. The output is (C,T). See [47] for more details.

We formalize AEAD security as two separate properties, multi-user confidentiality and authenticity (ciphertext integrity). This will allow more precise bounds in each attack scenario. Standard results show how these imply all-in-one security [56]. Games are provided in Fig. 2. We let mu-ROR$0_{\mathcal{S}}$ (resp. mu-ROR$1_{\mathcal{S}}$) be the mu-ROR$_{\mathcal{S}}$ game but setting $b = 0$ (resp. $b = 1$). The advantage of an mu-ROR$_{\mathcal{S}}$ adversary is defined as

$$\mathsf{Adv}^{\text{mu-ror}}_{\mathcal{S}}(\mathcal{A}) = \big|\Pr[\text{mu-ROR1}_{\mathcal{S}}(\mathcal{A}) \Rightarrow 1] - \Pr[\text{mu-ROR0}_{\mathcal{S}}(\mathcal{A}) \Rightarrow 0]\big| ,$$

where the probability is defined over the coins of the games including those used by $\mathcal{A}$. We define the advantage of an mu-CTXT$_{\mathcal{S}}$ adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\text{mu-ctxt}}_{\mathcal{S}}(\mathcal{A}) = \Pr[\text{mu-CTXT}_{\mathcal{S}}(\mathcal{A}) \Rightarrow 1] .$$

We use the notation ROR$_{\mathcal{S}}$ and CTXT$_{\mathcal{S}}$ to refer to the mu-ROR$_{\mathcal{S}}$ and mu-CTXT$_{\mathcal{S}}$ games restricted to a single key generation query. We let $\mathsf{Adv}^{\text{ror}}_{\mathcal{S}}$ and $\mathsf{Adv}^{\text{ctxt}}_{\mathcal{S}}$ be the associated advantage measures, defined like the equations above.

We assume that adversaries are nonce-respecting: they do not query encryption on the same nonce twice. We also assume that CTXT and mu-CTXT adversaries do not make pointless queries: a decryption query (i,N,A,C,T) for which a previous encryption query (i,N,A,M) returned (C,T).

DNDK construction			
k	root key length	k_d	derived key length
ℓ_h	nonce head length	ν_d	derived nonce length
ℓ_t	nonce tail length	ℓ_{kc}	key commitment length
ν	root nonce length ($\nu = \ell_h + \ell_t$)	n	block length

Adversarial resources	
κ	number of key generation queries
q	number of encryption queries
v	number of decryption queries
L	max plaintext length $\lvert M\rvert$ or ciphertext length $\lvert C\rvert$ in blocks
L_A	max input length ($\lvert A\rvert + \lvert M\rvert$) or ($\lvert A\rvert + \lvert C\rvert$) in blocks
$\vec{q}$-NH	max $\vec{q}_i$ nonce heads queried i times
$\vec{q}$-bound	max of i encryption queries to each of $\vec{q}_i$ key instances
$w(\vec{q})$	highest index i such that $\vec{q}_i > 0$

Figure 3: Variables used throughout the paper.

We also consider security for uniform random nonces. Let ROR$\$_{\mathcal{S}}$ be the same as ROR$_{\mathcal{S}}$ except that the encryption oracle only takes as input the pair A,M, and it selects a nonce $N \leftarrow\$ \{0,1\}^{\nu}$, computes (C,T) as before using N (in the $b = 1$ case), and then returns (N,C,T). Similarly CTXT$\$_{\mathcal{S}}$ has the same modifications to encryption queries compared to CTXT$_{\mathcal{S}}$. We define ROR$\$1_{\mathcal{S}}$ and ROR$\$0_{\mathcal{S}}$ by setting $b = 1$ and $b = 0$. We have the following advantage measures:

$$\mathsf{Adv}^{\text{ror\$}}_{\mathcal{S}}(\mathcal{A}) = |\Pr[\text{ROR\$1}_{\mathcal{S}}(\mathcal{A}) \Rightarrow 1] - \Pr[\text{ROR\$0}_{\mathcal{S}}(\mathcal{A}) \Rightarrow 1]|$$
$$\mathsf{Adv}^{\text{ctxt\$}}_{\mathcal{S}}(\mathcal{A}) = \Pr[\text{CTXT\$}_{\mathcal{S}}(\mathcal{A}) \Rightarrow 1] .$$

We provide some bounds on AEAD security in the ideal cipher model. This means that, in the relevant game, scheme algorithms can use oracle access to an ideal cipher (as defined above) and the adversary can also query the same ideal cipher.

Adversarial resources. As we are interested in security in practice, we follow a concrete security approach that measures adversarial advantage as a function of the adversary's resources in terms of computational time and number of oracle queries. We summarize notation for adversarial resources used throughout the paper in the bottom table of Fig. 3.

In general we will be concerned with the number of key generation queries κ, the number of encryption queries q, and the number of decryption queries v (for CTXT definitions). The maximum plaintext length L is the maximum length, in n-bit blocks, of any encryption query's plaintext M or any decryption query's ciphertext C (not including the tag). The maximum input length L_A is the maximum length, in blocks, of any encryption query's combined associated data and plaintext length ($|A| + |M|$) or any decryption query's combined associated data and ciphertext length ($|A| + |C|$). We will later define other adversarial resource constraints (the last two rows), and defer discussion of this notation to Sec. 5.

3 Overview

Our focus is transforms that allow use of existing AEAD schemes at larger scales, and with additional desirable properties such as key commitment. As background, it is well understood that our most widely used AEAD schemes suffer from secure scaling challenges.

Secure scaling. As an important case, consider the secure scaling of $\mathcal{S}$ being AES-GCM. A number of scaling hurdles have been observed in the past. First, security is maximized by using 96-bit nonces, as longer or shorter nonces are processed using GHASH in a way that degrades security [38,53]. For 96-bit random nonces, an adversary making q queries succeeds in breaking confidentiality with probability 2^{-32} after $q > 2^{32}$ queries. This is too small for modern workloads (c.f., [41]), forcing frequent re-keying.

One can instead use stateful nonces such as a counter to extend the lifetime of a key. This complicates deployment since all systems using a secret key to perform encryptions must maintain shared state (the counter). Even when possible to use counters, secure scaling is still limited by the fact that security at some point fails due to internal collisions on an input to the underlying block cipher. Assuming an adversary that makes at most q encryption queries of length at most σ blocks, then security only holds up to $(\sigma+q+1)^2/2^{129}$ [38]. Again, security fails should one encrypt 2^{64} bytes of data.

Derived-key transforms. Prior works have explored transforms for improving the secure scaling of an existing AEAD scheme. This line of work was initiated by Gueron and Lindell [33] who proposed encryption via:

$$\mathcal{S}.Enc(KDF(K,N),N,A,M)$$

where KDF is a key derivation function that must be secure as a PRF when assuming standard security properties of $\mathcal{S}$. They also provided a particular AES-based construction of KDF. This led to an RFC for AES-GCM-SIV, a nonce-misuse-resistant mode [32]. Fig. 4 (top left) also summarizes this transform. Follow-up work by Iwata and Seurin [39] pointed out problems in the original proofs and provided new proofs with weaker bounds. Hoang, Tessaro, and Thiruvengadam [34] later showed an alternate analysis in the multi-user setting that provides better bounds.

The Gueron-Lindell approach to scaling is limited to the size of nonces used by AES-GCM; they explicitly restrict to 96-bit nonces and while this may suffice for stateful nonces such as counters, it does not work well for random nonces that will collide too frequently with complete failure of security. (AES-GCM-SIV, being nonce-misuse-resistant does not fail completely in this case.)

One could instead use larger nonces, such as $|N| = 192$. AES-GCM supports such nonces, hashing them down to a 96-bit effective nonce size. But this approach would require a KDF function that accepts larger inputs, such as HMAC [4] or HKDF [43], which would add significantly to the computational overhead.

$DK \leftarrow KDF(K,N)$
$(C,T) \leftarrow \mathcal{S}.Enc(DK,N,A,M)$

$DK \leftarrow KDF(K,N)$
$(C,T) \leftarrow \mathcal{S}.Enc(DK,0^{\nu},A,M)$

$(DK,DN) \leftarrow KDF(K,N)$
$(C,T) \leftarrow \mathcal{S}.Enc(DK,DN,A,M)$

$NH \,\|\, NT \leftarrow \mathsf{split}_{\ell_h}(N)$
$(DK,DN) \leftarrow KDF(K,NH)$
$(C,T) \leftarrow \mathcal{S}.Enc(DK,NT,A,M)$

Figure 4: A variety of secure scaling transforms built from a KDF KDF (that supports domains and ranges implied by its use context) and an underlying AEAD $\mathcal{S}$. The bottom, right box is our derived-nonce, derived-key transform (DNDK).

Double-nonce, derived-key transforms. In order to support larger nonces while retaining ability to use small-block-length primitives (i.e., AES), we can use a beyond-birthday-bound (BBB) PRF as first explored by Bellare, Krovetz, and Rogaway [10]. This leads to what we call a double-nonce, derived-key construction:

$$\mathcal{S}.Enc(KDF(K,N),0^{\nu_d},A,M)$$

This is the approach used in the first version of the DNDK RFC, where KDF is instantiated by XORing the output of two applications of Iwata's XORP BBB PRF [36], one to each half of N. This double XORP construction can be proven secure as a weak PRF using Iwata's analysis combined with [6], though that does not suffice for formal security analyses that require a strong PRF (as ours require). It also uses at least six AES calls.

Derived-nonce, derived-key transforms. Note the double-nonce derived-key transform uses a fixed constant as nonce in encryption. We can improve efficiency and simplify analysis by instead using a derived nonce. For example by computing $(DN,DK) \leftarrow KDF(K,N)$ and then encrypting via

$$\mathcal{S}.Enc(DK,DN,A,M)\ .$$

We abuse notation here for brevity, assuming KDF outputs the contextually requisite bit strings. We refer to this as a derived-nonce, derived-key transform. Intuitively security will hold as long as the combination of DK and DN are unique. More precisely, DK of sufficient length will rarely repeat because it is pseudorandom, and thus the underlying $\mathcal{S}$ need only provide strong multi-key security bounds when each key is used a handful of times. For efficiency we would like to instantiate KDF with AES, but this is difficult because $|N| \geq 192$.

We instead take a different approach, observing that one can simplify derived-nonce generation by just reusing part of the input nonce. See Fig. 4, bottom right, where we use a function split_{ℓ_h} that splits the input into two parts with the first part having length ℓ_h. In words: take the input nonce N and split it into two parts: NH and NT. We have some parameters ℓ_h and ℓ_t that dictate the split sizes meaning that $|N| = \nu = \ell_h + \ell_t$

and $|NH| = \ell_h$, $|NT| = \ell_t$, and $N = NH \| NT$. As such we refer to NH as the nonce head and NT as the nonce tail.

Then we apply a KDF only to NH (and the key K) to generate a derived key DK that is used with NT to encrypt the message:

$$\mathcal{S}.Enc(KDF(K,NH),NT,A,M) .$$

In the draft RFC version 2 this is referred to as double-nonce, derived-key since it targets $\nu = 2 \cdot 96 = 192$.

A question will be how to choose the split. If NH is too short, we'll have high probability of repeating use of DK. Instead, we want to maximize its length, subject to the constraint that we want to support implementing KDF using AES. Hence our preferred instantiation sets $|NH| = \ell_h = 120$, which leaves one byte for encoding parameters and necessary domain separation when using AES. In theory we can actually set $\ell_h = 126$ (when not using key commitment; see discussion in the next section), but this is awkward in implementations that work in bytes.

4 DNDK Transforms

We now turn to formalizing our transform approach. We start by defining the derivation function, built from a PRF. Because we want to support the optional use of key commitment, we introduce an abstraction for a PRF that includes an explicit key commitment. We then build a KDF from one.

PRFs with explicit key commitment. Unlike standard CR-PRFs that are both collision-resistant across all inputs as well as secure as a PRF, our functions have two outputs and require that both look pseudorandom when the key is held secret, and that the second output is a commitment to the key. Formally, an explicitly committing PRF (ecPRF) is a function

$$F\colon \{0,1\}^k \times \{0,1\}^m \to \left(\{0,1\}^{k_d} \times \{0,1\}^{\ell_{kc}}\right)$$

mapping a secret key of k bits and message of length m bits to an output of k_d bits and a commitment value of ℓ_{kc} bits. For simplicity, we use as length parameters those related to our transform's derivation function below (also refer back to Fig. 3), and m will be set to fit the nonce head plus configuration encoding.

Looking ahead, key commitments will be optional. One may opt not to use them because they add overhead both in terms of computational cost and increasing ciphertext length, or because a different commitment technique is used (e.g., [8, 17]). Thus we allow $\ell_{kc} = 0$, and in this case F outputs a single bit string of length k_d.

We define two security goals for ecPRFs: PRF security and key commitment. Recall that game PRF1_F chooses a key $K \xleftarrow{\$} \{0,1\}^k$, runs an adversary $\mathcal{D}$, giving it access to an oracle that returns $F(K,X)$, and outputs the adversary's output bit. Game PRF0_F is the same except that the oracle instead returns a random function applied to X. Then advantage is defined by

$$\mathsf{Adv}^{\text{prf}}_F(\mathcal{D}) = |\Pr[\text{PRF1}_F(\mathcal{D}) \Rightarrow 1] - \Pr[\text{PRF0}_F(\mathcal{D}) \Rightarrow 1]| .$$

We define key commitment security for a function F via the game k-CMT_F that runs an adversary $\mathcal{C}$ that outputs a triple (K,K',X). The game outputs one if $F(K,X) = F(K',X)$ and $K \neq K'$. Notice that this is not the same as CR since we restrict to the same input X; CR would allow finding any $(K,X) \neq (K',X')$ that collide. Key commitment advantage is defined by

$$\mathsf{Adv}^{\text{k-cmt}}_F(\mathcal{C}) = \Pr[\text{k-CMT}_F(\mathcal{C}) \Rightarrow 1] .$$

For an explicitly committing PRF F that outputs pairs, we will be interested in key commitment security for the projection of F to its right output. Recall that we use the notation $F[2]$ to denote this.

In the ideal cipher model, the games above are augmented so that F, $\mathcal{D}$, and $\mathcal{C}$ all have oracle access to randomly sampled IC, IC^{-1}.

Construction from XORP. We can build a secure F in a variety of ways. A simple one would be to use a standard CR-PRF H such as HMAC-SHA256, running it twice with suitable domain separation, once to obtain the first output and again to obtain the second output (the key commitment). Our goal, however, is to build an F that is not only secure but also highly efficient while retaining sufficient simplicity for implementations. Furthermore, we wish the overall security of the schemes to rely on as few primitives and assumptions as possible, so a PRF construction that is built on a block cipher (specifically AES) is a promising choice.

Fig. 6 provides an efficient alternative that is faster than using constructions like HMAC. It is based on Iwata's beyond-the-birthday bound XORP construction [36], which utilizes an underlying block cipher $E\colon \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ with block length n. Let w,m be such that $n = m + \lceil \log_2(w+1) \rceil$. Then $\text{XORP}\colon \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^{wn}$ is defined by

$$\text{XORP}(K,X) = (E_K(X_0) \oplus E_K(X_1)) \,\|\, \cdots \,\|\, (E_K(X_0) \oplus E_K(X_w))$$

where $X_i = X \,\|\, \langle i \rangle$ and the encoding of i uses $\lceil \log_2 w + 1 \rceil$ bits. We use XORP with $w = 2$ if $\ell_{kc} = 0$ and XORP with $w = 4$ if $\ell_{kc} = 2n$.

$\underline{F(K,X)}$
$Y_0 \leftarrow E_K(X \,\|\, \langle 0 \rangle)$
$Y_1 \leftarrow Y_0 \oplus E_K(X \,\|\, \langle 1 \rangle)$
$Y_2 \leftarrow Y_0 \oplus E_K(X \,\|\, \langle 2 \rangle)$
If $\ell_{kc} = 2n$ then
  $Y_3 \leftarrow Y_0 \oplus E_K(X \,\|\, \langle 3 \rangle)$
  $Y_4 \leftarrow Y_0 \oplus E_K(X \,\|\, \langle 4 \rangle)$
  Return $(Y_1 \,\|\, Y_2, Y_3 \,\|\, Y_4)$
Return $Y_1 \,\|\, Y_2$

Figure 5: An explicitly committing PRF F for $\ell_{kc} \in \{0, 2n\}$ built from XORP.

PRF security. The PRF security of XORP is well-established. We restate the theorem as stated by Iwata, Mennink, and Vizár [37] which builds off work by Patarin [54].

Theorem 2 (XORP is a PRF; [37], Theorem 2)**.** *Let E be a block cipher with block length n, fix w,m such that $n = m + \lceil \log_2(w+1) \rceil$, and let $F\colon \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^{wn}$ be the XORP construction defined above. Let $\mathcal{D}$ be an* $\mathrm{PRF}_{\mathrm{XORP}}$ *adversary making at most $q \le 2^n/(67w)$ queries to the* F *oracle for a single key. Then there exists an explicit* PRP_E *adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\mathrm{prf}}_F(\mathcal{D}) \le \frac{w^2 q}{2^n} + \mathsf{Adv}^{\mathrm{prp}}_E(\mathcal{B})$$

Adversary $\mathcal{B}$ makes at most $(w+1)\cdot q$ queries and runs in time close to that of $\mathcal{D}$.

In the case that $\ell_{kc} = 0$, the above bound becomes $4q/2^n$. In the case that $\ell_{kc} = 2n$ it is $16q/2^n$.

Key commitment security. We now turn to analyzing key commitment security, which restricts attention to the $\ell_{kc} = 2n$ case. We note that $F[2]$ is equivalent to XORP with $w = 2$ just with different counter values. Thus key commitment security of $F[2]$ reduces to key commitment security of XORP for $w = 2$. As such, we show key commitment security of XORP, which may be of independent interest.

Then any k-$\mathrm{CMT}_{\mathrm{XORP}}$ adversary must solve the following computational problem: find K, K', X with $K \neq K'$ such that

$$E_K(X_0) \oplus E_K(X_1) = E_{K'}(X_0) \oplus E_{K'}(X_1)$$
$$E_K(X_0) \oplus E_K(X_2) = E_{K'}(X_0) \oplus E_{K'}(X_2)$$

where $X_i = X \,\|\, \langle i \rangle$. Recall that Albertini et al. [1] analyzed the key commitment security of CTR mode using a block cipher E, which is the same as the above except that the terms related to X_0 are dropped in the two equations. Their proof has a minor bug and uses a syntactically different notion of commitment security from k-CMT. For completeness, we therefore provide an alternative analysis of CTR-based KDFs in App. D.

It may seem that the X_0 terms complicate analysis compared to CTR, for example the structure of the equations above might suggest that Wagner's generalized birthday algorithm [59] could be applied. But because X_0, X_1, X_2 are simultaneously fixed by the same X value, one can't mix-and-match different K', X pairs and this does not represent an instance of the 4-list birthday problem. Instead, we can prove that XORP is key committing with increasingly good security as w grows. To gain some intuition, ignore the permutivity of E, and the analysis then corresponds to the probability that sampling two random functions map a sequence of w of (the same) inputs to the same sequence of n-bit outputs. The proof of the next theorem is given in App. E.

Theorem 3 (XORP is key committing)**.** *Let E be a block cipher with block length $n \ge 7$, fix w,m such that $n = m + \lceil \log_2(w+1) \rceil$ and $w < 0.5\sqrt{2^n}$, and let* XORP$\colon \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^{wn}$ *be the construction defined above. Then for any* k-$\mathrm{CMT}_{\mathrm{XORP}}$ *adversary $\mathcal{C}$ making at most p ideal cipher queries it holds that*

$$\mathsf{Adv}^{\text{k-cmt}}_{\mathrm{XORP}}(\mathcal{C}) \le \frac{p(p-1)}{2^{wn}} .$$

We only need this theorem for $w = 2$, in which case we get that security holds up to adversaries performing close to 2^{128} work. Moreover, this bound is tight (up to a factor of w) since an adversary can perform a birthday attack by fixing an X, and making $p = (w+1)2^\kappa$ queries with 2^κ different keys to compute XORP for each key. The adversary will succeed in finding a collision with probability at least $0.3 \cdot 2^\kappa \cdot (2^\kappa - 1)/2^{wn}$ by a lower bound on the birthday problem [26].

Looking ahead to our use within DNDK, the key-commitment security established above only applies when the same nonce and configuration is used (which together constitute X). We can actually give a stronger result about XORP that can be used to prove that DNDK is both key and configuration committing: see App. F.

DNDK derivation functions. A key component of our approach is derivation functions that generate derived nonces and derived keys from a root key, root nonce, and parameter string.[1] The context/parameter string specifies a number of values used by the derivation function:

- the secret root key length $k = |K|$,
- the input nonce head length ℓ_h,
- the input nonce tail length ℓ_t,
- the input nonce length $\nu = \ell_h + \ell_t$,
- the derived key length k_d,
- the derived nonce length ν_d, and
- the key commitment length ℓ_{kc}.

Hereafter, we capture the choice of parameters in the notation ρ. Looking ahead, the specification for DNDK as per the RFC actually fixes several of the parameters, allowing only $\nu = 8 \cdot i$ for some $12 \le i \le 27$ and whether to use a key commitment or not. Thus ρ can be encoded in five bits and we slightly abuse notation using ρ to denote also its encoding.

The DNDK version 2 derivation function, denoted *DNDKv2*, is defined in Fig. 6. It utilizes an underlying ecPRF F. It splits the nonce N into NH and NT. It then applies F to the root key, nonce head NH, and parameters to generate a derived key DK and key commitment. Finally, the derivation function outputs the nonce tail (with zero-padding as needed) as the derived nonce, along with the derived key and the key commitment.

We include an encoding of ρ because it binds key commitments to them. This may be useful in settings where implementations can somehow be asked to use the wrong

[1] The parameter string is referred to as the context in the RFC.

$\underline{DNDK.Enc(K,\rho,N,A,M)}$
If $Chk_\rho(K,N,A,M) = 0$ then
  Return $\perp$
$(DN,DK,KC) \leftarrow KDF(\rho,K,N)$
$(C,T) \leftarrow \mathcal{S}.Enc(DK,DN,A,M)$
Return (KC,C,T)

$\underline{DNDK.Dec(K,\rho,N,A,KC,C,T)}$
If $Chk_\rho(K,N,A,KC,C,T) = 0$ then
  Return $\perp$
$(DN,DK,KC') \leftarrow KDF(\rho,K,N)$
If $KC \neq KC'$ then Return $\perp$
$M \leftarrow \mathcal{S}.Dec(DK,DN,A,C,T)$
Return M

$\underline{KDF0(\rho,K,N)}$
$(DN \,\|\, DK, KC) \leftarrow F(K, \rho \,\|\, N)$
Return (DN,DK,KC)

$\underline{DNDKv2(\rho,K,N)}$
$NH \leftarrow N[1..\ell_h]$
$NT \leftarrow N[\ell_h+1..\mathrm{v}]$
$(DK,KC) \leftarrow F(K, NH \,\|\, \rho)$
$DN \leftarrow NT \,\|\, 0^{\mathrm{v}_d - \ell_t}$
Return (DN,DK,KC)

Figure 6: **(Left)** The *DNDK* transform combining a function *KDF* with an underlying AEAD $\mathcal{S}$. When key commitments are omitted ($|KC| = 0$) the $KC = KC'$ check always passes. **(Right)** Two derivation functions. The top is a generic ecPRF applied to the entire input. The second corresponds to the double nonce, derived key RFC version 2.

parameters to decrypt some ciphertext: decryption will fail unless ρ is correct. As discussed in Sec. 3, we will in general prefer the largest ℓ_h possible subject to constraints on encoding the parameters and counter needed by XORP within an AES 128-bit block.

DNDK transforms. Let *KDF* be a function that takes as input a root key K, a nonce N, and a parameter string ρ. It outputs a derived nonce DN, a derived key DK, and an optional key commitment value KC. When not using key commitments, $|KC| = 0$.

A DNDK transform $DNDK[KDF,\mathcal{S}]$ combines *KDF* with an underlying AEAD $\mathcal{S}$. We say that *KDF* and $\mathcal{S}$ are compatible if *KDF* outputs DN and DK values lying in the space of allowed nonces and keys for $\mathcal{S}$. The pseudocode for encryption and decryption are shown in the left-hand-side of Fig. 6. Unless specified otherwise we will have DNDK transforms use as root key generation the underlying AEAD's key generation algorithm, i.e., $DNDK[KDF,\mathcal{S}].Kg = \mathcal{S}.Kg$.

5 Confidentiality Analysis

We start by analyzing the confidentiality of DNDK using *DNDKv2* and a compatible AEAD $\mathcal{S}$, and when key commitments are omitted ($\ell_{kc} = 0$). We discuss confidentiality when $\ell_{kc} > 0$ in Sec. 7. We'll show in particular full bounds for DNDK-GCM where F is XORP and $\mathcal{S}$ is AES256-GCM.

Respectful and bounded adversaries. We first analyze single-user ROR security for adversaries that make limited use of repeated nonce heads. We have the following granular notion of what we call nonce head respect, which captures limits on the pattern of nonce-head reuse.

Definition 4. *An* $\mathrm{ROR}_{\mathcal{S}}$ *adversary is* $\vec{q}$*-NH respecting if: (1) it is nonce respecting, and (2) its encryption queries have at most* $\vec{q}_i$ *distinct nonce heads appearing in exactly i queries.*

Note that the definition is absolute, in the sense that we require that $\mathcal{A}$ abides by the nonce head restrictions regardless of any random choices made when executing within a game.

The maximum dimension of such a vector $\vec{q}$ is the maximum number of queries made by the adversary; we will often be more interested in the highest index that has a non-zero element, i.e., $w(\vec{q})$. For example, an adversary that is $\vec{q}$*-NH* respecting for $\vec{q} = (10,12,0,\ldots,0)$ makes at most 10 queries with unique nonce heads, and 24 queries with twice-repeated nonce heads. It has $w(\vec{q}) = 2$.

For such adversaries, we then show how to reduce $\mathrm{ROR}_{\mathcal{T}}$ security to the security of F and the mu-ROR security of $\mathcal{S}$, but the latter only needs to hold for adversaries that make restricted use of their key instances.

Definition 5. *An* $\text{mu-ROR}_{\mathcal{S}}$ *adversary* $\mathcal{A}$ *is* $\vec{q}$*-bound if it makes at most i encryption queries to each of* $\vec{q}_i$ *key instances.*

Continuing our example of $\vec{q} = (10,12,0,\ldots,0)$ above, we have that a $\vec{q}$-bound adversary can make up to one query to each of 10 different key instances and two queries to each of 12 different key instances. It has $w(\vec{q}) = 2$. The total number of key instances is at most $\kappa = \sum_i \vec{q}_i = 22$ and the total number of encryption queries is at most $q = \sum_i i \cdot \vec{q}_i = 34$.

Below we will use the following result about AES-GCM. Its proof proceeds by first reducing security to the multi-user security of the underlying block cipher E, followed by a hybrid argument applying the single-user bound from [38, Cor. 3] repeatedly. See App. G for a proof.

Theorem 6. *Let* $\mathcal{S}$ *be AES-GCM used with 96-bit nonces built from an n-bit block cipher E. Let* $\mathcal{A}$ *be a* $\vec{q}$*-bound* $\text{mu-ROR}_{\mathcal{S}}$ *adversary making queries each of whose maximum length is L. Then there exists an explicit* mu-PRP_E *adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\text{mu-ror}}_{\mathcal{S}}(\mathcal{A}) \leq \mathsf{Adv}^{\text{mu-prp}}_{E}(\mathcal{B}) + \sum_{i=1} \frac{\vec{q}_i \cdot (iL+i+1)^2}{2^{129}} \ .$$

Adversary $\mathcal{B}$ *runs in time that of* $\mathcal{A}$*, makes at most* $\kappa = \sum_i \vec{q}_i$ *key generation queries and at most* $\sum_i \vec{q}_i \cdot i \cdot (L+2)$ *enciphering queries.*

Nonce-head-respecting adversaries. We have the following theorem which uses the security of F to enable us to upper bound the advantage of any $\vec{q}$*-NH*-respecting $\mathrm{ROR}_{\mathcal{T}}$ adversary by the advantage of a $\vec{q}$-bound $\text{mu-ROR}_{\mathcal{S}}$ adversary.

Theorem 7 (DNDK confidentiality—nonce-head-respecting case)**.** *Let* $\mathcal{S}$ *be an AEAD scheme, F be an ec-PRF, and* $\mathcal{T} = DNDK[DNDKv2[F],\mathcal{S}]$ *for a fixed parameter set* ρ*. Let* $\mathcal{A}$ *be a* $\vec{q}$*-NH-respecting* $\mathrm{ROR}_{\mathcal{T}}$ *adversary whose queries have*

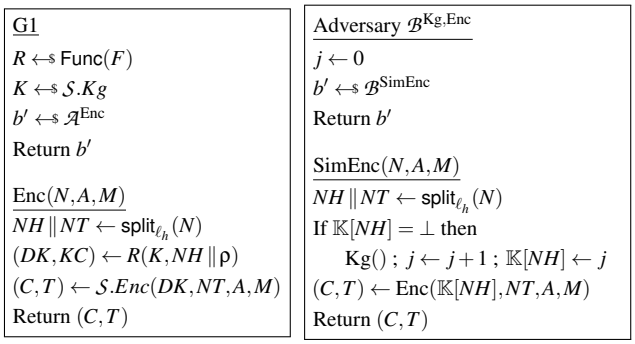

Figure 7: Game and adversary used in the proof of Thm. 7.

maximum plaintext length L blocks. Then we give a PRF_F *adversary* $\mathcal{D}$ *and* mu-$\mathrm{ROR}_{\mathcal{S}}$ *adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathrm{ror}}_{\mathcal{T}}(\mathcal{A}) \le \mathsf{Adv}^{\mathrm{prf}}_{F}(\mathcal{D}) + \mathsf{Adv}^{\text{mu-ror}}_{\mathcal{S}}(\mathcal{B}) \ .$$

$\mathcal{D}$ *makes at most* $q = \sum_i \vec{q}_i$ *queries. Adversary* $\mathcal{B}$ *is* $\vec{q}$*-bound and its queries have maximum plaintext length L blocks.*

Proof: We use a standard game-hopping argument. Let game $\mathrm{G0} = \mathrm{ROR1}_{\mathcal{T}}(\mathcal{A})$. Then game G1 is the same as G0 except that we replace all calls to F with calls to a random function with the same domain and range as F. A standard reduction gives an PRF_F adversary $\mathcal{D}$ such that

$$\Pr[\mathrm{G0} \Rightarrow 1] \le \Pr[\mathrm{G1} \Rightarrow 1] + \mathsf{Adv}^{\mathrm{prf}}_{F}(\mathcal{D})$$

where $\mathcal{D}$ runs G0, replacing all executions of F with calls it its oracle. Hence $\mathcal{D}$ runs in time that of $\mathcal{A}$ and makes at most q queries. Note that despite $\mathcal{A}$ being nonce-respecting, it may be that NH repeats across queries.

Game G2 is the same as G1 except that we replace all ciphertexts and tags (C,T) with random bit strings of the appropriate length. Then we construct an mu-$\mathrm{ROR}_{\mathcal{S}}$-adversary $\mathcal{B}$. Adversary $\mathcal{B}$ runs G1 except that any execution of $\mathcal{S}.Enc$ is replaced by a call to $\mathcal{B}$'s oracle, with an appropriately indicated key (see Fig. 7). Then by construction we have that $\Pr[\mathrm{G1} \Rightarrow 1] = \Pr[\text{mu-}\mathrm{ROR1}_{\mathcal{S}}(\mathcal{B}) \Rightarrow 1]$ and $\Pr[\mathrm{G2} \Rightarrow 1] = \Pr[\text{mu-}\mathrm{ROR0}_{\mathcal{S}}(\mathcal{B}) \Rightarrow 1]$, and so

$$\Pr[\mathrm{G1} \Rightarrow 1] \le \mathsf{Adv}^{\text{mu-ror}}_{\mathcal{S}}(\mathcal{B}) + \Pr[\mathrm{G2} \Rightarrow 1] \ .$$

Noting that $\mathrm{G2} = \text{mu-}\mathrm{ROR0}_{\mathcal{S}}$, and combining with the equations above gives the desired upper bound on $\mathcal{A}$'s advantage.

Finally, if $\mathcal{A}$ is $\vec{q}$-NH-respecting then $\mathcal{B}$ is $\vec{q}$-bound: table $\mathbb{K}$ maps each nonce head to a particular user key. □

The above theorem works for an arbitrary underlying AEAD, showing how its security under limitations on how often keys are reused follows from restrictions we have on nonce selection. We apply it to $\mathcal{S}$ being AES-GCM, giving us the following security bound in the ideal cipher model for adversaries that never repeat a nonce head (i.e., are $\vec{q}$-NH respecting with $w(\vec{q}) = 1$). We use the ideal cipher only to provide fully specified bounds including PRP security of underlying AES256 usage.

Corollary 8 (DNDK-GCM confidentiality—nonce-head-respecting case)**.** *Let* F *be XORP and* $\mathcal{S}$ *be AES-GCM with both built from an ideal cipher* $E : \{0,1\}^{k_d} \times \{0,1\}^n \to \{0,1\}^n$*. Let* $\mathcal{T} = DNDK[DNDKv2[F],\mathcal{S}]$ *for a fixed parameter set* ρ*. Let* $\mathcal{A}$ *be a* mu-$\mathrm{ROR}_{\mathcal{T}}$ *adversary that makes at most p primitive queries, at most q encryption queries of maximum length L blocks, and never repeats a nonce head. Then*

$$\mathsf{Adv}^{\mathrm{ror}}_{\mathcal{T}}(\mathcal{A}) \le \frac{(q+1)p + q(q-1)}{2^{k_d}} + \frac{4q}{2^n} + \frac{q(L+2)^2}{2^n} \ .$$

The proof is essentially the same as the proof of Cor. 10 below, except simpler because we do not need to take into account collisions on nonces or nonce heads. We therefore omit it for brevity.

Assuming we use AES256 with $k_d = 256$, the dominating term above will be the final term even for very large p (e.g., $p = 2^{128}$). The final term is maximized by maximizing L, which is at most $L = 2^{32} - 2$ (the maximum number of blocks for AES-GCM). Thus for a budget of encrypting 2^{60} blocks (to give 2^{64} bytes), this allows $q = 2^{28}$ queries of length L. The final advantage upper bound is in this case 2^{-36}.

DNDK with random nonces. We now turn to the likely primary use case for DNDK, which is with randomized nonces. To utilize the bounds above, we first show that the random-nonce confidentiality $\mathrm{ROR\$}_{\mathcal{T}}$ is upper bound by the advantage of an adversary against mu-$\mathrm{ROR}_{\mathcal{T}}$ for which nonce heads are either unique across all queries or repeat at most twice. The proof of the following appears in App. H.

Theorem 9 (DNDK confidentiality—random nonce case)**.** *Let* $\mathcal{T} = DNDK[DNDKv2[F],\mathcal{S}]$ *for a fixed parameter set* ρ *and PRF* F *and AEAD* $\mathcal{S}$*. Let* $\mathcal{A}$ *be a* $\mathrm{ROR\$}_{\mathcal{T}}$ *adversary making at most q encryption queries each of maximum plaintext length L blocks. Then we give a* $\mathrm{ROR}_{\mathcal{T}}$ *adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathrm{ror\$}}_{\mathcal{T}}(\mathcal{A}) \le \mathsf{Adv}^{\mathrm{ror}}_{\mathcal{T}}(\mathcal{B}) + \frac{q^2}{2^{\mathrm{v}}} + \frac{q^3}{3 \cdot 2^{2\ell_h}} \ .$$

Adversary $\mathcal{B}$ *is* $\vec{q}$*-NH-respecting for some* $\vec{q}$ *with* $w(\vec{q}) \le 2$ *and* $q \ge \sum_i i \cdot \vec{q}_i$*. The maximum plaintext length queried by* $\mathcal{B}$ *is L blocks.*

We can now prove the following corollary, which shows complete bounds for DNDK using GCM in the ideal cipher model. As before, we use the ideal cipher only to provide full bounds: the only place we use the ideal cipher model is to bound the PRP security of the underlying AES256 usage.

Corollary 10 (DNDK-GCM confidentiality—random nonce case)**.** *Let* F *be XORP and* $\mathcal{S}$ *be AES-GCM with both built from an ideal cipher* $E : \{0,1\}^{k_d} \times \{0,1\}^n \to \{0,1\}^n$*. Let* $\mathcal{T} = DNDK[DNDKv2[F],\mathcal{S}]$ *for a fixed parameter set* ρ*. Let* $\mathcal{A}$ *be a* $\mathrm{ROR\$}_{\mathcal{T}}$ *adversary that makes at most p primitive queries and at most q encryption queries of maximum*

length L blocks. Then

$$\mathsf{Adv}^{\mathrm{ror\$}}_{\mathcal{T}}(\mathcal{A}) \le \frac{(q+1)p+q(q-1)}{2^{k_d}} + \frac{4q}{2^n} + \frac{q(L+2)^2}{2^n} + \frac{q^2}{2^{\mathrm{v}}} + \frac{q^3}{3\cdot 2^{2\ell_h}} .$$

Proof: Applying Thm. 9 we obtain:

$$\mathsf{Adv}^{\mathrm{ror\$}}_{\mathcal{T}}(\mathcal{A}) \le \mathsf{Adv}^{\mathrm{ror}}_{\mathcal{T}}(\mathcal{B}_1) + \frac{q^2}{2^{\mathrm{v}}} + \frac{q^3}{3\cdot 2^{2\ell_h}}$$

for an adversary $\mathcal{B}_1$ that is $\vec{q}$-*NH* respecting for $w(\vec{q}) = 2$ and $\vec{q}_1 + 2\cdot\vec{q}_2 \le q$. That is, it never queries a nonce head more than twice. Then we can apply Thm. 7 to get that

$$\mathsf{Adv}^{\mathrm{ror}}_{\mathcal{T}}(\mathcal{B}_1) \le \mathsf{Adv}^{\mathrm{prf}}_{F}(\mathcal{D}) + \mathsf{Adv}^{\text{mu-ror}}_{\mathcal{S}}(\mathcal{B}_2)$$

for adversary $\mathcal{D}$ making at most $\sum_i \vec{q}$ queries (one for each distinct nonce head queried) and $\mathcal{B}_2$ that is $\vec{q}$-bound and makes at most q queries each of length at most L blocks.[2] Finally we can apply the security of XORP (Thm. 2) and AES-GCM for $\vec{q}$-bounded adversaries (Thm. 6) to get that $\mathsf{Adv}^{\mathrm{prf}}_{F}(\mathcal{D}) \le \frac{4q}{2^n} + \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{C}_1)$ for the adversary $\mathcal{C}_1$ specified in the proof of Thm. 2 and

$$\begin{aligned}\mathsf{Adv}^{\text{mu-ror}}_{\mathcal{S}}(\mathcal{B}_2) &\le \mathsf{Adv}^{\text{mu-prp}}_{E}(\mathcal{C}_2) + \frac{\vec{q}_1(L+2)^2}{2^{n+1}} + \frac{\vec{q}_2(2L+3)^2}{2^{n+1}} \\ &< \mathsf{Adv}^{\text{mu-prp}}_{E}(\mathcal{C}_2) + \frac{q(L+2)^2}{2^n}\end{aligned}$$

for the adversary $\mathcal{C}_2$ specified in the proof of Thm. 6. The second inequality is derived as follows:

$$\begin{aligned}\vec{q}_1(L+2)^2 + \vec{q}_2(2L+3)^2 &< \vec{q}_1(L+2)^2 + \vec{q}_2(2L+4)^2 \\ &= (\vec{q}_1 + 2\vec{q}_2 + 2\vec{q}_2)(L+2)^2 \\ &\le 2q(L+2)^2\end{aligned}$$

where we use $\vec{q}_1 + 2\vec{q}_2 \le q$. Finally we apply the ideal cipher model security of E using Thm. 1 to get that

$$\mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{C}_1) \le \frac{p}{2^{k_d}} \quad \text{and} \quad \mathsf{Adv}^{\text{mu-prp}}_{E}(\mathcal{C}_2) \le \frac{qp}{2^{k_d}} + \frac{q(q-1)}{2^{k_d}}$$

□

6 Authenticity Analysis

We now turn to analyzing the ciphertext integrity of DNDK using *DNDKv2*, for $\ell_{kc} \in \{0, 2n\}$. As with confidentiality, our analyses are broken down into two cases: nonce-head-respecting adversaries that make a bounded amount of repeat use of nonce heads, and adversaries against the scheme when used with random nonces.

We can translate the notions of nonce-head respect and boundedness to the CTXT setting: the definitions are the same as in the previous section except for the type of adversary considered. In the CTXT setting there are also decryption queries, and for these the adversary can query nonce heads (resp. key instances) as they choose. Similarly, for mu-CTXT$_{\mathcal{S}}$ a $\vec{q}$-bound adversary may make a number κ of key generation queries such that $\kappa > \sum_i \vec{q}_i$ and decryption queries may be targeted at arbitrary adversarially chosen key instances. For example, all v decryption queries made by an adversary could be for a single key instance.

A multi-user GCM CTXT bound. The following theorem establishes a multi-user security bound for GCM in the case of $\vec{q}$-bounded CTXT adversaries. At a high level, it proceeds via a hybrid argument over key instances, using a result in [38, Appendix C] that relies on an analysis technique first due to Bernstein [12]. For the following theorem let $\delta_n(a) = (1 - (a-1)/2^n)^{-a/2}$. See App. I for a proof.

Theorem 11. *Let $\mathcal{S}$ be AES-GCM used with 96-bit nonces and τ-bit tags built from an n-bit block cipher E. Let $\mathcal{A}$ be a $\vec{q}$-bound* mu-CTXT$_{\mathcal{S}}$ *adversary making at most κ key generation queries, encryption queries with maximum length L blocks, and at most v decryption queries with maximum plaintext length L and maximum input length L_A blocks. Then there exists an explicit* mu-PRP$_E$ *adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\text{mu-ctxt}}_{\mathcal{S}}(\mathcal{A}) \le \mathsf{Adv}^{\text{mu-prp}}_{E}(\mathcal{B}) + \frac{v(L_A+1)}{2^\tau}\cdot\delta_n(w(\vec{q})\cdot(L+1)+v+1) .$$

Adversary $\mathcal{B}$ runs in time close to that of $\mathcal{A}$, makes at most κ key generation queries and makes at most $(L+2)(q+v)$ for $q = \sum_i i\cdot\vec{q}_i$ enciphering queries.

Nonce-head-respecting adversaries. The following theorem establishes the CTXT security of DNDK in the face of any nonce-respecting adversary, and for an arbitrary underlying AEAD. See App. J for a proof.

Theorem 12. *Let $\mathcal{S}$ be an AEAD scheme, F be a PRF, and $\mathcal{T} = DNDK[DNDKv2[F], \mathcal{S}]$ for a fixed parameter set ρ. Let $\mathcal{A}$ be a $\vec{q}$-NH respecting* CTXT$_{\mathcal{T}}$ *adversary making at most q encryption and v decryption queries. Then we give an* PRF$_F$ *adversary $\mathcal{D}$ and* mu-CTXT$_{\mathcal{S}}$ *adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\mathrm{ctxt}}_{\mathcal{T}}(\mathcal{A}) \le \mathsf{Adv}^{\mathrm{prf}}_{F}(\mathcal{D}) + \mathsf{Adv}^{\text{mu-ctxt}}_{\mathcal{S}}(\mathcal{B}) .$$

Adversary $\mathcal{D}$ makes at most $q+v$ queries. Adversary $\mathcal{B}$ is $\vec{q}$-bound, makes at most v decryption queries, and makes at most $\kappa = v + \sum_i q_i$ key generation queries, and otherwise has the same resource constraints as $\mathcal{A}$.

We get the following corollary that bounds the CTXT security of any $\vec{q}$-*NH* respecting adversary. Its proof follows from application of Theorems 11, 12, and 1.

Corollary 13 (DNDK integrity—nonce respecting)**.** *Let F be XORP and $\mathcal{S}$ being AES-GCM with τ-bit tags with both built*

[2] Notice that $\mathcal{D}$ and $\mathcal{B}_2$ as specified in the proof of Thm. 7 do not use $\vec{q}_1$, $\vec{q}_2$ explicitly, meaning the reductions remain fully specified.

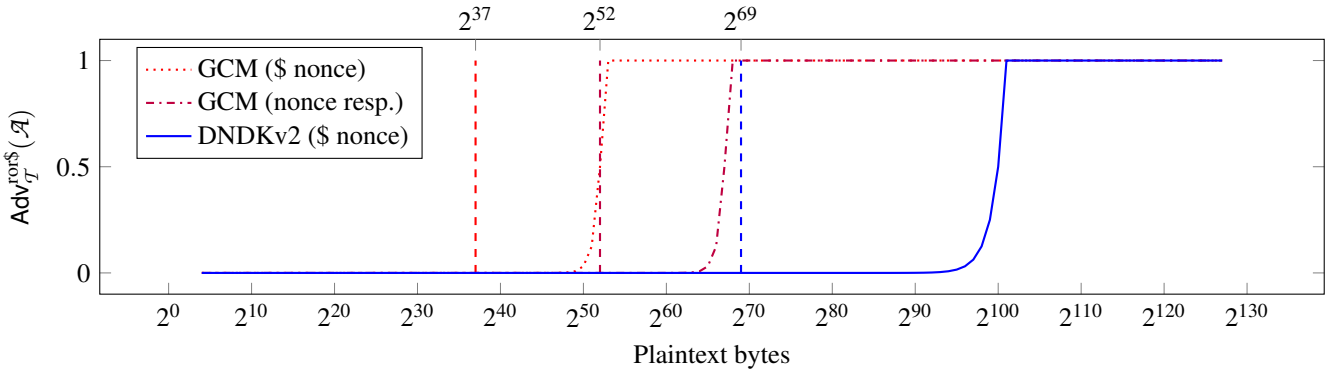

Figure 8: DNDKv2's confidentiality holds for much higher workloads compared to AES-GCM. The curves indicate an upper bound on the advantage $\mathsf{Adv}^{\mathrm{ror\$}}_{\mathcal{T}}(\mathcal{A})$. The first is $\mathcal{T}$ = GCM with random nonces (using [38, Cor. 3] with an added term for nonce collisions), the second $\mathcal{T}$ = GCM for guaranteed unique nonces (using [38, Cor. 3]), and the third $\mathcal{T}$ = DNDKv2 (using Cor. 10), each for $\mathcal{A}$ that maximizes the advantage function for increasing bytes of plaintext data encrypted, with $p = 2^{128}, \nu = 256, \ell_h = 120, n = 128, k_d = 256$. Vertical dashed lines indicate where the curves cross 2^{-32}.

from an ideal cipher $E\colon \{0,1\}^{k_d} \times \{0,1\}^n \to \{0,1\}^n$. *Let* $\mathcal{T} = DNDK[DNDKv2[F], \mathcal{S}]$ *for a fixed parameter set* ρ. *Let adversary* $\mathcal{A}$ *be a* $\vec{q}$*-NH respecting* $\mathrm{CTXT}_{\mathcal{T}}$ *adversary making at most* p *primitive queries, making queries with maximum plaintext length* L *and maximum input length* L_A*, making at most* v *decryption queries, and for which* $(w(\vec{q}) \cdot (L+1) + v + 1) \le 2^{64}$. *Let* $\kappa = v + \sum_i q_i$ *and* $q = \sum_i i \cdot q_i$. *Then*

$$\mathsf{Adv}^{\mathrm{ctxt}}_{\mathcal{T}}(\mathcal{A}) \le \frac{(\kappa+1)p + \kappa(\kappa-1)}{2^{k_d}} + \frac{16q}{2^n} + \frac{2v(L_A+1)}{2^\tau}\ .$$

To understand the statement above, recall that Bernstein [12] showed that for $a \le 2^{64}$ and $n = 128$ it is the case that $\delta_n(a) \le 1.7$. We use this fact in the proof, hence the restriction $(w(\vec{q}) \cdot (L+1) + v + 1) \le 2^{64}$.

Concretely this bound is strong. For example, when we have counters and always have unique nonce heads, then we can apply the theorem with $w(\vec{q}) = 1$. If L is maximum at $2^{32} - 2$, then the advantage bound is 2^{-32} for up to $v = 2^{63}$, $p = 2^{128}$, $L_A = 2^{32}$ (up to two blocks of authenticated data), and $\kappa = q = 2^{64}$. If we restrict L to be smaller messages, like 2^{16} blocks, and so $L_A = 2^{16}$ we can still allow up to $v = 2^{64} - 2^{16} - 1$ and the advantage bound becomes 2^{-47}.

Random nonces. The following theorem mirrors Thm. 9, showing that we can reduce to the nonce-respecting case and, moreover, exclude three-way nonce head collisions entirely.

Theorem 14. *Let* $\mathcal{S}$ *be an AEAD scheme,* F *be a PRF, and* $\mathcal{T} = DNDK[DNDKv2[F], \mathcal{S}]$ *for a fixed parameter set* ρ. *Let* $\mathcal{A}$ *be a* $\mathrm{CTXT\$}_{\mathcal{T}}$ *adversary making at most* q *encryption queries, at most* v *decryption queries with maximum input length* L_A *blocks, and all queries having maximum plaintext length* L *blocks. Then we give an* PRF_F *adversary* $\mathcal{D}$ *and* $\text{mu-CTXT}_{\mathcal{S}}$ *adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathrm{ctxt\$}}_{\mathcal{T}}(\mathcal{A}) \le \mathsf{Adv}^{\mathrm{ctxt}}_{\mathcal{T}}(\mathcal{B}) + \frac{q^2}{2^\nu} + \frac{q^3}{3 \cdot 2^{2\ell_h}}\ .$$

Adversary $\mathcal{B}$ *is* $\vec{q}$*-NH respecting for* $w(\vec{q}) \le 2$*, and otherwise uses the same resources as* $\mathcal{A}$.

We can likewise derive an upper bound in the ideal cipher model for random nonces. A proof follows by combining Thm. 14 with Thms. 11, 12, and 1.

Corollary 15 (DNDK authenticity—random nonces)**.** *Let* F *be XORP and* $\mathcal{S}$ *being AES-GCM with* τ*-bit tags with both built from an ideal cipher* $E\colon \{0,1\}^{k_d} \times \{0,1\}^n \to \{0,1\}^n$. *Let* $\mathcal{T} = DNDK[DNDKv2[F], \mathcal{S}]$ *for a fixed parameter set* ρ. *Let* $\mathcal{A}$ *be a nonce-respecting* $\mathrm{CTXT\$}_{\mathcal{T}}$ *adversary making at most* p *primitive queries, making at most* q *encryption queries, at most* v *decryption queries, and with queries having maximum plaintext length* L *and maximum input length* L_A. *Let* $\kappa = v + q$. *Then*

$$\mathsf{Adv}^{\mathrm{ctxt\$}}_{\mathcal{T}}(\mathcal{A}) \le \frac{(\kappa+1)p + \kappa^2}{2^{k_d}} + \frac{16q}{2^n} + \frac{2v(L_A+1)}{2^\tau} + \frac{q^2}{2^\nu} + \frac{q^3}{3 \cdot 2^{2\ell_h}}\ .$$

7 Discussion

We discuss aspects of DNDK, focusing particularly on its security.

Security scaling. We first return to the central motivating question: for how many message bytes can we use the same key with DNDK? To demonstrate the scaling provided by DNDK-GCM, we focus on confidentiality, specifically Cor. 10, which is the worst case—bounds are even better for nonce-head-respecting adversaries and for authenticity.

Fig. 8 plots the upper bound on ROR$ advantage as plaintext bytes encrypted scales. The upper bound is computed

using an unrealistically large computational budget for the adversary of $p = 2^{128}$, nonce length of $\nu = 256$, our suggested head length of $\ell_h = 120$, and AES256's key size $k_d = 256$ and block length $n = 128$. For each plaintext length, we compute the $L \in \{16, 32, 64, \ldots, 2^{31}, 2^{32} - 2\}$ that maximizes the advantage bound in Cor. 10.

DNDK's security scales much better than GCM's. Fig. 8 also plots GCM's security against nonce-respecting adversaries ([38, Cor. 3]), and for random nonces (the former bound but with an added term of $q(q-1)/2^{97}$ which is an upper bound on collisions for random nonces).

Confidentiality and key commitments. The results in Sec. 5 establish real-versus-random security of DNDK when key commitments are omitted. Recall that this confidentiality goal requires the entire ciphertext to be indistinguishable from random bits assuming nonce-respecting adversaries or for sufficiently large random nonces. But DNDKv2 uses key commitments that depend only on NH. Thus when $\ell_{kc} > 0$ DNDK does *not* achieve ROR security: simply make two queries with the same nonce head, and guess that you're in the real world should the key commitments match.

Pragmatically, what this translates to is that adversaries observing key commitments can infer whether the same root key, nonce head pair were used. This is related to key privacy [3, 22]. In the uniform random nonce case, key privacy in fact holds up to $2^{\ell_h/2} = 2^{60}$ encryptions with the same key, for our preferred parameter set with $\ell_h = 120$. From another viewpoint, for $q = 2^{64}$ only a *small* number of key commitment values will repeat (see App. B). We view this as an insignificant issue, but note that restricting q to at most 2^{50} reduces the repetitions probability below 1/2,000,000.

Even so, we emphasize that key commitment repetition *only* reveals repetition of root key, nonce head pairs. It does not allow inferring any information about plaintexts, and our proof techniques above can be readily translated to left-or-right confidentiality notions met by DNDK with $\ell_{kc} > 0$.

Finally, we point out that if key privacy is not necessary, one can compute key commitments independently of NH via $KC = E_K(0^{n-5} \,\|\, \rho \,\|\, \langle 3 \rangle) \,\|\, E_K(0^{n-5} \,\|\, \rho \,\|\, \langle 4 \rangle)$. This allows precomputing KC, saving two per-message AES calls.

Configuration binding. The DNDK construction (Fig. 6) includes in the input to F the parameters of the scheme. As discussed earlier, this means that using the wrong parameters to decrypt a ciphertext will result in an error, assuming key commitment is turned on. In adversarial settings, where somehow an attacker knows the secret keys *and* controls the configuration input used by a decryption implementation, then that attacker can omit from an adversarially crafted ciphertext a key commitment, set the key commitment configuration bit to 0, and mount key commitment attacks [1, 23, 45]. Thus in settings where key commitment is important, decryption implementations should use trusted configuration inputs (e.g., have hard coding the key-commitment flag).

In such settings, we can show that DNDK using CTR or XORP key commitments bind to both root keys and configurations; details appear in App. F. Briefly, we prove that XORP achieves a kind of "near-message" collision resistant in the sense that no efficient adversary can find $(K, X) \neq (K', X')$ such that $\mathrm{XORP}(K, X) = \mathrm{XORP}(K', X')$ for $(K, X) \neq (K', X')$ and X, X' differ in at most δ bits. Concretely, the bound loses a factor of 2^{δ} compared to strict key commitment (Thm. 3), but $\delta = 5$ in our context. This establishes that no efficient adversary can find $(K, N, \rho) \neq (K', N, \rho')$ where $\rho \neq \rho'$, yet the key commitment values will match. We also give an analogous result for CTR as a KDF; its proof is much simpler.

Multi-user ROR and CTXT security. We focused on the single-user setting for DNDK, which was the explicit design goal. A variety of prior works have highlighted the importance of multi-user security (e.g., [14, 34]). Our results can be readily translated to the multi-user setting via the generic hybrid, but this would be loose as it would imply an advantage bound with additional multiplicative factor κ, the number of DNDK key instances attacked. Tighter bounds should be achievable via direct analysis, extending our notions of nonce-head respect and boundedness to the multi-user setting.

Tightness of DNDK bounds. We believe our bounds in the nonce-respecting case are tight up to constant factors; we focus discussion on the confidentiality bounds for brevity.

Looking at Cor. 8, the dominant terms are $q^2/2^{k_d}$, $qp/2^{k_d}$, and $q(L+2)^2/2^n$. Each corresponds to a real attack: (1) Make $q \approx 2^{k_d/2}$ encryption queries with the same nonce tail and distinct nonce heads, and with probability close to 1 the adversary will get a repeat DK, NT pair, violating confidentiality. (2) Make q encryption queries with the $NT = 0^{\ell_t}$ and different nonce heads on a message of k_d zeros, and p ideal cipher queries $Y_i \leftarrow IC(K_i, NT \,\|\, \langle 1 \rangle)$. Check if any Y_i values match any of the first block of ciphertext, if so make an additional constant number of ideal cipher queries to confirm the key K_i. (3) Make $q \approx 2^n/L^2$ encryption queries with the same nonce head, for some arbitrary plaintext of length L. Look for a pair of ciphertext blocks that match.[3]

The above relates to why our analyses did not try to use the multi-user security bounds for AES-GCM established by Hoang, Tessaro, and Thiruvengadam [34] instead of using Theorems 6 and 11. Their analysis focuses on the d-bounded model where the adversary can reuse the same AES-GCM nonce at most d times across all key instances. In our context this would be the number of key instances which are queried with the derived nonce—this could be as high as the total number of encryption queries in the nonce-respecting setting. In this case using their bound [34, Theorem 3.1] would be looser than what we get in Cor. 8.

For the uniform random nonce case, our analysis is loose, because in fact security does not fail after $2^{2\ell_h/3}$ queries, even

[3] An analysis of a related attack appears in [34].

though the bound in Cor. 10 is vacuous for so many queries. We also basically assume the worst-case number of two-way collisions, merely restricting $w(\vec{q}) \leq 2$. But in reality the number of colliding derived keys is likely to be a small constant, not close to q as our bound allows. See App. B where we provide some estimates of the expected number of collisions. We conjecture that one can improve bounds further here, possibly using the techniques of [34].

Our bounds even for the random nonce case are very good, granting the ability to encrypt 2^{64} bytes under a single key with sufficiently good bounds on security, as discussed earlier.

Different nonce splits. In type DNDKv2-style transforms (bottom right of Fig. 4 in Sec. 3) the nonce is split into two parts and use one part of the split for the derived nonce (NT). A key question is how to choose the split. As discussed in Sec. 3 we advocate for maximizing the size of the nonce head (NH), which is used for generating the derived key. The reason is that this leads to better security bounds: the larger NH the fewer collisions one obtains on the derived key, which translates to fewer bytes encrypted under any particular derived key.

8 Alternative Instantiations

DNDK has evolved across several versions, and there has been a steady amount of follow-up work suggesting alternative ways of deriving keys and nonces. We survey these various approaches here, and compare them to DNDKv2.

DNDKv1. The first author began work on DNDK in 2023, and developed it largely in public via RFC drafts [29] (April 2024) through [31]. This has been helpful for receiving valuable community feedback, and it also has prompted some works that followed-up on the public RFC drafts. The first derivation function, DNDKv1 (see Fig. 9 in App. K), assumes 24-byte nonces that are split into two halves. Each half has an instance of XORP applied, with the two XORP outputs then XOR'd to generate the derived key and key commitment value. The nonce used with AES-GCM is set to zeros.

DNDKv1 has several limitations that are rectified by DNDKv2. First, after DNDKv1 was published, Degabriele pointed out [21] the lack of PRF security for adversarially chosen nonces for the double XORP construction. For example, note that DNDKv1 applied to the nonce $N = NH \,\|\, NT$ will produce the same output as for input $N' = NT \,\|\, NH$, assuming the same secret key and configuration. This obviates modular proofs of DNDKv1 that would rely on strong PRF security, but the issue does not appear to affect DNDKv1's security which is meant to be used with only random nonces. Nevertheless, and subsequent to these correspondences, the first author independently developed DNDKv2.

DNDKv2 improves on DNDKv1 in other ways as well. It uses fewer AES invocations. It allows nonce splits that achieve better concrete security. It also works better in the case of structured nonces that utilize counters in NT.

Pencil. In parallel with development of DNDKv2, a concurrent work by Bhaumik and Degabriele [13] explored construction of weak and strong beyond-birthday-bound PRFs suitable for use within DNDK-type transforms, specifically the type that derives both a key and nonce from the KDF (bottom left of Fig. 4). Their Pencil construction requires seven AES calls compared to DNDKv2's three; they also do not explore key commitment.

Derivation using CTR. One can do key derivation using a CTR-style mode by replacing $E_K(X \,\|\, \langle 0 \rangle)$ in our construction of F with 0^{128}. This relates to an observation about AES-GCM-SIV by Bose, Hoang, and Tessaro [14]. Doing so saves one AES call in deriving the per-message key. However, this requires using a nonstandard assumption that AES with keys sampled from a subset of $\{0,1\}^{256}$ is sound, which holds in the ideal cipher model but complicates standard model analyses. XORP avoids that complexity.

XORP derivation and CTR key commitment. One can combine XORP and CTR, using the former for key derivation and the latter for key commitment: just follow Fig. 5 but instead setting $Y_3 \leftarrow E_K(X_3)$ and $Y_4 \leftarrow E_K(X_4)$. This has the advantage of a simpler proof of key and configuration commitment (App. F).

Derivation using CMAC. Valsardo [58] proposed a construction called XAES-256-GCM which provides extended nonces. It can be viewed as DNDKv2, using a nonce of 192 bits and instantiated using two invocations of CMAC for the KDF. This is an attempt to align it with the NIST standard for AES key derivation from SP800-108r1 [18] using CMAC as defined in SP800-38B [24]. The latter, however, recommends a limit of 2^{48} invocations of CMAC with the same key, making the assertion that XAES-256-GCM is FIPS compliant questionable when used to encrypt more than 2^{47} messages (each encryption requires two CMAC invocations) because this usage departs from NIST's recommended practice.

XAES-256-GCM's KDF is also more complicated than DNDKv2's, requiring finite field operations that have no security purpose. Like CTR mode, security of the construction requires non-standard assumptions on AES, given that the keys for AES256-GCM are chosen from a subset of the key space. Recent work by Benhamouda et al. [11] provide a security analysis based on a new, non-standard assumption on AES-GCM. Another preprint shows how to add CMAC-based key commitment security as well [42].

9 Performance of DNDK

We report on the performance of our DNDK framework as applied to GCM, an instantiation that matches the DNDK-GCM RFC draft version 2. Here the underlying AEAD scheme is

KC?	Operation	Cycles
	AES256 key expansion	74
No	*DNDKv2*	47
Yes	*DNDKv2*	71
No	Key expansion + *DNDKv2*	115
Yes	Key expansion + *DNDKv2*	139
No	DNDK-GCM w/o AES256-GCM	235

Table 1: DNDK-GCM overheads in cycles (on an Intel Skylake CPU) with and without key commitment (KC).

AES256-GCM (with 256-bit keys) and the derivation function *DNDKv2* is built over the XORP PRF (see Fig. 6). The parameters of this instantiation are:

$$k = k_d = 256\ ,\ \ell_h = 120\ ,\ \ell_t = 72\ ,\ \nu = 192\ ,\ \nu_d = 96\ .$$

Recall that the configuration ρ is encoded in five bits as

$$128 \cdot \mathsf{KC_Choice} + 8 \cdot \left(\frac{\nu}{8} - 12\right)$$

where $\nu = 192$ and KC_Choice is a selector bit that indicates whether or not key commitment is required. With key commitment (KC_Choice $= 1$), we have $\rho = 224$, which is encoded (and parsed) as the 5-bit string 11100. Without key commitment we have $\rho = 96$ which is encoded (and parsed) as the 5-bit string 01100. Thus, the input X to the PRF $F(K,X)$ is $NH \,\|\, \rho$ (of length $120+5 = 125$ bits) and the five (or three) formatted blocks that are used with XORP are $NH \,\|\, \rho \,\|\, \langle i \rangle$ for $1 \le i \le 4$. The derived nonce is $DN = NT \,\|\, 0^{24}$.

Estimating the overheads. To estimate the extra computational overheads of the transform compared to the underlying AEAD ($\mathcal{S}$), we focus on the encryption function, i.e., *DNDK.Enc* (decryption follows in a similar way). Every invocation of *DNDK.Enc* invokes *DNDKv2* and follows (serially) with an invocation of $\mathcal{S}$. Since $\mathcal{S}$ is invoked with a fresh key (for every encryption except with tiny probability), the overhead estimation needs to account for some key setup cost.

Suppose that $\mathcal{S}$ is AES256-GCM. Then, the key-setup involves an expansion of one AES256 key, the computation of the GHASH key ($H \leftarrow E(DK, 0^{128})$) and, if desired, the computation of a few powers of H that can be subsequently used for deferred reduction in subsequent GHASH computations (we use six powers of H). The XORP-based *DNDKv2* function requires five calls to AES256 with the root key if a key commitment is required and only three calls if it is not. These AES256 computations are executed over independent blocks so they can be parallelized for improved performance.

With 14 AES rounds for AES256 and assuming a platform with AES-NI, we pay $\sim$42 cycles for three AES256 calls and $\sim$70 cycles for five AES256 calls. Key expansion for AES256 consumes $\sim$75 cycles. Recall that one needs to account for the encryption of a zero block and computing six powers of H. We estimate that the overhead should not exceed $\sim$150 cycles without key commitment and $\sim$240 cycles with it.

Bytes	AES256-GCM	DNDK-GCM	
		No KC	With KC
16	102	328	338
48	177	403	410
64	202	434	438
128	274	493	509
256	409	642	648
1,024	1,104	1,314	1,320
2,048	1,951	2,174	2,183
4,096	3,840	4,010	4,026
16,384	14,578	14,805	14,868
32,768	29,124	29,177	29,147
1,048,576	921,593	921,698	922,118

Table 2: Performance (in cycles) of DNDK-GCM (with and without key commitment) compared to AES256-GCM for different message lengths and no associated data.

Other PRF options. To compare the performance of XORP ($\sim$42 and $\sim$70 cycles) to some alternative PRFs, consider the following performance. These are measured on a Skylake processor over 16 KB buffers. HMAC-SHA256 performs at $\sim$8 cycles per byte (cpb), BLAKE2b-512 at $\sim$4.44 cpb, CMAC at $\sim$3.34 cpb, BLAKE3 at $\sim$1.89 cpb. For the comparison, multiply these by 32 (no key commitment) or by 64 (with key commitment). These estimates suggest that XORP is very competitive, while requiring just one underlying primitive when used in conjunction with AES256-GCM.

Experimental performance results. We have already implemented and released an implementation of DNDK-GCM within the public, open source Haberdashery library [35]. To evaluate its performance, we report on microbenchmarks measured on a 6th generation Intel processor (microarchitecture codename Skylake). The specifics are Intel Xeon CPU (cpu family: 6; model: 85) running at 1.60 GHz. This platform has 48 GB RAM, 1MiB L1d cache, 1MiB L1i cache, 4MiB L2 cache, and 16MiB L3 cache.

The performance reported here was measured in processor cycles (per single core), where lower count is better. Every measured function was isolated and first run for 200 ms (as a warm-up) to evaluate the number of iterations (*iters*) per 100 ms run. Then, the function was run *iters* times that were clocked (using the perf_event_open with attribute PERF_COUNT_HW_CPU_CYCLES). The results are then averaged. To minimize the effect of background tasks that run on the system, every experiment was repeated 10 times and the minimum result was recorded.

The results are summarized in Tables 1 and 2. Tab. 1 shows the cycle counts for the overhead steps of DNDK-GCM (i.e., the steps on top of the raw underlying AES256-GCM). Tab. 2 reports the performance of AES256-GCM and DNDK-GCM (with and without key commitment).

Acknowledgments

This research was supported by BSF grant number 2023770, NSF grant CNS-2427390, and a gift by the Google Cyber NYC program. We thank Ed Knap and Qiye Tan for posting code to [35] and performance profiling.

Open Science

We have posted the script used to generate data for Fig. 8 and our implementation of DNDK-GCM at `https://zenodo.org/records/16951698`. A third-party implementation is also available in the public, open source Haberdashery library [35].

References

[1] Ange Albertini, Thai Duong, Shay Gueron, Stefan Kölbl, Atul Luykx, and Sophie Schmieg. How to abuse and fix authenticated encryption without key commitment. In *USENIX Security Symposium*, 2022.

[2] Ange Albertini, Thai Duong, Shay Gueron, Stefan Kölbl, Atul Luykx, and Sophie Schmieg. How to abuse and fix authenticated encryption without key commitment. Cryptology ePrint Archive, Paper 2020/1456, 2020.

[3] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *Advances in Cryptology – ASIACRYPT*, 2001.

[4] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In *Advances in Cryptology — CRYPTO*, 1996.

[5] Mihir Bellare, John Chan, Paul Grubbs, Viet Tung Hoang, Sanketh Menda, Julia Len, Thomas Ristenpart, and Phillip Rogaway. Ask your cryptographer if context-committing AEAD is right for you. Presented at Real World Crypto, 2023.

[6] Mihir Bellare, Oded Goldreich, and Hugo Krawczyk. Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier. In *Advances in Cryptology – CRYPTO*, 1999.

[7] Mihir Bellare, Shay Gueron, Viet Tung Hoang, Julia Len, Sanketh Menda, and Thomas Ristenpart. Building the next generation of AEAD. Presented at Real World Crypto, 2024.

[8] Mihir Bellare and Viet Tung Hoang. Efficient schemes for committing authenticated encryption. In *Advances in Cryptology – EUROCRYPT*, 2022.

[9] Mihir Bellare, Viet Tung Hoang, and Cong Wu. The landscape of committing authenticated encryption. The Third NIST Workshop on Block Cipher Modes of Operation, 2023.

[10] Mihir Bellare, Ted Krovetz, and Phillip Rogaway. Luby-rackoff backwards: Increasing security by making block ciphers non-invertible. In *Advances in Cryptology – EUROCRYPT*, 1998.

[11] Fabrice Benhamouda, Shai Halevi, Panos Kampanakis, and Hugo Krawczyk. Blockcipher-based key derivation without PRP/PRF switching. Cryptology ePrint Archive, Paper 2025/878, 2025.

[12] Daniel J. Bernstein. Stronger security bounds for Wegman-Carter-Shoup authenticators. In *Advances in Cryptology – EUROCRYPT*, 2005.

[13] Ritam Bhaumik and Jean Paul Degabriele. Pencil: A domain-extended PRF with full n-bit security for strengthening GCM and more. Cryptology ePrint Archive, Paper 2025/383, 2025.

[14] Priyanka Bose, Viet Tung Hoang, and Stefano Tessaro. Revisiting AES-GCM-SIV: multi-user security, faster key derivation, and better bounds. In *Advances in Cryptology – EUROCRYPT*, 2018.

[15] Matthew Campagna and Shay Gueron. AWS key management service (KMS); handling cryptographic bounds for use of AES-GCM. Presented at Real World Crypto, 2018. `https://rwc.iacr.org/2018/Slides/Gueron.pdf`.

[16] Matthew Campagna and Shay Gueron. Key management systems at the cloud scale. *Cryptography*, 3(3), 2019.

[17] John Chan and Phillip Rogaway. On committing authenticated-encryption. In *European Symposium on Research in Computer Security*, 2022.

[18] Lily Chen. Recommendation for key derivation using pseudorandom functions. NIST Special Publication 800-108 Rev. 1, National Institute of Standards and Technology (NIST), 2008.

[19] Yu Long Chen, Antonio Flórez-Gutiérrez, Akiko Inoue, Ryoma Ito, Tetsu Iwata, Kazuhiko Minematsu, Nicky Mouha, Yusuke Naito, Ferdinand Sibleyras, and Yosuke Todo. Key committing security of AEZ. The Third NIST Workshop on Block Cipher Modes of Operation, 2023.

[20] Joan Daemen, Silvia Mella, and Gilles Van Assche. Committing authenticated encryption based on shake. Cryptology ePrint Archive, Paper 2023/1494, 2023. `https://eprint.iacr.org/2023/1494`.

[21] Jean Paul Degabriele. Personal communication, 2024.

[22] Anand Desai. The security of all-or-nothing encryption: Protecting against exhaustive key search. In *Advances in Cryptology – CRYPTO*, 2000.

[23] Yevgeniy Dodis, Paul Grubbs, Thomas Ristenpart, and Joanne Woodage. Fast message franking: From invisible salamanders to encryptment. In *Advances in Cryptology–CRYPTO*, 2018.

[24] Morris Dworkin. Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. NIST Special Publication 800-38B, 2005.

[25] Morris J Dworkin. SP 800-38d: Recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC, 2007.

[26] Shafi Goldwasser and Mihir Bellare. Lecture notes on cryptography, 2008. `https://cseweb.ucsd.edu/~mihir/papers/gb.pdf`.

[27] Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. Message franking via committing authenticated encryption. In *Advances in Cryptology – CRYPTO*, 2017.

[28] Shay Gueron. Constructions based on the AES round and polynomial multiplication that are efficient on modern processor architectures. Third NIST Workshop on Block Cipher Modes of Operation, 2023.

[29] Shay Gueron. Double Nonce Derive Key AES-GCM (DNDK-GCM). Internet-Draft draft-gueron-cfrg-dndkgcm-00, Internet Engineering Task Force, 2023. Work in Progress.

[30] Shay Gueron. I want to encrypt 2^{64} bytes with AES-GCM using a single key. Presented at Real World Crypto 2024, 2024.

[31] Shay Gueron. Double Nonce Derive Key AES-GCM (DNDK-GCM). Internet-Draft draft-gueron-cfrg-dndkgcm-02, Internet Engineering Task Force, March 2025. Work in Progress.

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

[33] Shay Gueron and Yehuda Lindell. Better bounds for block cipher modes of operation via nonce-based key derivation. In *Conference on Computer and Communications Security*. ACM, 2017.

[34] Viet Tung Hoang, Stefano Tessaro, and Aishwarya Thiruvengadam. The multi-user security of GCM, revisited: Tight bounds for nonce randomization. In *ACM Conference on Computer and Communications Security*, 2018.

[35] Facebook Incubator. Haberdashery: A collection of useful React components. `https://github.com/facebookincubator/haberdashery`, 2025. Accessed: 2025-03-26.

[36] Tetsu Iwata. New blockcipher modes of operation with beyond the birthday bound security. In *Fast Software Encryption*, 2006.

[37] Tetsu Iwata, Bart Mennink, and Damian Vizár. CENC is optimally secure. Cryptology ePrint Archive, Paper 2016/1087, 2016.

[38] Tetsu Iwata, Keisuke Ohashi, and Kazuhiko Minematsu. Breaking and repairing GCM security proofs. In *Advances in Cryptology – CRYPTO*, 2012.

[39] Tetsu Iwata and Yannick Seurin. Reconsidering the security bound of AES-GCM-SIV. *IACR Transactions on Symmetric Cryptology*, pages 240–267, 2017.

[40] Panos Kampanakis, Matt Campagna, Eric Crocket, and Adam Petcher. Practical challenges with AES-GCM and the need for a new cipher. Third NIST Workshop on Block Cipher Modes of Operation, 2023.

[41] Panos Kampanakis, Matt Campagna, Eric Crocket, Adam Petcher, and Shay Gueron. Practical challenges with AES-GCM and the need for a new cipher. 2023.

[42] Panos Kampanakis, Shai Halevi, Nevine Ebeid, and Matt Campagna. Blockcipher-based key commitment for nonce-derived schemes. Cryptology ePrint Archive, Paper 2025/758, 2025.

[43] Hugo Krawczyk. Cryptographic extraction and key derivation: The HKDF scheme. In *Advances in Cryptology –CRYPTO*, 2010.

[44] Ted Krovetz and Phillip Rogaway. The OCB Authenticated-Encryption Algorithm. RFC 7253, 2014.

[45] Julia Len, Paul Grubbs, and Thomas Ristenpart. Partitioning oracle attacks. In *USENIX Security Symposium*, 2021.

[46] John Preuß Mattsson, Ben Smeets, and Erik Thormarker. Proposals for standardization of encryption schemes. The Third NIST Workshop on Block Cipher Modes of Operation, 2023.

[47] David McGrew and John Viega. The Galois/counter mode of operation (GCM). *Submission to NIST Modes of Operation Process*, 2004.

[48] Sanketh Menda, Mihir Bellare, Viet Tung Hoang, Julia Len, and Thomas Ristenpart. The OCH authenticated encryption scheme. In *ACM Conference on Computer and Communications Security*, 2025.

[49] Meta. The labyrinth encrypted message storage protocol, 2023.

[50] Nicky Mouha and Atul Luykx. Multi-key security: The Even-Mansour construction revisited. In *Advances in Cryptology – CRYPTO*, 2015.

[51] Chanathip Namprempre, Phillip Rogaway, and Thomas Shrimpton. Reconsidering generic composition. In *Advances in Cryptology – EUROCRYPT*, 2014.

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

[53] Yuichi Niwa, Keisuke Ohashi, Kazuhiko Minematsu, and Tetsu Iwata. GCM security bounds reconsidered. In *Fast Software Encryption*, 2015.

[54] Jacques Patarin. Introduction to mirror theory: Analysis of systems of linear equalities and linear non equalities for cryptography. Cryptology ePrint Archive, Paper 2010/287, 2010.

[55] Phillip Rogaway, Mihir Bellare, and John Black. OCB: A block-cipher mode of operation for efficient authenticated encryption. *ACM Transactions on Information and System Security (TISSEC)*, 6(3):365–403, 2003.

[56] Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In *Advances in Cryptology – EUROCRYPT*, 2006.

[57] Kazuhiro Suzuki, Dongvu Tonien, Kaoru Kurosawa, and Koji Toyota. Birthday paradox for multi-collisions. In *Information Security and Cryptology – ICISC*, 2006.

[58] Filippo Valsorda. The XAES-256-GCM extended-nonce AEAD. `https://github.com/C2SP/C2SP/blob/main/XAES-256-GCM.md`, 2024.

[59] David A. Wagner. A generalized birthday problem. In *Advances in Cryptology – CRYPTO*, 2002.

A Three-way Collisions

We use in various proofs a bound on three-way collisions. Let $2 \le r \le q \le s$. Suppose that q balls are thrown, one by one (independently) at random, into s bins. An r-way collision is the event where there exists at least one bin that contains at least r balls. Then, the probability that an r-way collision occurs is at most $q^r \cdot r!/s^{r-1}$. In particular, for $r = 3$, the probability for a 3-way collision is at most $q^3/(6s^2)$ (see [57, Theorem 2]).

B Random NH Collision Estimates

Suppose that q balls are thrown uniformly at random (one at a time, independently) into $N = 2^{\ell_h}$ bins. A *2-way collision* is the event in which a pair of balls land in the same bin. We wish to find an estimated upper bound for the probability that the total number of such pairwise collisions exceeds a threshold w, under the following assumptions.

1. Neglect 3-way (or higher) collisions: Assume that no bin receives more than two balls. This is a valid approximation when the number of balls is relatively small compared to the number of bins, so such higher-order collisions are rare (e.g., $q \ll N^{2/3}$).
2. Low-density regime: Assume that $q \le \sqrt{N \cdot V}$ for some relatively small parameter V. Under this assumption, the number of occupied bins grows nearly linearly with q, and the collision rate stays low.

Let T be the number of distinct unordered pairs of balls that collide (i.e., land in the same bin). There are $\binom{q}{2} \approx \frac{q^2}{2}$ such pairs, and each pair collides with probability $\frac{1}{N}$. Assuming that the collisions between pairs are approximately independent (a reasonable approximation for our regime), we can model T as a Poisson random variable with mean $\lambda = \frac{q^2}{2N}$. This is a standard application of "the law of rare events" where we have many independent trials, each having a small probability of success. For a Poisson random variable $T \sim \text{Poisson}(\lambda)$, the probability that T exceeds a value $w = (1+\delta)\lambda$ can be efficiently upper bounded by using the Chernoff bound:

$$P(T \ge w) \le \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right)^{\lambda}$$

This bound decays exponentially in λ and indicates that large deviations from the expected number of collisions are unlikely. We show two cases as follows.

Case 1. $w = 2\lambda$**.** Set $\delta = 1$. Then,

$$P(T \ge 2\lambda) \le \left(\frac{e}{4}\right)^{\lambda} \approx e^{-0.6137\lambda} \approx 2^{-0.886\lambda}$$

Case 1. $w = 1.8\lambda$**.** Set $\delta = 0.8$. Then,

$$\frac{e^{0.8}}{1.8^{1.8}} \approx \frac{2.2255}{2.881} \approx 0.772$$

so

$$P(T \ge 1.8\lambda) \le (0.772)^{\lambda} \approx e^{-0.258\lambda} \approx 2^{-0.372\lambda}$$

For example, if $N = 2^{120}$ and $q = 2^{64}$, we have $\lambda = \frac{q^2}{2N} = 128$, and hence $2^{-0.372\lambda} \approx 2^{-47}$. This implies that the (approximated) probability to encounter more than $128 \times 1.8 = 230$ collisions is at most 2^{-47}, i.e., the number of collisions is negligible relative to q.

C A Useful Lemma

We use the following lemma in the several proofs.

Lemma 16. *Fix $n > 0$ and $w \le 0.5 \cdot \sqrt{2^n}$. Then*

$$\prod_{i=1}^{w} \frac{1}{2^n - i} < \frac{2}{2^{wn}}$$

Proof: We use the fact that $(1 - w/2^n) \ge e^{-2w/2^n}$ and the fact that our restriction that $w \le 0.5\sqrt{2^n}$ implies that $e^{2w^2/2^n} < 2$, as follows:

$$\begin{aligned}
\prod_{i=1}^{w} \frac{1}{2^n - i} &< \prod_{i=1}^{w} \frac{1}{2^n - w} \\
&= \left(\frac{1}{2^n(1 - w/2^n)} \right)^w \\
&\le \left(\frac{e^{2w/2^n}}{2^n} \right)^w \\
&< \frac{2}{2^{wn}} .
\end{aligned}$$

□

D Key Commitment of the CTR KDF

We provide an analysis of CTR as a KDF using our formulation of k-CMT security, which differs from the q-KC notion from [1]. Our notion dispenses with AEAD encryption and decryption oracles, aligning it more closely with commitment and collision resistance notions—our security goal and theirs are formally equivalent (up to small changes in resource use). Our results here can therefore be seen as a restatement of [1, Thm. 1] and an alternative[4] proof. For simplicity we focus on the case where the requested output is a multiple of n.

Let E be a block cipher with n-bit block length. To define the CTR derivation function fix an integer $w > 0$ and let

$$\mathrm{CTR}(K, X) = E_K(X \,\|\, \langle 1 \rangle) \;\|\; \cdots \;\|\; E_K(X \,\|\, \langle w \rangle)$$

for $X \in \{0,1\}^{n - \lceil \log_2 w \rceil}$ and where we assume $\langle i \rangle$ is some canonical encoding in $\lceil \log_2 w \rceil$ bits. We let $X_i = X \,\|\, \langle i \rangle$. We have the following theorem.

Theorem 17 (CTR's key commitment)**.** *Let E be an ideal cipher with keylength k and block length n, fix $w > 0$ such that $n = m + r$ for $r = \lceil \log_2 w \rceil < n$. Let* CTR *be the CTR-mode KDF defined above. Then for any* k-CMT$_{\mathrm{CTR}}$ *adversary $\mathcal{C}$ making at most p ideal cipher queries it holds that*

$$\mathsf{Adv}^{\text{k-cmt}}_{\mathrm{CTR}}(\mathcal{C}) \le \frac{p(p-1)}{2^{wn}} .$$

[4] There's a minor error in their proof: they state that $|S| \le 2^R \cdot 2^n!/(2^n - L + 1)!$ [2, App. F] but in fact $|S| \le 2^R \cdot 2^n!/(2^n - L)!$.

Proof: Let E_K denote the ideal cipher randomly chosen permutation for key K. To win the k-CMT$_{\mathrm{CTR}}$ game in the ideal cipher model, the adversary must query on keys $K \ne K'$ and a value X such that

$$E_K(X_i) = E_{K'}(X_i) \quad \text{for} \quad 1 \le i \le w . \tag{1}$$

We have then that

$$\begin{aligned}
\mathsf{Adv}^{\text{k-cmt}}_{\mathrm{XORP}}(\mathcal{C}) &\le \sum_{K \ne K'} \Pr\left[\bigwedge_{i=1}^{w} E_K(X_i) = E_{K'}(X_i) \right] && (2) \\
&= \binom{p}{2} \cdot \prod_{i=0}^{w-1} \frac{1}{2^n - i} && (3) \\
&< \binom{p}{2} \cdot \prod_{i=1}^{w} \frac{1}{2^n - i} \\
&< \frac{p(p-1)}{2^{wn}} . && (4)
\end{aligned}$$

We justified (2) above. For (3), we have that $\mathcal{C}$ with p queries can query at most $\binom{p}{2}$ unique pairs of points, together with the fact that E'_K is independently chosen compared to E_K. Finally to justify (4), we use Lem. 16. □

E Proof of Thm. 3

Let E_K denote the ideal cipher randomly chosen permutation for key K. To win the k-CMT$_{\mathrm{CTR}}$ game in the ideal cipher model, the adversary must query on keys $K \ne K'$ and a value X such that

$$E_K(X_0) \oplus E_K(X_i) = E_{K'}(X_0) \oplus E_{K'}(X_i) \quad \text{for} \quad 1 \le i \le w . \tag{5}$$

Notice that neither the left or right hand sides of the equality can be zero, since E_K and $E_{K'}$ are permutations and X_i are all distinct points.

For $i \in [1..w]$, let CP_i be the event that at some point in the game a query with key K' is made such that a previous K, X was queried such that $E_K(X_0) \oplus E_K(X_i) = E_{K'}(X_0) \oplus E_{K'}(X_i)$. We have then that

$$\begin{aligned}
\mathsf{Adv}^{\text{k-cmt}}_{\mathrm{XORP}}(\mathcal{C}) &\le \sum_{K \ne K'} \Pr[\wedge_{i=1}^{w} \mathsf{CP}_i] \\
&= \binom{p}{2} \cdot \prod_{i=1}^{w} \frac{1}{2^n - i} && (6) \\
&< \frac{p(p-1)}{2^{wm}} && (7)
\end{aligned}$$

To justify (6), we apply the chain rule of probability and get that the right-hand-side probability within the sum is equal to

$$\Pr\left[\mathsf{CP}_w \mid \wedge_{i=1}^{w-1} \mathsf{CP}_i\right] \cdot \ldots \cdot \Pr[\mathsf{CP}_2 \mid \mathsf{CP}_1] \cdot \Pr[\mathsf{CP}_1] \tag{8}$$

where the probabilities are just over the choice of $E_{K'}$. Then we have that for all $1 \le i \le j$

$$\Pr\left[\mathsf{CP}_i \mid \wedge_{j=1}^{i-1} \mathsf{CP}_j\right] = \frac{1}{2^n - i}$$

where this notation assumes the condition is dropped for the case that $i = 1$. To see why, observe that $\Pr[\mathsf{CP}_1] = 1/(2^n - 1)$ since $E_{K'}(X_0)$ can be chosen arbitrarily and, once set, $E_{K'}(X_1)$ must equal $E_K(X_1) \oplus E_{K'}(X_0) \oplus E_K(X_0)$. For $i > 0$ we have that $E_{K'}(X_i)$ must equal $E_K(X_i) \oplus E_{K'}(X_0) \oplus E_K(X_0)$ and the conditioning has already fixed i points in the range of $E_{K'}$. Substituting back into (8) we get that

$$\Pr\left[\wedge_{i=1}^{w} \mathsf{CP}_i\right] = \prod_{i=1}^{w} \frac{1}{2^n - i}$$

To justify (7), apply Lem. 16.

F Key and Config Commitment

The key commitment security of XORP established by Thm. 3 assumes a single value for X. In our use within DNDKv2 this means that the theorem does not prove binding to a configuration ρ under adversarially chosen keys. Here we turn to showing that XORP in fact also ensures that DNDKv2 commits to the configuration, with a slightly weaker bound than (just) key commitment.

δ-relaxed key commitment security. To formalize this, we define δ-relaxed key commitment security of a function $F\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ as follows, where $\delta \in \mathbb{N}$ is a parameter. Game gk-CMT$_{F,\delta}$ runs an adversary $\mathcal{C}$ that outputs two key-message pairs $(K,X) \neq (K',X')$ where both pairs are members of the set $\mathcal{K} \times \mathcal{X}$. The adversary wins if:

(1) $F(K,X) = F(K',X')$, and

(2) $\Delta(X,X') \leq \delta$ where Δ is Hamming distance.

In words, the adversary must find key, input pairs that collide, subject to the constraint that the two inputs are close in Hamming distance.[5] We define advantage as

$$\mathsf{Adv}^{\text{gk-cmt}}_{F,\delta}(\mathcal{C}) = \Pr\left[\text{gk-CMT}_{F,\delta}(\mathcal{C}) \Rightarrow 1\right] .$$

For our use in DNDK-style modes, achieving gk-CMT security implies that key commitment values simultaneously bind to the root key and configuration as long as the adversary only controls δ or fewer bits of the configuration (encoded in X). Of course, an adversarially controlled configuration may confuse decryption, depending on how it is implemented, into treating different parts of a ciphertext as key commitment values. We leave modeling and analyzing this more general case to future work.

Insecurity of CTR, XORP for large δ. If F is collision-resistant in all inputs, then one can show in a straightforward way δ-relaxed key commitment for arbitrarily large δ. But the block-cipher-based KDFs we consider—CTR and XORP—are not collision-resistant in all inputs.

As an example, consider CTR as defined in App. D for $w = 1$. Then we can see that for $\delta = n$ there exists a trivial attack: have $\mathcal{C}$ pick arbitrary K, K', X with $K \neq K'$, query $Y \leftarrow E_K(X_1)$ and then query $X' \leftarrow E^{-1}_{K'}(Y)$. Output K, K', X together with the first $n - r$ bits of X' for $r = \lceil \log_2 w \rceil$. This achieves $\mathsf{Adv}^{\text{gk-cmt}}_{F,n}(\mathcal{C}) = 1/2^r$ and since r is small, this succeeds with good probability. (One could achieve advantage close to one by just rejection sampling on K' to ensure the right format for X'.)

[5] The definition could readily be generalized to other distance measures.

CTR achieves generalized key commitment security. We start by analyzing the simpler CTR-mode PRF. Fix an integer $w > 0$ and define CTR as per App. D. We have the following theorem statement.

Theorem 18 (CTR is key and config committing)**.** *Let E be an ideal cipher with keylength k and blocklength n, fix w, m, δ such that $n = m + r$ for $r = \lceil \log_2 w \rceil < n$ and $\delta \in [0..m]$, and let F be the CTR-mode PRF defined in App. D. Then for any* gk-CMT$_{\text{CTR},\delta}$ *adversary $\mathcal{C}$ making at most p ideal cipher queries it holds that*

$$\mathsf{Adv}^{\text{gk-cmt}}_{CTR,\delta}(\mathcal{C}) \leq \frac{p(p-1)}{2^{wn-\delta}} .$$

For CTR used as KDF we have that $\delta = 5$ since configuration uses only 5 bits. Thus key and config commitment when $w = 2$ holds up to a bit less than $p = 2^{112}$ computation.

Proof: Let E_K denote the ideal cipher randomly chosen permutation for key K. First we observe that to win the gk-CMT$_{\text{CTR},\delta}$ game, the adversary must output $K \neq K'$. Suppose otherwise. Then $E_K(X_i) = E_K(X'_i)$ implies that $X = X'$ because E_K is a permutation, and so $(K,X) = (K,X')$, which is not a win and so a contradiction. Thus to win the gk-CMT$_{\text{CTR},\delta}$ game in the ideal cipher model, the adversary must query on keys $K \neq K'$ and values X, X' such that

$$E_K(X_i) = E_{K'}(X'_i) \quad \text{for} \quad 1 \leq i \leq w \tag{9}$$

and $\Delta(X,X') \leq \delta$.

For any $K \neq K'$, let CP be the event that at some point in the game a first query with key K' is made such that: (1) a previous query occurred for key K and that defined $E_K(X_i)$ for $i \in [1..w]$; and (2) the permutation $E_{K'}$ is chosen such that $E_{K'}(X'_i) = E_K(X_i)$ for $i \in [1..w]$ and with $\Delta(X,X') \leq \delta$. Note that here we are generously giving the adversary credit if they can query such a $E_{K'}$ on any point (either forward or inverse query).

We then have that

$$\mathsf{Adv}^{\text{k-cmt}}_{\text{XORP}}(\mathcal{C}) \leq \sum_{K \neq K'} \Pr\left[\mathsf{CP}_{K,K'}\right] \tag{10}$$

$$\leq \binom{p}{2} \cdot \prod_{i=0}^{w-1} \frac{2^\delta}{2^n - i} \tag{11}$$

$$< \binom{p}{2} \cdot \prod_{i=1}^{w} \frac{2^\delta}{2^n - i}$$

$$< \frac{p(p-1)}{2^{wn-\delta}} . \tag{12}$$

We justified (10) above. For (11), we have that for any particular X', the probability that $E_{K'}(X'_i) = E_K(X_i)$ for all i is

equal to $\prod_{i=0}^{w-1} 1/(2^n - i)$. A union bound over all of the 2^δ possible X' gives that

$$\Pr\left[\mathsf{CP}_{K,K'}\right] \le \prod_{i=0}^{w-1} \frac{2^\delta}{2^n - i} \,. \tag{13}$$

We can then use that any $\mathcal{C}$ using p queries can query at most $\binom{p}{2}$ unique pairs of keys K, K'. Finally to justify (12), we use Lem. 16. □

Theorem 19 (XORP is key and config committing)**.** *Let E be an ideal cipher with keylength k and blocklength $n \ge 7$, fix w, m, δ such that $n = m + r$ for $r = \lceil \log_2(w+1) \rceil$ and $w < 0.1\sqrt{2^n}$ and $\delta \in [0..m]$, and let* $\mathrm{XORP}\colon \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^{wn}$ *be the construction defined above. Then for any* $\text{gk-CMT}_{\mathrm{XORP},\delta}$ *adversary $\mathcal{C}$ making at most p ideal cipher queries it holds that*

$$\mathsf{Adv}^{\text{gk-cmt}}_{\mathrm{XORP},\delta}(\mathcal{C}) \le \frac{2p(p-1)}{2^{wn-\delta}} \,.$$

assuming $w < 2^{n-1} \cdot (\ln p)/p$.

The final restriction on w is a technicality of our proof and probably can be lifted, but we leave improving the bound as an open question. Nevertheless, even for $p = 2^{128}$ and $n = 128$ we have that w can be much larger than two (in fact, up to $w = 44$).

Proof: Let E_K denote the ideal cipher randomly chosen permutation for key K. To win the $\text{gk-CMT}_{\mathrm{CTR},\delta}$ game in the ideal cipher model, the adversary must query on keys K, K' and values X, X' such that either $K \ne K'$ or $X \ne X'$ and

$$E_K(X_0) \oplus E_K(X_i) = E_{K'}(X'_0) \oplus E_{K'}(X'_i) \quad \text{for} \quad 1 \le i \le w \,. \tag{14}$$

and $\Delta(X, X') \le \delta$.

First we observe that to win the $\text{gk-CMT}_{\mathrm{XORP},\delta}$ game, the adversary can output $K = K'$. Thus unlike for CTR, we need to address both the $K \ne K'$ and $K = K'$ cases. Let $\mathsf{win}_{K \ne K'}$ be the event that $\mathcal{C}$ outputs a winning (K, K, X, X') and let $\mathsf{win}_{K=K'}$ be the event that $\mathcal{C}$ outputs a winning (K, K', X, X') with $K \ne K'$. Both events are defined over the game $\text{gk-CMT}^{\mathcal{C}}_{\mathrm{XORP},\delta}$. Then

$$\mathsf{Adv}^{\text{gk-cmt}}_{\mathrm{XORP},\delta}(\mathcal{C}) \le \Pr\left[\mathsf{win}_{K \ne K'}\right] + \Pr\left[\mathsf{win}_{K=K'}\right] \,.$$

We now bound each term in turn.

Distinct keys case: Suppose $\mathcal{C}$ wins by outputting $K \ne K'$. Then for any $K \ne K'$, let CP be the event that at some point in the game a first query with key K' is made such that: (1) a previous query occurred for key K and that defined $E_K(X_i)$ for $i \in [1..w]$; and (2) the permutation $E_{K'}$ is chosen such that $E_{K'}(X_0) \oplus E_{K'}(X'_i) = E_K(X_0) \oplus E_K(X_i)$ for $i \in [1..w]$ and with $\Delta(X, X') \le \delta$. Note that here we are generously giving the adversary credit if they can query such a $E_{K'}$ on any point (either forward or inverse query).

We then have that

$$\Pr\left[\mathsf{win}_{K=K'}\right] \le \sum_{K \ne K'} \Pr\left[\mathsf{CP}_{K,K'}\right] \tag{15}$$

$$\le \binom{p}{2} \cdot \prod_{i=1}^{w} \frac{2^\delta}{2^n - i} \tag{16}$$

$$< \frac{p(p-1)}{2^{wn-\delta}} \,. \tag{17}$$

We justified (15) above. For (16), we have that for any particular X', the probability that $E_{K'}(X_0) \oplus E_{K'}(X'_i) = E_K(X_0) \oplus E_K(X_i)$ for all i is equal to $\prod_{i=1}^{w} 1/(2^n - i)$ because $E_K(X'_0)$ can be arbitrary and once fixed, there are $2^n - 1$ ways to choose $E_{K'}(X'_1)$, $2^n - 1$ ways to choose $E_{K'}(X_2)$ and so on. A union bound over all of the 2^δ possible X' gives that

$$\Pr\left[\mathsf{CP}_{K,K'}\right] \le \prod_{i=0}^{w-1} \frac{2^\delta}{2^n - i} \,. \tag{18}$$

We can then use that any $\mathcal{C}$ using p queries can query at most $\binom{p}{2}$ unique pairs of keys K, K'. Finally to justify (17), we use Lem. 16.

Same key case: Now we turn to the case that $\mathcal{C}$ wins by outputting $K = K'$. Let $\mathcal{C}$'s transcript of queries be $q^1, \ldots, q^p$ where each q_i is represented as a tuple $(b^\alpha, K^\alpha, U^\alpha, V^\alpha)$ where $b^\alpha = 1$ indicates it was a forward query to $E_K(U^\alpha) = V^\alpha$ and $b^\alpha = 0$ an inverse query $E_K^{-1}(V^\alpha) = U^\alpha$. We say a query is *useful* if $U = X \,\|\, \langle i \rangle$ for some $i \in [0..w]$. Queries that are not useful cannot appear in computation of XORP for winning outputs.

Then consider any useful query $(b^\beta, K, U^\beta, V^\beta)$ and let X^β be the first $n - r$ bits of U^β. Assume that there exists no query $\gamma < \beta$ in the transcript such that $U^\gamma = X_i^\beta$ for some $i \in [0..w]$. In words, β is the first time a query to E_K or E_K^{-1} involves X^β.

Now consider any useful query $(b^\alpha, K, U^\alpha, V^\alpha)$ with X^α the first $n - r$ bits of U^α and $\alpha < \beta$ and with $\Delta(X^\alpha, X^\beta) \le \delta$. There can be at most 2^δ such prior queries. Then we want to analyze the probability that α and β lead to a win, i.e., that $E_K(X_0^\beta) \oplus E_K(X_i^\beta) = E_K(X_0^\alpha) \oplus E_K(X_i^\alpha)$ for $i \in [1..w]$. Let $\mathsf{CP}_{\alpha,\beta}$ represent the event that α and β leads to a win. Let j be such that $X_j^\beta = U^\beta$, i.e., index j was the first queried. Then we have that

$$\Pr\left[\mathsf{CP}_{\alpha,\beta}\right] = \Pr\left[\wedge_{i=1}^{w} E_K(X_0^\beta) \oplus E_K(X_i^\beta) = E_K(X_0^\alpha) \oplus E_K(X_i^\alpha)\right]$$
$$< \left(\frac{1}{2^n - \beta}\right)^w$$

where we use that at most p queries have been made to E_K or its inverse and the fact that, by assumption, for each $0 < i \ne j$ the value $E_K(X'_i)$ is freshly chosen and if $j > 0$ then the value $E_K(X_0)$ is freshly chosen. In other words, for each of the w equalities, one n-bit value is freshly chosen. Given that there

are at most p queries, we have that

$$\begin{aligned}
\Pr\left[\mathsf{win}_{K\neq K'}\right] &\leq \sum_{\beta=2}^{p}\sum_{\alpha=1}^{2^\delta}\Pr\left[\mathsf{CP}_{\alpha,\beta}\right] \\
&< \sum_{\beta=2}^{p}\frac{2^\delta}{(2^n-\beta)^w} \\
&\leq \sum_{\beta=2}^{p}\frac{2^\delta}{2^n(1-\beta/2^n)^w} \\
&\leq \sum_{\beta=2}^{p}\frac{2^\delta e^{2wp/2^n}}{2^{wn}} \qquad (19)\\
&\leq \frac{e^{2wp/2^n}\cdot(p-1)}{2^{wn-\delta}} \\
&< \frac{p(p-1)}{2^{wn-\delta}}\,. \qquad (20)
\end{aligned}$$

where (19) uses that $(1-\beta/2^n) > e^{2\beta/2^n}$ and, finally, (20) uses the fact that since we assume $w < 2^{n-1}\cdot(\ln p)/p$ it holds that $e^{2wp/2^n} < p$.

□

G Proof of Thm. 6

We start by letting game G0 = mu-ROR1$_{\mathcal{S}}(\mathcal{A})$. Let G1 be the same as G0 except that we replace E with a family of random permutations. Then we have that

$$\Pr[\text{G0}\Rightarrow 1] \leq \Pr[\text{G1}\Rightarrow 1] + \mathsf{Adv}^{\text{mu-prp}}_{E}(\mathcal{B}) \qquad (21)$$

for the adversary $\mathcal{B}$ that runs G1 except that instead of initializing a new random permutation during Kg call it instead makes a call to its own key generation oracle. Subsequent queries by $\mathcal{A}$ to its encryption oracle are answered by replacing invocations of the random permutation with queries to $\mathcal{B}$'s oracle E. By construction, we have that $\Pr[\text{mu-PRP1}_E(\mathcal{B})\Rightarrow 1] = \Pr[\text{G0}\Rightarrow 1]$ and $\Pr[\text{mu-PRP0}_E(\mathcal{B})\Rightarrow 1] = \Pr[\text{G1}\Rightarrow 1]$. Moreover, $\mathcal{B}$ makes $\kappa = \sum_i \vec{q}_i$ key generation queries, and for $\vec{q}_i$ of these keys makes at most $i(L+2)$ queries to its oracle E for a total of at most $\sum_i \vec{q}_i\cdot i\cdot(L+2)$ queries. It runs in time that of $\mathcal{A}$ plus a small overhead.

Returning to our game hopping sequence, let G2 be the same as G1 except that encryption queries return a random (C,T) pair of the appropriate length. We now introduce a sequence of intermediate games G1$_j$ for $1\leq j\leq \kappa+1$. Game G1$_j$ is defined as G1 except that the first $j-1$ key instances are replaced with their idealized counterpart. That is, those key generation queries do not generate a new random permutation instance, and queries on those keys are returned with random (C,T) pairs of the appropriate lengths. Note that G1$_1$ = G1 and G1$_{\kappa+1}$ = G2. Let $\hat{q}_j$ be an upper bound on the number of queries made to key instance j. Then we have that

$$\Pr[\text{G1}_i\Rightarrow 1] \leq \Pr[\text{G1}_{i+1}\Rightarrow 1] + \frac{(\hat{q}_jL+\hat{q}_j+1)^2}{2^{n+1}}$$

by applying [38, Cor. 3]. Summing over the sequence of hybrids we have:

$$\Pr[\text{G1}\Rightarrow 1] \leq \Pr[\text{G2}\Rightarrow 1] + \sum_{j=1}^{\kappa}\frac{(\hat{q}_jL+\hat{q}_j+1)^2}{2^{n+1}}$$

Because $\mathcal{A}$ is $\vec{q}$-bound, we know the adversary makes at most i queries to $\vec{q}_i$ of the key instances. Thus we we can rewrite the summation according to $\vec{q}$, yielding

$$\Pr[\text{G1}\Rightarrow 1] \leq \Pr[\text{G2}\Rightarrow 1] + \sum_{i=1}\vec{q}_i\frac{(iL+i+1)^2}{2^{n+1}} \qquad (22)$$

Observing that G2 = mu-ROR0$_{\mathcal{S}}(\mathcal{A})$ and combining equations 21 and 22 yields the claimed advantage bound.

H Proof of Thm. 9

Let game G0 = ROR\$1$_{\mathcal{T}}(\mathcal{A})$, but with a flag bad set should an execution of Enc select a nonce N that (1) repeats or (2) has a nonce head NH that was selected in two previous queries. Let G1 be the same as G0 except that should bad be set, the game ends, outputting a random bit. Let G2 be the same as G1 except that ciphertext, tag pairs are randomly selected, and G3 be the same as G2 except that after bad is set it no longer halts. By construction G3 = ROR\$0$_{\mathcal{T}}(\mathcal{A})$. We have that

$$\begin{aligned}
\Pr[\text{G0}\Rightarrow 1] &\leq \Pr[\text{G1}\Rightarrow 1] + \Pr[\mathsf{bad}_0] \\
&\leq \Pr[\text{G2}\Rightarrow 1] + \mathsf{Adv}^{\text{ror}}_{\mathcal{T}}(\mathcal{B}) + \Pr[\mathsf{bad}_0] \\
&\leq \Pr[\text{G3}\Rightarrow 1] + \mathsf{Adv}^{\text{ror}}_{\mathcal{T}}(\mathcal{B}) + \Pr[\mathsf{bad}_0] + \Pr[\mathsf{bad}_3]
\end{aligned}$$

where bad_i denotes the event that bad is set in game Gi and $\mathcal{B}$ is the adversary that works as follows. It runs $\mathcal{A}$, simulating its encryption queries (A,M) as follows. It chooses a nonce $N \leftarrow\!\!\$\ \{0,1\}^{\nu}$, and parses N as $NH\,\|\,NT$. If N is a repeat of a prior query's or NH has been selected for two prior queries, $\mathcal{B}$ halts and outputs a random bit. Otherwise, it calls it's own encryption oracle $\text{Enc}(N,A,M)$ and returns the result. We have that $\Pr[\text{G1}\Rightarrow 1] = \Pr[\text{ROR1}_{\mathcal{T}}(\mathcal{B})\Rightarrow 1]$ and $\Pr[\text{G2}\Rightarrow 1] = \Pr[\text{ROR0}_{\mathcal{T}}(\mathcal{B})\Rightarrow 1]$, which justifies the second inequality above. Finally, we have that $\Pr[\mathsf{bad}_0] = \Pr[\mathsf{bad}_3]$, assuming without loss of generality that the same number queries are made in each game.

Using a union bound over the two conditions that trigger bad being set nonce collisions and nonce-head triple collisions (see App. A), we have that

$$\Pr[\mathsf{bad}_0] \leq \frac{q^2}{2^{\nu+1}} + \frac{q^3}{6\cdot 2^{2\ell_h}}\,.$$

Combining with the statements above yields the bound on $\mathcal{A}$'s advantage. Because $\mathcal{B}$ never queries a nonce twice, nor queries a nonce head three times, we have that $\mathcal{B}$ is $\vec{q}$-NH respecting for some $\vec{q}$ for which $q\geq\sum_i\vec{q}_i$ and $w(\vec{q})=2$.

I Proof of Thm. 11

Let $\mathcal{A}$ be a $\vec{q}$-bound mu-CTXT$_{\mathcal{S}}$ adversary for $\mathcal{S}$ being AES-GCM. We start by letting game G0 = mu-CTXT$_{\mathcal{S}}(\mathcal{A})$. Let G1 be the same as G0 except that we replace E with a family of random permutations. Then we have that

$$\Pr[\text{G0} \Rightarrow 1] \leq \Pr[\text{G1} \Rightarrow 1] + \mathsf{Adv}^{\text{mu-prp}}_{E}(\mathcal{B}) \qquad (23)$$

for the adversary $\mathcal{B}$ that runs G1 except that instead of initializing a new random permutation during a Kg call it instead makes a call to its own key generation oracle. Subsequent queries by $\mathcal{A}$ to its encryption oracle are answered by replacing invocations of the random permutation with queries to $\mathcal{B}$'s oracle E. By construction, we have that $\Pr[\text{mu-PRP1}_E(\mathcal{B}) \Rightarrow 1] = \Pr[\text{G0} \Rightarrow 1]$ and $\Pr[\text{mu-PRP0}_E(\mathcal{B}) \Rightarrow 1] = \Pr[\text{G1} \Rightarrow 1]$. Moreover, $\mathcal{B}$ makes the same number κ of key generation queries as $\mathcal{A}$, and makes at most $(L+2)(q+v)$ queries to its E where $q = \sum_i i\vec{q}_i$.

Now we turn to bounding the probability that win is set in game G1. We do so using a hybrid argument. Let win_i be the event in G1 that win was set on a decryption query to key instance i. Let $\hat{q}_i$ be the maximum number of encryption queries to instance i, let $\hat{v}_i$ be the maximum number of decryption queries to instance i, and $\hat{\sigma}_i$ be the total plaintext length (the sum of all encryption query's message sizes in blocks) for instance i. Then applying [38, Corollary 4] we have that

$$\Pr[\mathsf{win}_i] \leq \frac{\hat{v}_i(L_A+1)}{2^{\tau}} \cdot \delta_n(\hat{\sigma}_i + \hat{q}_i + \hat{v}_i + 1)$$

where $\delta_n(a) = (1-(a-1)/2^n)^{-a/2}$. Note that $v \geq \sum_i \hat{v}_i$. Thus we can simplify as follows:

$$\begin{aligned}
\Pr[\text{G1} \Rightarrow 1] &\leq \sum_{i=1}^{\kappa} \Pr[\mathsf{win}_i] \\
&\leq \sum_{i=1}^{\kappa} \frac{\hat{v}_i(L_A+1)}{2^{\tau}} \cdot \delta_n(\hat{\sigma}_i + \hat{q}_i + \hat{v}_i + 1) \\
&\leq \frac{v(L_A+1)}{2^{\tau}} \cdot \delta_n(w(\vec{q}) \cdot (L+1) + v + 1)\ .
\end{aligned}$$

J Proof of Thm. 12

We use a standard game-hopping argument. Let game G0 = CTXT$_{\mathcal{T}}(\mathcal{A})$. Then game G1 is the same as G0 except that we replace all calls to F with calls to a random function with the same domain and range as F. A standard reduction gives an PRF$_F$ adversary $\mathcal{D}$ such that $\Pr[\text{G0} \Rightarrow 1] \leq \Pr[\text{G1} \Rightarrow 1] + \mathsf{Adv}^{\text{prf}}_{F}(\mathcal{D})$, where $\mathcal{D}$ runs G0, replacing all executions of F with calls it its oracle. Hence $\mathcal{D}$ runs in time that of $\mathcal{A}$ and makes at most q queries.

Now in game G1 we are in position to reduce to an mu-CTXT$_{\mathcal{S}}$ adversary $\mathcal{B}$. It runs G1 except that for each distinct nonce head NH queried to either Enc or Dec it increments a counter (for tracking key instances), makes a call to its key generation oracle, and inserts the counter into into a new table entry $\mathbb{K}[NH]$. Encryption and decryption executions are replaced with calls to $\mathcal{B}$'s encryption and decryption oracles, using $\mathbb{K}[NH]$ as the key instance. Then by construction we have that $\Pr[\text{G1} \Rightarrow 1] = \Pr[\text{mu-CTXT}_{\mathcal{S}}(\mathcal{B}) \Rightarrow 1]$. Combining with the equations above gives the desired upper bound on $\mathcal{A}$'s advantage. Finally, if $\mathcal{A}$ is $\vec{q}$-NH-respecting then $\mathcal{B}$ is $\vec{q}$-bound: table $\mathbb{K}$ maps each nonce head to a particular user key.

K DNDKv1

Fig. 9 details the key derivation function of DNDKv1.

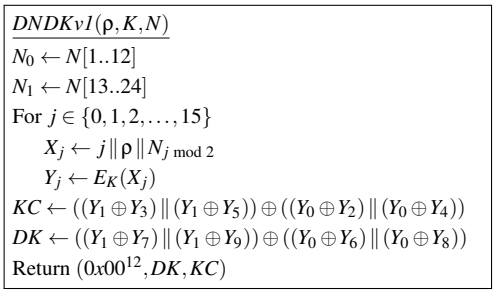

Figure 9: DNDKv1 KDF function specified in [28, 30].