CENCPP – Beyond-birthday-secure Encryption from Public Permutations

Arghya Bhattarcharjee[1], Avijit Dutta[1], Eik List[2], and Mridul Nandi[1]

[1] Indian Statistical Institute, Kolkata, India
bhattacharjeearghya29(at)gmail.com,
mridul.nandi(at)gmail.com,
[2] Indian Institute of Technology, Kharagpur, India
avirocks.dutta13(at)gmail.com,
[3] Bauhaus-Universität Weimar, Weimar, Germany
<firstname>.<lastname>(at)uni-weimar.de

Abstract. Public permutations have been established as valuable primitives since the absence of a key schedule compared to block ciphers alleviates cryptanalysis. While many permutation-based authentication and encryption schemes have been proposed in the past decade, the birthday bound in terms of the primitive's block length n has been mostly accepted as the standard security goal. Thus, remarkably little research has been conducted yet on permutation-based modes with higher security guarantees. Only recently at CRYPTO'19, Chen et al showed two constructions with higher security based on the sum of two public permutation. Their work has sparked increased interest in this direction by the community. However, since their proposals were domain-preserving, the question of encryption schemes with beyond-birthday-bound security was left open.

This work tries to address this gap by proposing CENCPP, a nonce-based encryption scheme from public permutations. Our proposal is a variant of Iwata's block-cipher-based mode CENC that we adapt for public permutations, thereby generalizing Chen et al.'s Sum-of-Even-Mansour construction to a mode with variable output lengths. Like CENC, our proposal enjoys a comfortable rate-security trade-off that needs $w + 1$ calls to the primitive for w primitive outputs. We show a tight security level for up to $O(2^{2n/3}/w^2)$ primitive calls. While $w \geq 1$ can be arbitrary, two independent keys suffice; moreover, although we propose CENCPP first in a generic setting with $w + 1$ independent permutations, we show that only $\log_2(w + 1)$ bits of the input for domain separation suffice to obtain a single-permutation variant that still maintains a security level of up to $O(2^{2n/3}/w^4)$ queries.

Keywords: Symmetric-key cryptography · permutation · provable security.

1 Introduction

Permutation-based Cryptography has been established as an important branch of symmetric-key cryptography during the 2010s decade since they avoid the task of designing and analyzing a secure key schedule. After the selection of Keccak as SHA-3 standard [NIS15], permutations have found their way into manyfold applications beyond hashing, such as encryption (e.g., OPP [GJMN16]), authentication (Chaskey [MMH$^+$14]), or authenticated encryption (e.g., NORX [AJN14], Ascon [DEMS16], Ketje [BDP$^+$16], or STRIBOB [SB15]).

The Security of Many Block-cipher-based Modes such as GCM [MV04] or OCB3 [KR11] is limited by the birthday bound of the primitive's state size (usually indicated by n bits). This limitation renders the privacy guarantees void when some internal collision occurs, which happens with non-negligible probability after $O(2^{n/2})$ blocks have been processed under the same key. While this level of security is often sufficient, it can become problematic for settings that need primitives with small block lengths [BL16], or for applications that employ large amounts of data. Moreover, higher security is not undesirable in the approaching era of quantum computers.

In the domain of block ciphers, the community has consequently proposed various modes with higher guarantees over the previous decades, e.g., CENC [Iwa06] or the Sum of GCM [IM16], just to name examples. Moreover, the usage of tweakable block ciphers (TBCs) [LRW02], that take an additional public input called tweak, has allowed the construction of modes with significantly enhanced security guarantees. For example, the modes ΘCB3 [KR11] or $\mathbb{OTR}$ [Min14] can overcome the birthday bound with appropriate primitives. As a result, a series of research introduced highly secure encryption modes [PS16], MACs [IMPS17,Nai15], and AE schemes [BGIM19,PS16] based upon them.

For Permutation-based Modes, the birthday-bound limitation has been usually tolerated, e.g. in Farfalle [BDH$^+$17] or OPP [GJMN16]. This generic lack of security and efficiency has been compensated by using permutations with larger state sizes. Moreover, it is not easy to overcome the birthday bound when the primitive is public due to the existence of well-known generic attacks e.g. [DDKS13,DKS12]. Various approaches tried to increase the security by multiple calls to the primitive, e.g., in multiple rounds of Even-Mansour constructions [CLL$^+$14,CLM19,CS14,CS15].

Nevertheless, permutation-based modes do not have to be limited in general. Often, the designers of permutation-based schemes have argued that the mere size of their underlying permutation renders birthday attacks infeasible – a valid and pragmatic argument. However, an equally pragmatic argument is the fact that the state size of current permutations poses considerable costs either to implementation size, area, or performance. Therefore, efficient permutation-based modes with higher security appear attractive, be it with some restrictions such as the need for multiple keys.

One step in this direction has been taken recently by Chen et al. [CLM19] who proposed two permutation-based PRFs, called Sum-of-Even-Mansour constructions (SoEM) and Sum-of-Key-alternating-Ciphers. They provided security proofs for both with up to $O(2^{2n/3})$ queries. Still, their constructions map only fixed-length inputs to fixed-length outputs, which left the question of designing an encryption scheme with similar security still open.

This Work tries to move a step forwards in this direction. We propose $\mathsf{CENCPP}[w]$, a mode from n-bit permutations with $O(2^{2n/3}/w^2)$ security where w is a small user-chosen integer. Our proposal is a straight-forward adaption of Iwata's CENC mode [Iwa06]. So, this represents a trade-off, where w can be chosen to be still below the usual number of round keys e.g. for the AES [NIS01] or $\mathsf{Deoxys\text{-}BC}$ [JNP14]. It can be instantiated directly with usual permutations such as Keccak-f and requires only two independent keys.
While our generic proposal of $\mathsf{CENCPP}[w]$ considers $(w+1)$ independent permutations, we suggest a variant that needs only a single public permutation while sacrificing only $\log_2(w+1)$ bits of the input space for separating domains. That is, we derive domain-separated single-primitive variants of SoEM and CENCPP, that we call $\mathsf{DS\text{-}SoEM}$ and $\mathsf{DS\text{-}CENCPP}[w]$, and show their security. We show that two independent keys are sufficient and necessary for our security guarantees by providing also distinguishers for all constructions in $O(2^{n/2})$ if single keys or simpler key scheduling approaches would be taken. Moreover, we provide distinguishers in $O(2^{2n/3})$ queries to note that the security is effectively tight except the logarithmic factor in w.

The Remainder is structured as follows: Section 2 recalls necessary preliminaries before Section 3 defines CENCPP. We employ two different keys for our security, and show that it is necessary to combine the keys for most primitive calls. Our approach of using keys is necessary. We show that a simpler key scheduling would lead to a birthday-bound distinguisher in Section 4. Next, the security of the generic CENCPP is analyzed in Section 5. In Section 6, we propose domain-separated variants of SoEM and CENCPP, called $\mathsf{DS\text{-}SoEM}$ and $\mathsf{DS\text{-}CENCPP}$. We provide a design rationale with brief description of distinguishers on weaker variants in Section 7. Thereupon, we analyze first $\mathsf{DS\text{-}SoEM}$ in Section 8 and 9. Finally, Section 10 concludes this work.

2 Preliminaries

In General, we will use lowercase letters x, y for indices and integers, uppercase letters X, Y for binary strings and functions, calligraphic uppercase letters $\mathcal{X}, \mathcal{Y}$ for sets and spaces. We write $\mathbb{F}_2$ for the finite field of characteristic 2 and $\mathbb{F}_2^n$ for an n-element vector of elements in $\mathbb{F}_2$, or bit strings. $X \parallel Y$ denotes the concatenation of binary strings X and Y, and $X \oplus Y$ for their bitwise XOR, that is, addition in $\mathbb{F}_2$. We indicate the length of X in bits by $|X|$, and write X_i for the i-th block. We denote by $X \twoheadleftarrow \mathcal{X}$ that X is chosen uniformly at random from the set $\mathcal{X}$. We define $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$ for the set of all functions $F : \mathcal{X} \to \mathcal{Y}$, $\mathsf{Perm}(\mathcal{X})$ for the set of all permutations $\pi : \mathcal{X} \to \mathcal{X}$, and $\widetilde{\mathsf{Perm}}(\mathcal{T}, \mathcal{X})$ for the set of tweakable permutations $\widetilde{\pi} : \mathcal{T} \times \mathcal{X} \to \mathcal{X}$ over $\mathcal{X}$ with tweak space $\mathcal{T}$. We define by $X_1, \ldots, X_j \xleftarrow{x} X$ an injective splitting of a string X into blocks of

x-bit such that $X = X_1 \parallel \cdots \parallel X_j$, $|X_i| = x$ for $1 \leq i \leq j-1$, and $|X_j| \leq x$. For positive integer m, we use $\mathcal{X}^{\leq m} =^{\text{def}} \bigcup_{i=0}^{m} \mathcal{X}^i$. By $\langle X \rangle_n$, we denote the encoding of an integer X into an n-bit string, e.g., $\langle 135 \rangle_8 = (10000111)_2$. For any n-bit string $X = (X[n-1] \ldots X[1]X[0])$ and non-negative integer $x \leq n$, let $\mathsf{lsb}_x(X)$ and $\mathsf{msb}_x(X)$ denote the functions that return the x least significant and most significant bits of X, respectively. We omit writing n if it is clear from the context. For $q \in \mathbb{N}$, we define $[q] =^{\text{def}} \{1, \ldots, q\}$ and $[0..q] =^{\text{def}} \{0, \ldots, q\}$. Given a vector space $\mathcal{V} \subseteq \mathbb{F}$ of a field $\mathbb{F}$, and an element $\alpha \in \mathcal{K}$, we define the space $\alpha \cdot \mathcal{V} =^{\text{def}} \{\alpha \cdot V : V \in \mathcal{V}\}$. Moreover, given two spaces $\mathcal{V}, \mathcal{W} \subset \mathbb{F}$, we define by $\mathcal{V} + \mathcal{W} =^{\text{def}} \{V \in \mathcal{V}, W \in \mathcal{W} : V + W\}$, where addition is in $\mathbb{F}$.

A Distinguisher $\mathbf{A}$ is an efficient Turing machine that interacts with a given set of oracles that appear as black boxes to $\mathbf{A}$. We write $\Delta_{\mathbf{A}}\left(\mathcal{O}^1; \mathcal{O}^2\right)$ for the advantage of $\mathbf{A}$ to distinguish between oracles $\mathcal{O}^1$ and $\mathcal{O}^2$. All probabilities are defined over the random coins of the oracles and those of the adversary, if any. We write $\mathbf{Adv}_F^X(q, \sigma) =^{\text{def}} \max_{\mathbf{A}}\{\mathbf{Adv}_F^X(\mathbf{A})\}$ for the maximum over all X-adversaries $\mathbf{A}$ on F that ask at most q queries of at most σ blocks in total to its oracles. W.l.o.g., we assume that $\mathbf{A}$ never asks queries to which it already knows the answer.
Note that hereafter, we consider information-theoretic distinguishers $\mathbf{A}$, whose resources are upper bounded only in terms of their maximal numbers of queries and blocks that they can ask to their available oracles. One can derive the computation-theoretic counterparts in straight-forward manner. We parametrize our distinguishers, where we use q_c for the number of queries to a construction and σ to the total number of blocks to the construction. Since we analyze constructions $\Pi[\pi_0, \ldots, \pi_w]$ based on public permutations $\pi_0, \ldots, \pi_w$ in the ideal-permutation model, we further use q_p for the number of queries to the primitive oracles.

For primitives, we will often use sets the $\mathcal{K} = \mathbb{F}_2^n$ for keys, $\mathcal{B} = \mathbb{F}_2^n$ for message blocks, $\mathcal{N} = \mathbb{F}_2^\nu$ for nonces, and $\mathcal{D} = \mathbb{F}_2^\mu$ for counters, where n, ν, μ are small integers.

PRF Security refers to the maximal advantage of distinguishing the outputs of a scheme from random bits of the expected length. Given two non-empty sets or spaces $\mathcal{X}, \mathcal{Y}$, let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a function, $\rho \twoheadleftarrow \mathsf{Func}(\mathcal{X}, \mathcal{Y})$ and $K \twoheadleftarrow \mathcal{K}$ be a secret key. Then, the PRF advantage of $\mathbf{A}$ is defined as

$$\mathbf{Adv}_{F_K}^{\mathsf{PRF}}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}\left(F_K; \rho\right).$$

A Nonce-based Encryption Scheme $\Pi = (\mathcal{E}, \mathcal{D})$ is a tuple of algorithms for encryption and decryption with signatures $\mathcal{E} : \mathcal{K} \times \mathcal{N} \times \mathbb{F}_2^* \to \mathbb{F}_2^*$ and $\mathcal{D} : \mathcal{K} \times \mathcal{N} \times \mathbb{F}_2^* \to \mathbb{F}_2^*$, where $\mathcal{N}$ denotes a nonce space. The nonce $N \in \mathcal{N}$ must not repeat over all encryption queries. Distinguishers that obey this requirement are called nonce-respecting. We assume that Π is correct, i.e., for all $K, N, M \in \mathcal{K} \times \mathcal{N} \times \mathbb{F}_2^*$, it holds that $\mathcal{D}_K(N, \mathcal{E}_K(N, M)) = M$.

Let $K \twoheadleftarrow \mathcal{K}$ and $\rho : \mathcal{N} \times \mathbb{F}_2^* \to \mathbb{F}_2^*$ be a function that, on input (N, M), computes $C \leftarrow \mathcal{E}_K(N, M)$ for random $K \twoheadleftarrow \mathcal{K}$ and outputs $C' \twoheadleftarrow \mathbb{F}_2^{|C|}$. The nE-security of a nonce-respecting distinguisher $\mathbf{A}$ is defined as

$$\mathbf{Adv}_{\Pi_K}^{\mathsf{nE}}(\mathbf{A}) \overset{\text{def}}{=} \underset{\mathbf{A}}{\Delta}\left(\mathcal{E}_K; \rho\right).$$

In The Ideal-permutation Model, the distinguisher has one or multiple additional oracles $\pi^{\pm}$ that provides access to the public permutation π in forward and backward direction. Since this work studies schemes based on public permutations, we study the security notions such as PRF and nE security to the ideal-permutation model. We write $\Pi[\pi]$ and $\mathcal{E}[\pi]$, $\mathcal{D}[\pi]$, etc. to indicate that an encryption scheme Π and its algorithms are based on a primitive π.

The H-coefficient Technique is a proof method by Patarin [Pat08,Pat10]. Here, we refer to the modernized variant by Chen and Steinberger [CS14]. Let $\mathbf{A}$ be a distinguisher that interacts with its oracles $\mathcal{O}$ and obtains outputs from a real world $\mathcal{O}_{\text{real}}$ or an ideal world $\mathcal{O}_{\text{ideal}}$. The results of the interaction are collected in a transcript or view τ. The oracles can sample random coins before the experiment (often a key or an ideal primitive that is sampled beforehand), and are then deterministic [CS14]. We choose two random variables Θ_{real} for the distribution of transcripts in the real world and correspondingly Θ_{ideal} for that in the ideal world, respectively. A transcript τ is called attainable if $\mathbf{A}$ can observe τ with non-zero probability in the ideal world. The fundamental Lemma of the H-coefficients technique, whose proof can be found e.g., in [CS14,Pat08], states that we can split the set of all attainable transcripts into two disjoint sets $\textsc{GoodT}$ and $\textsc{BadT}$ and bound the distinguishing advantage as follows:

Lemma 1 (Fundamental Lemma o. t. H-coefficient Technique [Pat08]). Assume, there exist $\epsilon_1, \epsilon_2 \geq 0$ such that for any transcript $\tau \in \textsc{GoodT}$, it holds

$$\frac{\Pr\left[\Theta_{\text{real}} = \tau\right]}{\Pr\left[\Theta_{\text{ideal}} = \tau\right]} \geq 1 - \epsilon_1 \quad \text{and} \quad \Pr\left[\Theta_{\text{ideal}} \in \textsc{BadT}\right] \leq \epsilon_2 .$$

Then, for all distinguishers $\mathbf{A}$, it holds that $\Delta_{\mathbf{A}}\left(\mathcal{O}_{\text{real}}; \mathcal{O}_{\text{ideal}}\right) \leq \epsilon_1 + \epsilon_2$.

The technique has been generalized by Hoang and Tessaro [HT16] in their expectation method, which allowed to derive the fundamental lemma as a corollary.

The Sum-capture Lemma by Chen et al. [CLL+14] states the following.

Lemma 2. Let $n, q \in \mathbb{N}$ such that $9n \leq q \leq 2^{n-1}$. Let $\mathcal{T} = \{T^1, \ldots, T^q\} \subseteq \mathbb{F}_2^n$ such that the values T^i for $i \in [q]$ are with-replacement samples from $\mathbb{F}_2^n$. Let $\mathcal{X}, \mathcal{Y} \subset \mathbb{F}_2^n$ be arbitrary and $\mathcal{S} =^{\text{def}} \{(T, X, Y) \in \mathcal{T} \times \mathcal{X} \times \mathcal{Y} : T = X \oplus Y\}$. Then, it holds that

$$\Pr\left[|\mathcal{S}| \geq \frac{q|\mathcal{X}||\mathcal{Y}|}{2^n} + 3\sqrt{nq|\mathcal{X}||\mathcal{Y}|}\right] \leq \frac{2}{2^n} .$$

where the randomness is defined over $\mathcal{T}$.

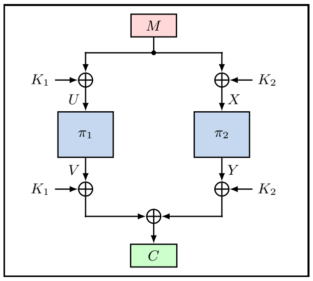


Fig. 1: The construction SoEM22 by Chen et al. [CLM19], with two independent permutations π_i and independent keys each.

3 The CENCPP Mode

This section will define the generic CENCPP. Standing on the shoulders of existing constructions, we prior give the necessary details of SoEM and CENC first.

3.1 SoEM

At CRYPTO'19, Chen et al. [CLM19] proposed two constructions with beyond-birthday-bound security from two public permutations each: SoEM (Sum of Even-Mansour constructions) and SoKAC (Sum of Key-alternating Ciphers). Both designs represent fixed-length PRFs which they provided analyses for up to $O(2^{2n/3})$ queries for both. An improved analysis that showed subtleties of the proof of SoKAC 21 was presented later in [Nan20]. The former sums the results of two single-round Even-Mansour ciphers, whereas the latter defines a variant of Encrypted Davies-Meyer [MN17a] from public instead of keyed primitives.
Chen et al. parametrized their constructions as SoEM$\lambda\kappa$ and SoKAC$\lambda\kappa$, where λ denoted the number of permutations, and κ the number of keys. Figure 1 illustrates SoEM22, which will be relevant in this work. Both modes require two calls to the public permutations and two independent permutations. Moreover, SoEM demanded two independent keys. Chen et al. considered SoEM12 with a single permutation: $\pi(M \oplus K_1) \oplus K_1 \oplus \pi(M \oplus K_2) \oplus K_2$, and SoKAC12 as $\pi(\pi(M \oplus K_1) \oplus K_2) \oplus K_1 \oplus \pi(M \oplus K_1) \oplus K_2$, and showed distinguishers with $O(2^{n/2})$ queries for both.

3.2 CENC

CENC is a nonce-based block-cipher-based mode that generalizes the sum of permutations by Iwata [Iwa06]. It uses the nonce concatenated with a counter as block cipher input, splits each sequence of w subsequent message blocks into chunks, and processes them by XORP.

In XORP, the message M is split into w blocks of n bits, for a small positive integer w. Let n, ν, μ be integers such that $n = \nu + \mu$ and $w + 1 \leq 2^{\mu}$. Let

$E : \mathcal{K} \times \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a block cipher, and let $\mathcal{N} = \mathbb{F}_2^\nu$ be a nonce space. The remaining μ bits of the input are used for a counter. Let $K \in \mathcal{K}$ be a secret key and $N \in \mathcal{N}$ be a nonce. Then, $\mathsf{XORP}[E_K, w](N, s)$ computes a key stream $S_1 \parallel \ldots \parallel S_w$ as

$$S_i \overset{\text{def}}{=} E_K(N \parallel \langle s \rangle_\mu) \oplus E_K(N \parallel \langle s + i \rangle_\mu), \text{ for } i \in [w] .$$

Thus, it makes $w + 1$ block-cipher calls with pairwise distinct inputs, where $E_K(X \parallel \langle s \rangle_\mu)$ with the starting value s of the counter is XORed to each of the other blocks. $\mathsf{XORP}[E_K, w]$ can be simply used as a length-restricted encryption scheme by XORing its output to a message M of $|M| \leq n \cdot w$ bits. The final chunk is simply truncated to the length of the final message block. We slightly adapt the definition by [Iwa06,IMV16] to

$$\mathsf{XORP}[E_K, w] : \mathcal{N} \times \mathbb{F}_2^\mu \to (\mathbb{F}_2)^{n \cdot w},$$

where $\mathsf{XORP}[E_K, w](N, i)$ uses $N \parallel \langle i \rangle_\mu$, $N \parallel \langle i + 1 \rangle_\mu$, …instead of $N \parallel \langle 0 \rangle_\mu$, $N \parallel \langle 1 \rangle_\mu$, …as inputs to E_K.

CENC concatenates several instances of $\mathsf{XORP}[E_K, w]$ with pair-wise distinct inputs. Let $M \in \mathbb{F}_2^*$ be a message s. t. $(M_1 \parallel \ldots \parallel M_m) \xleftarrow{n} M$. Let $\ell = \lceil m/w \rceil$ denote the number of chunks. It must hold that $\ell \cdot (w + 1) < 2^\mu$. Then

$$\mathsf{CENC}[E_K, w](N, M) \overset{\text{def}}{=} \mathsf{msb}_{|M|} \left(\parallel_{i=0}^{\ell-1} \mathsf{XORP}[E_K, w] \left(N, i \cdot (w + 1)\right)\right) \oplus M.$$

3.3 CENCPP

In the following, we adapt CENC to the public-permutation setting. Let π_0, …, $\pi_w \in \mathsf{Perm}(\mathbb{F}_2^n)$ be permutations, and let $K_0, K_1 \in \mathbb{F}_2^n$ be independent secret keys. We define $\boldsymbol{\pi} =^{\text{def}} (\pi_0, \ldots, \pi_w)$ as shorthand form. Furthermore, $\mathcal{D} \subseteq \mathbb{F}_2^\mu$ be a set of domains, s. t. $n = \nu + \mu$. For brevity, we define a key vector $\mathbf{K} = (K_0, K_1)$. We combine both keys K_0 and K_1 for the individual permutations as $2^i K_0 \oplus 2^{2i} K_1$, for all $i \in [0..w]$, where we assume that the element 2 is primitive in $\mathbb{F}_{2^n}$.
We adapt XORP to XORPP to note that it is based on the XOR of **public** permutations. We define $\mathsf{XORPP}[\boldsymbol{\pi}, w] : (\mathbb{F}_2^n)^2 \times \mathbb{F}_2^n \to (\mathbb{F}_2^n)^w$, instantiated with $w + 1$ permutations π_0, …, π_w, a key space $(\mathbb{F}_2^n)^2$ as given in Algorithm 1. We write XORPP as short for $\mathsf{XORPP}[\boldsymbol{\pi}, w]$ when w and the permutations are clear from the context. Given that the permutations are independent, CENCPP use the same input $(N \parallel \langle i \rangle_\mu)$ for each permutation call in one call of XORPP. Then, we define the encryption and decryption algorithms $\mathcal{E}$ and $\mathcal{D}$ of the nonce-based mode CENCPP as given in Algorithm 1.

3.4 Discussion

Further Constructions with beyond-birthday security from public permutations are naturally possible. However, our proposal of CENCPP seems very efficient.

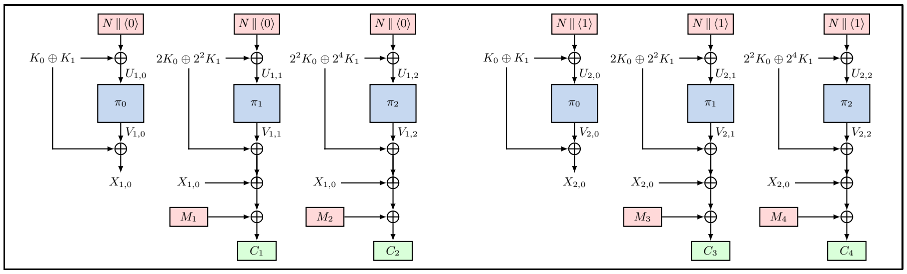


Fig. 2: Schematic illustration of the encryption of a four-block message $M = (M_1, \ldots, M_4)$ with $\mathsf{CENCPP}[(\pi_0, \pi_1, \pi_2), 2]_{K_0,K_1}$. The final chunk is simply truncated if its length is less than $2n$ bits. N is a nonce, K_0 and K_1 are independent secret keys and π_0, π_1, and π_2 are independent public permutations.

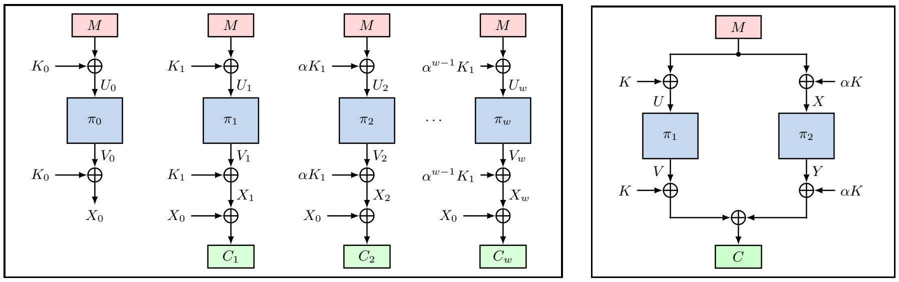
Algorithm 1 Definition of CENCPP.

```
function CENCPP[$\boldsymbol{\pi}, w$].$\mathcal{E}_{\mathbf{K}}(N, M)$
    $(M_1, \ldots, M_m) \xleftarrow{n} M$
    $\ell \gets \lceil m/w \rceil$
    for $i \gets 0..\ell - 1$ do
        $j \gets i \cdot w$
        $(S_{j+1} \parallel \cdots \parallel S_{j+w})$
            $\gets \mathsf{XORPP}[\boldsymbol{\pi}, w]_{\mathbf{K}}(N \parallel \langle i \rangle_\mu)$
        for $k \gets j + 1..j + w$ do
            $C_k \gets \mathsf{msb}_{|M_k|}(S_k) \oplus M_k$
    return $(C_1 \parallel \cdots \parallel C_m)$

function CENCPP[$\boldsymbol{\pi}, w$].$\mathcal{D}_{\mathbf{K}}(N, C)$
    return $\mathsf{CENCPP}[\boldsymbol{\pi}, w].\mathcal{E}_{\mathbf{K}}(N, C)$

function XORPP[$\boldsymbol{\pi}, w$]$_{\mathbf{K}}(M)$
    $(K_0, K_1) \gets \mathbf{K}$
    $(\pi_0, \ldots, \pi_w) \gets \boldsymbol{\pi}$
    $U_0 \gets M \oplus (K_0 \oplus K_1)$
    $X_0 \gets \pi_0(U_0) \oplus (K_0 \oplus K_1)$
    for $j \gets 1..w$ do
        $L_j \gets (2^j \cdot K_0) \oplus (2^{2j} \cdot K_1)$
        $U_j \gets M \oplus L_j$
        $X_j \gets \pi_j(U_j) \oplus L_j$
        $C_j \gets X_j \oplus X_0$
    return $(C_1 \parallel \cdots \parallel C_w)$
```

Instantiating $\mathsf{CENC}[\pi, w]$ with a two-round Even-Mansour construction could be an obvious generic way. This approach can provide approximately the security of the primitive, i.e. $2n/3$ bits, and would employ $\lceil 2\frac{w+1}{w} \rceil$ calls to the permutation for w message blocks.

In their proposal of $\mathsf{AES\text{-}PRF}$, Mennink and Neves therefore increased the performance of their construction [MN17b] by instantiating it with five-round AES. However, its security margin is thin [DIS$^+$18], so that this approach bears some risk of breaking from improved cryptanalysis in the close future.

More Related Works exist in the secret-permutation setting. Cogliati and Seurin [CS18] showed that a variant of EDM with a single keyed permutation – that is $E_K(E_K(M) \oplus M)$ – possesses roughly $O(2^{2n/3})$ security. The work by Guo et al. [GSWG19] followed this direction, showing $O(2^{2n/3}/n)$ security for the single-permutation variants of EDM and its dual EDMD– $E_K(E_K(M)) \oplus E_K(M)$. Moreover, they proved a similar security result also for the sum from a single permutation and its inverse, SUMPIP: $E_K(M) \oplus E_K^{-1}(M)$. This reminds of the Decrypted Wegman-Carter Davies-Meyer construction by Datta et al. [DDNY18] that would also possess a security bound of $O(2^{2n/3})$ but limited the input space to a $2n/3$-bit message. SUMPIP could retain beyond-birthday-bound security



(a) XORPP′.

(b) SoEM′.

Fig. 3: Example of using a weak key schedule for XORPP and SoEM.

with public permutations, i.e.

$$\pi(M \oplus K_1) \oplus K_1 \oplus \pi^{-1}(M \oplus K_2) \oplus K_2$$

could be secure beyond $O(2^{n/2})$ queries when using a public primitive π. However, such an instantiation would need both encryption and decryption direction of the primitive implemented, which is less practical than a construction that only needs a single direction. Plus, for CENCPP, we are unaware how this instantiation would help since it needs at least three independent permutations. We leave the security of modes similar to SUMPIP as an open question.

4 Distinguisher on Low-rank Key Schedules of XORPP

The keys K_0 and K_1 are combined a non-intuitive way in CENCPP, using $\alpha^i K_0 \oplus \alpha^{2i} K_1$ for $i \in [0..w]$. In general, this approach of a key schedule can be described as the multiplication with a Vandermonde matrix $\mathbf{L}$

$$\underbrace{\begin{bmatrix} 1 & \alpha^1 & \alpha^2 & \cdots & \alpha^w \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2w} \end{bmatrix}^\top}_{\mathbf{L}} \cdot \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}$$

where the elements are in $\mathbb{F}_{2^n}$, and $\alpha \in \mathbb{F}_{2^n}$ is a primitive element, which is often $\alpha = 2$, that is the polynomial $\mathtt{x}^1$ for practical values of $\mathbb{F}_{2^n}$. We assume $p(\mathtt{x})$ is some irreducible modulus polynomial in $\mathbb{F}_{2^n}$. Having SoEM as base, it is more tempting to employ a key scheduling of K_0, K_1, αK_1, $\alpha^2 K_1$, $\ldots$, instead, that is omitting the addition of K_0 for all subsequent permutation calls. In matrix form, such a key schedule would be written as

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \alpha & \cdots & \alpha^{w-1} \end{bmatrix}^\top}_{\mathbf{L}'} \cdot \begin{bmatrix} K_0 \\ K_1 \end{bmatrix} .$$

While the latter appears much simpler, after transposing its matrix form to $w+1$ rows, it contains rows that are not independent. We can exploit this fact by cancelling two outputs so that the distinguishing problem reduces to that for single-key SoEM. Since the steps are not intuitive, we illustrate the birthday-bound distinguisher in the following. We denote CENCPP with the key-schedule matrix $\mathbf{L}'$ as CENCPP' that uses it in calls to a variant of XORPP with this key schedule. Let us call that XORPP' for consistency. First, we show that we can reduce the security of CENCPP' to the security of SoEM with the key usage of $K, \alpha \cdot K$, as illustrated in Figure 3b. Similarly, we denote that variant as $\mathsf{SoEM}' =^{\text{def}} \mathsf{SoEM}[\pi_1, \pi_2]_{K,\alpha K}$.

4.1 Reduction to SoEM'

Suppose, $\mathbf{A}$ is an information-theoretic distinguisher on SoEM'. We have a transcript $\tau = \{K\} \cup \tau_p \cup \tau_c$, consisting of the key and two disjoint parts of the transcript. The primitive-query transcript τ_p consists of q_p chosen primitive queries and their corresponding responses (U^i, V^i) to π_1 and (X^k, Y^k) to π_2 each. The construction-query transcript τ_c consists of q_c chosen construction queries and their corresponding responses (M^j, C^j). After the interaction with its oracles, $\mathbf{A}$ is given the transcript, that is, also the key $K \twoheadleftarrow \mathbb{F}_{2^n}$. $\mathbf{A}$ sees $C = W \oplus Z$ where

$$\begin{aligned} W &\overset{\text{def}}{=} \pi_1(M \oplus K) \oplus K \\ Z &\overset{\text{def}}{=} \pi_2(M \oplus (\alpha \cdot K)) \oplus (\alpha \cdot K). \end{aligned}$$

In comparison, an adversary $\mathbf{A}'$ on $\mathsf{CENCPP}[\pi_0, \pi_2, \pi_2]_{K_0,K_1}$ with key schedule as illustrated in Figure 3a can compute $C_1 \oplus C_2 = (X_1 \oplus X_0) \oplus (X_2 \oplus X_0) = W \oplus Z = C$. Thus, it holds that

$$\mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{CENCPP}[\pi_0,\ldots,\pi_w,w]_{K_0,K_1}}(\mathbf{A}') \geq \mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{SoEM}'}(\mathbf{A}),$$

where $\mathbf{A}$ and $\mathbf{A}'$ ask the same number of construction queries q_c and primitive queries q_p to each of the primitives.

4.2 Distinguisher on SoEM'

Next, we consider an information-theoretic distinguisher on SoEM'. Its goal is to find vector spaces $\mathcal{U}, \mathcal{X}, \mathcal{M} \subset \mathbb{F}_{2^n}$ of cardinalities $|\mathcal{U}|, |\mathcal{X}|, |\mathcal{M}| \in O(2^{n/2})$ such that $\mathcal{M} + \mathcal{U} = \mathcal{M} + \mathcal{X} = \mathbb{F}_{2^n}$. This ensures that there exist $U \in \mathcal{U}$, $X \in \mathcal{X}$, and $M \in \mathcal{M}$ such that $M + U = K$ and $M + X = \alpha \cdot K$, where additions and multiplication are in the field. Once those spaces have been found, $\mathbf{A}$ can proceed as follows:

1. It queries all values $U^i \in \mathcal{U}$ to its primitive oracle π_1, and stores them together with the corresponding responses V^i, (U^i, V^i), into a list.
2. Similarly, it queries all values $X^k \in \mathcal{X}$ to its primitive oracle π_2, and stores them together with the corresponding responses Y^k, (X^k, Y^k), into a second list.

3. Moreover, it queries all values $M^j \in \mathcal{M}$ to its construction oracle and stores the tuples (M^j, C^j) into a third list.
4. It repeats the queries for the cosets $\mathcal{U} + c$, $\mathcal{X} + c$, and $\mathcal{M} + c$, where addition is in the field. This means, **A** asks the queries $U^i \oplus c$, $X^k \oplus c$, and $M^j \oplus c$, where $c \in \mathbb{F}_{2^n}$ is an arbitrary non-zero constant, for all $i, j, k \in [2^{n/2}]$. Define $M'^j = M^j \oplus c$, $U'^i = U^i \oplus c$ and $X'^k = X^k \oplus c$.
5. If there exist tuples (M^j, U^i, X^k) such that $C_1^j \oplus C_2^j = (U^i \oplus V^i) \oplus (X^k \oplus Y^k)$ and $C'^j_1 \oplus C'^j_2 = (U'^i \oplus V'^i) \oplus (X'^k \oplus Y'^k)$, output real, and random otherwise.

For the real construction, the success probability is one. For the random construction, the probability to produce two pairs at random is approximately $O(2^{-n})$ since we can build $O(2^{n/2} \cdot 2^{n/2})$ pairs that match both n-bit equations with probability 2^{-n} each.

Remark 1. In the following, we show an example which works if the message length is even, which is natural since n is usually even in practice. We exemplify it for the case when $\alpha = 2$. The distinguisher works since the message space of $\mathcal{U} + \mathcal{X} = \mathcal{X} + 2\mathcal{X} = 3\mathcal{X} = \mathcal{M}$ covers $O(2^{n/2})$ messages M only and still can cover all key combinations. A general approach is possible for all values of $\alpha = 2^i$ for any $i < n$, even for the cases when $U + X = 2^i \cdot K$ or $U + X = (2^i + 2^j) \cdot K$. However, we are not aware yet of how to apply it for general multiples of K. The bottleneck is to find spaces of cardinalities $|\mathcal{U}|, |\mathcal{X}|, |\mathcal{M}| \in O(2^{n/2})$.

Example: As example, we sketch the procedure for $\alpha = 2$. Here, we can use $\mathcal{X} = \sum_{i=0}^{n/2-1} \mathtt{x}^{2i}$. In terms of bit strings, this means that we consider all values whose odd-indexed bits are fixed to zero and the even-indexed bits can take any value:

$$\mathcal{X} \overset{\text{def}}{=} \{X \in \mathbb{F}_2^n : X_{2i+1} = 0, \text{ for all } i \in [0..n/2]\} .$$

Thus, the structure of the elements in bit-wise notation is $X = (0, X_{n-2}, 0, X_{n-4}, \ldots, 0, X_0)$, that is all odd-indexed bits are zero. Clearly, the size is $|\mathcal{X}| = 2^{n/2}$. Moreover, we employ $\mathcal{U} = 2 \cdot \mathcal{X}$, that is

$$\mathcal{U} = \{U \in \mathbb{F}_2^n : X_{2i} = 0, \text{ for all } i \in [0..n/2]\} .$$

Thus, the structure of elements is $U = (U_{n-1}, 0, U_{n-3}, 0, \ldots, U_1, 0)$, i.e., all even-indexed bits are zero.
The goal of **A** is to find a construction query index $j \in [q_c]$ and primitive query indices $i, k \in [q_p]$ such that $(\widehat{U}^j = M^j \oplus K = U^i) \wedge (\widehat{X}^j = M^j \oplus 2 \cdot K = X^k)$. For this purpose, it constructs messages from $\mathcal{M} = \mathcal{U} + \mathcal{X}$, which have the structure of

$$M = (M_{n-2}, M_{n-2}, M_{n-4}, M_{n-4}, \ldots, M_0, M_0) + \begin{cases} 0 & \text{if } K < 2^{n-1} \\ p(\mathtt{x}) & \text{otherwise.} \end{cases}$$

Let $K = (K_{n-1}, K_{n-2}, \ldots, K_0)$. **A** can proceed as follows:

1. Query all $2^{n/2}$ values $U^i \in \mathcal{U}$ to its primitive oracle π_1 and store $U^i \oplus V^i$ into a list $\mathcal{L}_U$.
2. Query all $2^{n/2}$ values $X^k \in \mathcal{X}$ to its primitive oracle π_2 and store $X^k \oplus Y^k$ into a list $\mathcal{L}_X$.
3. Query all $2 \cdot 2^{n/2}$ values $M^j \in \mathcal{M}$ to its construction oracle and store $M^j \oplus C^j$ into a list $\mathcal{L}_M$.
4. Let $c = 1$. Repeat the queries ${U'}^i$, ${X'}^k$, and ${M'}^j$, for all $i, j, k \in [2^{n/2}]$ and store the results ${U'}^i \oplus {V'}^i$ into a list $\mathcal{L}'_U$, ${X'}^k \oplus {Y'}^k$ into a list $\mathcal{L}'_X$, and ${M'}^j \oplus {C'}^j$ into a list $\mathcal{L}'_M$.
5. If there exist tuples (M^j, U^i, X^k) in the lists such that $C_1^j \oplus C_2^j = (U^i \oplus V^i) \oplus (X^k \oplus Y^k)$ and ${C'}_1^j \oplus {C'}_2^j = ({U'}^i \oplus {V'}^i) \oplus ({X'}^k \oplus {Y'}^k)$, output real, and random otherwise.

For the real construction, the success probability is one. We obtain two cases: (1) $K < 2^{n-1}$, and (2) $K \geq 2^{n-1}$. We consider the former case first. Here, the correct tuple (M^j, U^i, X^k) is given by

$$\begin{aligned}
M^j &= (K_{n-2}, K_{n-2}, \ldots, K_2, K_2, K_0, K_0), \text{ which implies} \\
\widehat{U}^j &= M^j \oplus K \\
&= (K_{n-2}, K_{n-2}, \ldots, K_2, K_2, K_0, K_0) \oplus (K_{n-1}, K_{n-2}, \ldots, K_3, K_2, K_1, K_0) \\
&= (K_{n-1} \oplus K_{n-2}, 0, \ldots, K_3 \oplus K_2, 0, K_1 \oplus K_0, 0), \\
\widehat{X}^j &= M^j \oplus \mathtt{2}K \\
&= (K_{n-2}, K_{n-2}, \ldots, K_2, K_2, K_0, K_0) \oplus (K_{n-2}, K_{n-3}, \ldots, K_2, K_1, K_0, 0) \\
&= (0, K_{n-2} \oplus K_{n-3}, \ldots, 0, K_2 \oplus K_1, 0, K_0)\,,
\end{aligned}$$

where $\widehat{U}^j$ follows from the fact that $K < 2^{n-1}$. Since $\widehat{U}^j \in \mathcal{U}$ and $\widehat{X}^j \in \mathcal{X}$, there exist i, k such that $U^i = \widehat{U}^j$ and $X^k = \widehat{X}^j$. The addition of c to all values in the tuple to create $({M'}^j, {U'}^i, {X'}^k)$ leaves the conditions fulfilled.
In the latter case, let the representation of $p(\mathtt{x})$ be $(p_{n-1}, \ldots, p_1, p_0)$. In this case, the correct tuple (M^j, U^i, X^k) is given by

$$\begin{aligned}
M^j &= (K_{n-2}, K_{n-2}, \ldots, K_2, K_2, K_0, K_0), \\
&\quad \oplus (p_{n-1}, p_{n-2}, \ldots, p_3, p_2, p_1, p_0), \text{ which implies} \\
\widehat{U}^j &= M^j \oplus K \\
&= (K_{n-2}, K_{n-2}, \ldots, K_2, K_2, K_0, K_0) \oplus (K_{n-1}, K_{n-2}, \ldots, K_3, K_2, K_1, K_0) \\
&\quad \oplus (p_{n-1}, p_{n-2}, \ldots, p_3, p_2, p_1, p_0) \\
&= (K_{n-1} \oplus K_{n-2}, 0, \ldots, K_3 \oplus K_2, 0, K_1 \oplus K_0, 0) \\
&\quad \oplus (p_{n-1}, p_{n-2}, \ldots, p_3, p_2, p_1, p_0), \\
\widehat{X}^j &= M^j \oplus \mathtt{2}K \\
&= (K_{n-2}, K_{n-2}, \ldots, K_2, K_2, K_0, K_0) \oplus (K_{n-2}, K_{n-3}, \ldots, K_2, K_1, K_0, 0) \\
&\quad \oplus (p_{n-1}, p_{n-2}, \ldots, p_3, p_2, p_1, p_0) \\
&= (0, K_{n-2} \oplus K_{n-3}, \ldots, 0, K_2 \oplus K_1, 0, K_0) \oplus (p_{n-1}, p_{n-2}, \ldots, p_3, p_2, p_1, p_0),
\end{aligned}$$

where $\widehat{U}^j$ follows from the fact that $K \geq 2^{n-1}$. Again $\widehat{U}^j \in \mathcal{U}$ and $\widehat{X}^j \in \mathcal{X}$, which means that there exists a tuple and the distinguisher succeeds.

5 Security Analysis of CENCPP

This section studies the nE security of CENCPP. Prior, we briefly revisit the analysis of CENC.

5.1 Recalling the Security of CENC

The Security of XORP. In [Iwa06], Iwata showed that CENC[w] is secure for up to $2^{2n/3}/w$ message blocks as long as E_K is a secure block cipher. At Dagstuhl'07 [Iwa07], he added an attack that needed $2^n/w$ queries, and showed $O(2^n/w)$ security if the total number of primitive calls remained below $\sigma < 2^{n/2}$. He conjectured that CENC may be secure for up to $2^n/w$ blocks. In [IMV16], Iwata et al. confirmed that conjecture by a simple corollary from Patarin. We briefly recall their conclusion. In [Pat10, Theorem 6], Patarin showed the indistinguishability for the sum of multiple independent secret permutations. [IMV16] provided an explanation how this bound could be adapted to address the security of XORP:

$$\mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{XORP}[E_K,w]} \leq \frac{w^2 q}{2^n} + \mathbf{Adv}^{\mathsf{PRP}}_{E_K}((w+1)q, t). \tag{1}$$

Theorem 3 in [IMV16] conjectured for m being a multiple of w:

$$\mathbf{Adv}^{\mathsf{nE}}_{\mathsf{CENC}[E_K,w]}(q, m, t) \leq \frac{mwq}{2^n} + \mathbf{Adv}^{\mathsf{PRP}}_{E_K}\left(\frac{w+1}{w} mq, t\right).$$

Thus, CENC provided a convenient trade-off of $w+1$ calls per w message blocks while ensuring security for up to $2^n/w$ calls to E_K.

5.2 The Security of CENCPP

In the following, let n, w be positive integers, $\pi_0, \ldots, \pi_w \twoheadleftarrow \mathsf{Perm}(\mathbb{F}_2^n)$ be independent public permutations and $K_0, K_1 \twoheadleftarrow \mathcal{K}$ be independent secret keys. We write $\mathbf{K} = (K_0, K_1)$ and $\boldsymbol{\pi} = (\pi_0, \ldots, \pi_w)$ for brevity. Again, we conduct a two-step analysis, where we consider (1) the PRF security of XORPP[$\boldsymbol{\pi}, w$] and (2) the PRF security of CENCPP[$\boldsymbol{\pi}, w$].

Theorem 1. It holds that

$$\mathbf{Adv}^{\mathsf{nE}}_{\mathsf{CENCPP}[\boldsymbol{\pi},w]_{\mathbf{K}}}(q_p, q_c, \sigma) \leq \mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{XORPP}[\boldsymbol{\pi},w]_{\mathbf{K}}}\left(q_p, \frac{m}{w} q_c, \sigma\right).$$

Proof. The proof follows a similar argumentation as that of CENC in [IMV16]. For a maximal number of message chunks $\ell = \lceil \sigma/w \rceil$, CENCPP[$\boldsymbol{\pi}, w]_{\mathbf{K}}$ consists of the application of ℓ instances of $\mathsf{XORPP}[\boldsymbol{\pi}, w]_{\mathbf{K}}$. We can replace $\mathsf{XORPP}[\boldsymbol{\pi}, w]_{\mathbf{K}}$ by a random function ρ at the cost of

$$\mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{XORPP}[\boldsymbol{\pi},w]_{\mathbf{K}}}\left(q_p, \frac{m}{w}q,\sigma\right).$$

Since the resulting construction is indistinguishable from random bits, our claim in Theorem 1 follows.

Theorem 2. Let $q_c + (w+1)q_p \leq 2^{n-w}$ and $q_c \geq 9n$. It holds that

$$\mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{XORPP}[\boldsymbol{\pi},w]_{\mathbf{K}}}(q_p, q_c, \sigma) \leq \frac{(w+1)^2 q_p^2 q_c}{2^{2n}} + \frac{(w+1)^3 q_p q_c}{2^{2n}} + \frac{(w+1)^3 q_p^2 q_c^2}{2^{3n}} + \frac{2q_c(q_p+q_c)^{w+1}}{2^{n(w+1)}} + \frac{(w+1)^2}{2^n} + \frac{(w+1)^2 q_p \sqrt{3nq_c}}{2^n}.$$

Proof of Theorem 2. The analysis basically extends and adapts that by Chen et al. from two to more permutations. In the real world, $\mathbf{A}$ has access to a construction oracle that it can ask at most q_c tuples of nonces and messages to and will receive the corresponding ciphertexts from. Moreover, $\mathbf{A}$ has access to primitive oracles $\mathcal{O}_0, \ldots, \mathcal{O}_w$ that it can ask queries U_j^i or V_j^i to and obtains $V_j^i \leftarrow \pi_j(U_j^i)$ or $U_j^i \leftarrow \pi_j^{-1}(V_j^i)$ for $i \in [q_p]$ and $j \in [0..w]$, respectively. We say that it asks at most q_p queries to each of them. In the ideal world, the construction queries are answered by random bits of the expected length; the primitive oracles are the same in both worlds.

We partition the transcript τ into parts: $\tau = \tau_c \cup \tau_0 \cup \ldots \cup \tau_w$, where each partial transcript captures the queries and responses from a particular oracle. The construction transcript consists of the keys and the queries to and responses from the construction oracle: $\tau_c = \{(K_0, K_1), (M^1, C^1), \ldots, (M^{q_c}, C^{q_c})\}$. The primitive transcripts $\tau_j = \{(U_j^1, V_j^1), \ldots, (U_j^{q_p}, V_j^{q_p})\}$ contain exactly the queries to and responses from permutation π_j. We assume that the transcript does not contain duplicate elements. The keys K_0, K_1 are given to the distinguisher after its interaction but before it outputs its decision bit. In both worlds, those keys are sampled uniformly at random. With their help, the adversary can itself compute the inputs $\widehat{U}_i^j$ and outputs $\widehat{V}_i^j$ of permutations $i \in [0..w]$ and queries $j \in [q_c]$. We partition the set of all attainable transcripts into two disjoint sets of GoodT and BadT that represent good and bad transcripts. We say that $\tau \in$ BadT iff any of the following bad events holds; otherwise, τ is called a good transcript. Prior, we define sets $\mathcal{S}_{\alpha,\beta} =^{\mathrm{def}} \{(i,j,k) : \widehat{V}_\alpha^i \oplus \widehat{V}_\beta^i = V_\alpha^j \oplus V_\beta^k\}$ for $i \in [q_c]$, $j, k \in [q_p]$, and distinct $\alpha < \beta \in [0..w]$. Let $\theta = q_p^2 q_c / 2^n + q_p\sqrt{3nq_c}$ be the threshold from Lemma 2.

Bad Events. We extend the three bad events from [CLM19] as follows. To aid the reader, we recall that U_j^i is the input of the i-th primitive query to the primitive π_j that is answered by V_j^i and vice versa. $\widehat{U}_j^i$ the input of the i-th query to that would go to π_j in the real construction and produce $\widehat{V}_j^i$.

- bad_1: There exists a construction query index $j \in [q_c]$ and primitive query indices $i, k \in [q_p]$ as well as distinct permutation indices $\alpha, \beta \in [0..w]$ such that $(\widehat{U}^j_\alpha = U^i_\alpha) \wedge (\widehat{U}^j_\beta = U^k_\beta)$.
- bad_2: There exist distinct $\alpha, \beta \in [0..w]$ such that $|\mathcal{S}_{\alpha,\beta}| \geq \theta$.
- bad_3: There exists a construction query index $j \in [q_c]$ and primitive query indices $i, k \in [q_p]$ as well as distinct permutation indices $\alpha, \beta \in [0..w]$ such that $(\widehat{U}^j_\alpha = U^i_\alpha) \wedge (\widehat{V}^j_\beta = V^k_\beta)$.
- bad_4: There exists a construction query index $j \in [q_c]$ and a primitive query index $i \in [q_p]$ as well as permutation indices $\alpha, \beta, \gamma \in [0..w]$ with $\beta \neq \gamma$ such that $(\widehat{U}^j_\alpha = U^i_\alpha) \wedge (\widehat{V}^j_\beta = \widehat{V}^j_\gamma)$.
- bad_5: There exist distinct construction query indices $j, k \in [q_c]$, primitive query index $i \in [q_p]$ as well as permutation indices $\alpha, \beta, \gamma \in [0..w]$ such that $(\widehat{U}^j_\alpha = U^i_\alpha) \wedge (\widehat{U}^k_\gamma = U^k_\gamma) \wedge (\widehat{V}^j_\beta = \widehat{V}^k_\beta)$.

The probability that a transcript in the ideal world is bad is at most

$$\Pr\left[\Theta_{\text{ideal}} \in \textsc{BadT}\right] \leq \sum_{i=1}^{2} \Pr[\mathsf{bad}_i] + \Pr[\mathsf{bad}_3 | \neg \mathsf{bad}_2] + \sum_{i=4}^{5} \Pr[\mathsf{bad}_i]\,.$$

Lemma 3. Let $q_c + (w+1)q_p \leq 2^{n-w}$. It holds that

$$\Pr\left[\Theta_{\text{ideal}} \in \textsc{BadT}\right] \leq \frac{(w+1)^2 q_p^2 q_c}{2^{2n}} + \frac{(w+1)^3 q_p q_c}{2^{2n}} + \frac{(w+1)^2}{2^n} + \frac{(w+1)^3 q_p^2 \binom{q_c}{2}}{2^{3n}} + \frac{(w+1)^2 q_p \sqrt{3nq_c}}{2^n}\,.$$

Proof. In the following, we study the probabilities of the individual bad events.

bad_1. This event considers the collisions between two construction-query inputs and two primitive-query inputs. For this event, it must hold that

$$M^j \oplus (2^{\alpha-1} K_0 \oplus 2^{2(\alpha-1)} K_1) = U^i_\alpha$$
$$M^j \oplus (2^{\beta-1} K_0 \oplus 2^{2(\beta-1)} K_1) = U^k_\beta.$$

Thus, the difference is always given by $2^{\beta-\alpha} K_0 \oplus 2^{2(\beta-\alpha)} K_1 \neq 0$. Moreover, the differences between each tuple $(\alpha, \beta) \neq (\alpha', \beta')$ are pairwise independent. Thus, the equations are always independent since the keys are sampled independently at random. Thus, the probability is 2^{-2n}. Over all indices, we obtain

$$\Pr[\mathsf{bad}_1] = \sum_{j \in [q_c]} \sum_{i \in [q_p]} \sum_{k \in [q_p]} \sum_{0 \leq \alpha < \beta \leq w} \Pr\left[\widehat{U}^j_\alpha = U^i_\alpha \wedge \widehat{U}^j_\beta = U^k_\beta\right] \leq \frac{\binom{w+1}{2} q_p^2 q_c}{2^{2n}}\,.$$

bad_2. For fixed α, β, the probability of this event is given by Lemma 2. Over the union bound of all combinations of α and β, we obtain that

$$\sum_{0 \leq \alpha < \beta \leq w} \Pr\left[|\mathcal{S}_{\alpha,\beta}| \geq \theta\right] \leq \frac{2\binom{w+1}{2}}{2^n}\,.$$

bad_3. This event is similar to bad_1; it considers collisions between a construction-query input and a primitive-query input as well as between a construction-query and primitive-query output. For this event, it must hold that

$$M^j \oplus (2^{\alpha-1}K_0 \oplus 2^{2(\alpha-1)}K_1) = U^i_\alpha$$
$$\widehat{X}^j_\beta \oplus (2^{\beta-1}K_0 \oplus 2^{2(\beta-1)}K_1) = V^k_\beta\,.$$

The first equation reveals $\widehat{V}^j_\alpha = V^i_\alpha$, which yields $\widehat{X}^j_\alpha$ and therefore $\widehat{X}^j_0 = C^j_\alpha \oplus \widehat{X}^j_\alpha$. Thus, the adversary can deduce $\widehat{X}^j_\beta$ for all $\beta \neq \alpha$. Since all values C^j_β are sampled uniformly and independently at random, and so are the keys in both equations, and the linear equation system between each two equations has maximal rank, the probability is 2^{-2n}. Since bad_2 does not hold, there are at most θ such tuples. Over all indices, we obtain therefore

$$\Pr[\mathsf{bad}_3 | \neg\mathsf{bad}_2] = \sum_{j\in[q_c]} \sum_{i\in[q_p]} \sum_{k\in[q_p]} \sum_{0\le\alpha<\beta\le w} \Pr\left[\widehat{U}^j_\alpha = U^i_\alpha \wedge \widehat{V}^j_\beta = V^k_\beta\right]$$
$$\le \frac{\binom{w+1}{2} q_p^2 q_c}{2^{2n}} + \frac{\binom{w+1}{2} q_p \sqrt{3nq_c}}{2^n}\,.$$

bad_4. In this event, a construction-query input collides with a primitive-query input, which allows to derive a candidate of $\widehat{V}^j_\alpha$, which reveals all further permutation outputs for the j-th construction query. Thereupon, one of them collides with a further construction-query output. The probability for the above is at most 2^{-2n} since $\{\alpha, \beta, \gamma\}$ contain at least two independent indices. W.l.o.g., assume $\beta \neq \alpha$. Then, C^j_α and C^j_β are chosen independently uniformly at random from $\mathbb{F}_2^n$. Since there are at most w choices for $\gamma \neq \beta$:

$$\Pr[\mathsf{bad}_4] = \sum_{j\in[q_c]} \sum_{i\in[q_p]} \sum_{\alpha\in[0..w]} \sum_{0\le\beta<\gamma\le w} \Pr\left[\widehat{U}^j_\alpha = U^i_\alpha \wedge \widehat{V}^j_\beta = \widehat{V}^j_\gamma\right]$$
$$\le \frac{(w+1)\binom{w+1}{2} q_p q_c}{2^{2n}}\,.$$

bad_5. In this event, the permutation inputs of two distinct construction queries collide with a primitive-query input each. Both input collisions allow to derive a candidate of $\widehat{V}^j_\alpha$ and $\widehat{V}^k_\gamma$, which reveals all further permutation outputs for both construction queries. Thereupon, one of the outputs collides between the construction-query outputs. The probability for the collisions with the primitive-query inputs is 2^{-2n} since C^j_α and C^k_γ are chosen independently uniformly at random from $\mathbb{F}_2^n$. The probability of the third collision between $\widehat{V}^j_\beta = \widehat{V}^k_\beta$ is again 2^{-n}. We obtain that

$$\Pr[\mathsf{bad}_5] \le \sum_{1\le j<k\le q_c} \sum_{i\in[q_p]} \sum_{\ell\in[q_p]} \sum_{\alpha,\beta,\gamma\in[0..w]} \Pr\left[\widehat{U}^j_\alpha = U^i_\alpha \wedge \widehat{V}^j_\beta = \widehat{V}^k_\beta \wedge \widehat{U}^k_\gamma = U^\ell_\gamma\right]$$
$$\le \frac{(w+1)^3 q_p^2 \binom{q_c}{2}}{2^{3n}}\,.$$

The bound in Lemma 3 follows. □

Good Transcripts. It remains to consider the interpolation probabilities of good attainable transcripts.

Lemma 4. It holds that

$$\frac{\Pr[\Theta_{\text{real}} = \tau]}{\Pr[\Theta_{\text{ideal}} = \tau]} \geq 1 - \frac{2q_c(q_p + q_c)^{w+1}}{2^{n(w+1)}}.$$

Proof. Let $\mathsf{All}_{\text{real}}(\tau)$ denote the set of all oracles in the real world, and $\mathsf{All}_{\text{ideal}}(\tau)$ the set of all oracles in the ideal world that produce $\tau \in \text{GOODT}$. Let $\mathsf{Comp}_{\text{real}}(\tau)$ denote the fraction of oracles in the real world that are compatible with τ and $\mathsf{Comp}_{\text{ideal}}(\tau)$ the corresponding fraction in the ideal world. It holds that

$$\frac{\Pr\left[\Theta_{\text{real}} = \tau\right]}{\Pr\left[\Theta_{\text{ideal}} = \tau\right]} = \frac{|\mathsf{Comp}_{\text{real}}(\tau)| \cdot |\mathsf{All}_{\text{ideal}}(\tau)|}{|\mathsf{Comp}_{\text{ideal}}(\tau)| \cdot |\mathsf{All}_{\text{real}}(\tau)|}.$$

We can easily bound the number for three out of four terms:

$$|\mathsf{All}_{\text{real}}(\tau)| = (2^n)^2 \cdot (2^n!)^{w+1}$$

since there exist $(2^n)^2$ keys and $2^n!$ possible ways for each of the $w+1$ independent permutations π_ι. The same argument holds in the ideal world

$$|\mathsf{All}_{\text{ideal}}(\tau)| = (2^n)^2 \cdot (2^n!)^{w+1} \cdot (2^{wn})^{2^n},$$

combined with $(2^{wn})^{2^n}$ random functions for the answers to the construction queries. Moreover,

$$|\mathsf{Comp}_{\text{ideal}}(\tau)| = (2^{wn})^{2^n - q_c} \cdot \prod_{i=0}^{w} (2^n - q_p)!$$

compatible oracles exist in the ideal world, where $(2^{wn})^{2^n - q_c}$ are the oracles that produce the correct construction-query outputs for the $2^n - q_c$ remaining non-queried inputs, and for all permutations, there exist $(2^n - q_p)!$ compatible primitives each.

It remains to determine $|\mathsf{Comp}_{\text{real}}(\tau)|$. Like Chen et al., we regroup the queries from the transcript parts. We generalize their claim [CLM19] to the following to cover all $w+1$ permutations:

Claim. For a good transcript, $\tau \in \text{GOODT}$, any construction query $(M^j, C^j) \in \tau_c$ collides with at most one primitive query (U^i_α, V^i_α) for some $\alpha \in [0..w]$, but never with multiple primitive queries.

We regroup the queries from τ_c, τ_0, …, τ_w to τ_c^{new}, τ_0^{new}, …, τ_w^{new}. The new transcript sets are initialized by their corresponding old parts, and reordered as follows:

If there exist $j \in [q_c]$, $i \in [q_p]$, and $\alpha \in [0..w]$ such that $\widehat{U}^j_\alpha = U^i_\alpha$, then (M^j, C^j_α) is removed from τ_c^{new} and $(U_\beta, V_\beta) = (\widehat{U}^j_\beta, \widehat{V}^j_\beta)$ is added to $\tau_\beta^{\mathsf{new}}$, for all $\beta \in [0..w]$ with $\beta \neq \alpha$.

Given q_c constructions queries and q_p queries to each of the permutations in the original transcript, the numbers of queries moved from τ_c into the primitive partial transcripts τ_i is denoted by s_i. The number of queries in the new construction transcript is denoted by $q' = q_c - \sum_{i=0}^{w} s_i$. In the following, for a given transcript τ_0^{new} of q' elements, it remains to count the number of permutations $(\boldsymbol{\pi})$ that are compatible with the transcript. The set of occurred (i.e., prohibited) outputs of π_α are denoted by V_α^{out}. For $j = [0..q'-1]$, let

$$\lambda_{j+1} \overset{\text{def}}{=} \left| \left\{ (V_0^1, \ldots, V_0^{j+1}, \ldots, V_w^1, \ldots, V_w^{j+1}) \right\} \right| \tag{2}$$

be the number of solutions that satisfy

(1) $\left\{ (V_0^1, \ldots, V_0^j, \ldots, V_w^1, \ldots, V_w^j) \right\}$ satisfy the conditions recursively,
(2) It holds that

$$\begin{aligned} V_0^{j+1} \oplus V_1^{j+1} &= C_1^{j+1} \oplus (K_0 \oplus K_1) \oplus (2K_0 \oplus 2^2 K_1) \\ &\vdots \\ V_0^{j+1} \oplus V_w^{j+1} &= C_w^{\alpha+1} \oplus (K_0 \oplus K_1) \oplus (2^w K_0 \oplus 2^{2w} K_1) \end{aligned} \tag{3}$$

(3.0) It holds that $V_0^{\alpha+1} \notin \{V_0^1, \ldots, V_0^\alpha\} \cup V_0^{\mathsf{out}}$.
...
(3.w) It holds that $V_w^{\alpha+1} \notin \{V_w^1, \ldots, V_w^\alpha\} \cup V_w^{\mathsf{out}}$.

Then, the goal is to define a recursive expression for $\lambda_{\alpha+1}$ from λ_α such that a lower bound can be found for the expression $\lambda_{\alpha+1}/\lambda_\alpha$. It holds that

$$|\mathsf{Comp}_{\mathrm{real}}(\tau)| = \lambda_{q'} \cdot (2^n - (q_p + s_0 + q'))! \cdot \cdots \cdot (2^n - (q_p + s_w + q'))! \cdot (2^n)^{w \cdot q_c},$$

where the second term represents the number of permutations compatible with π_0 and the rightmost term contains the number of permutations compatible with π_w. We obtain

$$\frac{\Pr[\Theta_{\mathrm{real}} = \tau]}{\Pr[\Theta_{\mathrm{ideal}} = \tau]} = \frac{\lambda_{q'} \cdot \prod_{i=0}^{w} (2^n - (q_p + s_i + q'))!}{((2^n - q_p)!)^{w+1}}. \tag{4}$$

Let $\mathcal{B}_{(1,2)}$ denote the set of solutions that comply with only Conditions (1) and (2), without considering Conditions (3.0) through (3.w). Moreover, let $\mathcal{B}_{(3.\iota:i)}$ denote the set of solutions compatible with Conditions (1) and (2), but not with (3.$\iota : i$), for $i = 1, \ldots, \alpha + |V_\iota^{\mathsf{out}}|$. From inclusion-exclusion, it follows that

$$\begin{aligned} \lambda_{\alpha+1} &= \left|\mathcal{B}_{(1,2)}\right| - \left| \bigcup_{i=1}^{\alpha+|V_0^{\mathsf{out}}|} \mathcal{B}_{(3.0:i)} \right| \cup \cdots \cup \left| \bigcup_{i=1}^{\alpha+|V_0^{\mathsf{out}}|} \left|\mathcal{B}_{(3.w:i)}\right| \right| \\ &\geq \left|\mathcal{B}_{(1,2)}\right| - \left| \sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|} \left|\mathcal{B}_{(3.0:i)}\right| \right| - \cdots - \left| \sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|} \left|\mathcal{B}_{(3.w:i)}\right| \right| \end{aligned}$$

$$
\begin{aligned}
&+ \sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|} \sum_{i'=1}^{\alpha+|V_1^{\mathsf{out}}|} \left|\mathcal{B}_{(3.0:i)} \cap \mathcal{B}_{(3.1:i')}\right| + \cdots \\
&+ \sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|} \sum_{i'=1}^{\alpha+|V_1^{\mathsf{out}}|} \left|\mathcal{B}_{(3.(w-1):i)} \cap \mathcal{B}_{(3.w:i')}\right| \\
\geq{}& 2^n \cdot \lambda_\alpha - \sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|} \lambda_\alpha - \cdots - \sum_{i=1}^{\alpha+|V_w^{\mathsf{out}}|} \lambda_\alpha .
\end{aligned}
$$

So, it follows that

$$
\lambda_{\alpha+1} \geq 2^n \cdot \lambda_\alpha - (\alpha + q_p + s_0) \cdot \lambda_\alpha - \ldots - (\alpha + q_p + s_w) \cdot \lambda_\alpha .
$$

Therefore,

$$
\frac{\lambda_{\alpha+1}}{\lambda_\alpha} \geq 2^n - (w+1)\alpha - (w+1)q_p - \sum_{i=0}^{w} s_i
$$

with $\lambda_0 = 1$. It follows from Equation (4) that

$$
\begin{aligned}
(4) ={}& \prod_{j=0}^{s_0-1} \frac{2^n}{2^n - q_p - j} \cdot \ldots \cdot \prod_{j=0}^{s_w-1} \frac{2^n}{2^n - q_p - j} \cdot \prod_{i=0}^{q'-1} \frac{\lambda_{\alpha+1}}{\lambda_\alpha} \cdot \frac{(2^n)^w}{\prod_{j=0}^{w}(2^n - q_p - i - s_j)} \\
\geq{}& \prod_{i=0}^{q'-1} \frac{(2^n - (w+1)\alpha - (w+1)q_p - \sum_{j=0}^{w} s_j)}{\prod_{j=0}^{w}(2^n - q_p - i - s_j)} \cdot 2^{nw} \\
\geq{}& \prod_{i=0}^{q'-1} \left(1 - \frac{\prod_{j=0}^{w}(q_p + i + s_j)}{\prod_{j=0}^{w}(2^n - q_p - i - s_j)}\right) \\
\geq{}& \prod_{i=0}^{q'-1} \left(1 - \frac{\prod_{j=0}^{w}(q_p + q' + s_j)}{\prod_{j=0}^{w}(2^n - q_p - q' - s_j)}\right) \\
\geq{}& \prod_{i=0}^{q'-1} \left(1 - \frac{\prod_{j=0}^{w}(q_p + q' + s_j)}{(2^n - q_p - q' - s_j)^{w+1}}\right) \geq \left(1 - \frac{(q_p + q)^{w+1}}{(2^n - q_p - q' - s_j)^{w+1}}\right)^{q'} \\
\geq{}& 1 - \frac{2q'(q_p + q)^{w+1}}{2^{n(w+1)}} \geq 1 - \frac{2q_c(q_p + q_c)^{w+1}}{2^{n(w+1)}},
\end{aligned}
$$

using the fact that $q_p + q' + s_j \ll 2^{n-w}$. The bound in Lemma 4 follows. □

Our claim in Theorem 2 follows from Lemma 1, 3, and 4. □

6 Domain-separated Variants

This section derives a single-primitive variant of SoEM that uses domain separation for distinct permutations.

Algorithm 2 Definition of DS-CENCPP, DS-XORPP, and DS-SoEM.

```
**function** $\text{DS-CENCPP}[\pi, w].\mathcal{E}_{\mathbf{K}}(N, M)$
  $(M_1, \ldots, M_m) \xleftarrow{n} M$
  $\ell \leftarrow \lceil m/w \rceil$
  **for** $i \leftarrow 0..\ell - 1$ **do**
    $j \leftarrow i \cdot w$
    $(S_{j+1} \parallel \cdots \parallel S_{j+w})$
        $\leftarrow \mathsf{DS\text{-}XORPP}[\pi, w]_{\mathbf{K}}(N \parallel \langle i \rangle_{\mu})$
    **for** $k \leftarrow j + 1..j + w$ **do**
      $C_k \leftarrow S_k \oplus M_k$
  **return** $\mathsf{msb}_{|M|}(C_1 \parallel \cdots \parallel C_m)$
```

```
**function** $\text{DS-CENCPP}[\pi, w].\mathcal{D}_{\mathbf{K}}(N, C)$
  **return** $\mathsf{DS\text{-}CENCPP}[\pi, w].\mathcal{E}_{\mathbf{K}}(N, C)$
```

```
**function** $\text{DS-XORPP}[\pi, w]_{\mathbf{K}}(M)$
  $(K_0, K_1) \leftarrow \mathbf{K}$
  $U_0 \leftarrow (M \oplus \mathsf{msb}_{n-d}(K_0 \oplus K_1)) \parallel \langle 0 \rangle_d$
  $X_0 \leftarrow \pi(U_0) \oplus (K_0 \oplus K_1)$
  **for** $j \leftarrow 1..w$ **do**
    $L_j \leftarrow (\mathsf{2}^j \cdot K_0) \oplus (\mathsf{2}^{2j} \cdot K_1)$
    $U_j \leftarrow (M \oplus \mathsf{msb}_{n-d}(L_j)) \parallel \langle j \rangle_d$
    $X_j \leftarrow \pi(U_j) \oplus L_j$
    $C_j \leftarrow X_j \oplus X_0$
  **return** $(C_1 \parallel \cdots \parallel C_w)$
```

```
**function** $\text{DS-SoEM}[\pi, w]_{\mathbf{K}}(M)$
  $(K_0, K_1) \leftarrow \mathbf{K}$
  $U \leftarrow (\mathsf{msb}_{n-d}(K_0) \oplus M) \parallel \langle 0 \rangle_d$
  $X \leftarrow (\mathsf{msb}_{n-d}(K_1) \oplus M) \parallel \langle 1 \rangle_d$
  $V \leftarrow \pi(U) \oplus K_0$
  $Y \leftarrow \pi(X) \oplus K_1$
  $C \leftarrow V \oplus Y$
  **return** $C$
```

DS-SoEM. First, we propose DS-SoEM, a sum of Even-Mansour constructions that uses $(n-d)$-bit message inputs and fixes d bits to encode domains that are distinct for each permutation. Let $\pi \in \mathsf{Perm}(\mathbb{F}_2^n)$ and $\mathbf{K} = (K_0, K_1) \in (\mathbb{F}_{2^n})^2$. We define $\mathsf{DS\text{-}SoEM}[\pi]_{K_0,K_1} : (\mathbb{F}_{2^n})^2 \times \mathbb{F}_2^{n-d} \to \mathbb{F}_2^n$ to compute $\mathsf{DS\text{-}SoEM}[\pi]_{K_0,K_1}(M)$, as listed in Algorithm 2. Note that we use $(n-d)$ bits of the key in forward direction only, i.e., the domain is **not** masked. Furthermore, for DS-SoEM, a single bit (i.e. $d = 1$) suffices to set a zero bit for the call to the left and a one bit for the domain input to the right permutation. An illustration is given in Figure 4a.

DS-XORPP. In a similar manner, we can define $\mathsf{DS\text{-}XORPP}[\pi, w]$. Here, $d \geq \lceil \log_2(w+1) \rceil$ bits are necessary to separate the domains. Let again $\mathbf{K} =^{\text{def}} (K_0, K_1) \in (\mathbb{F}_{2^n})^2$. We define

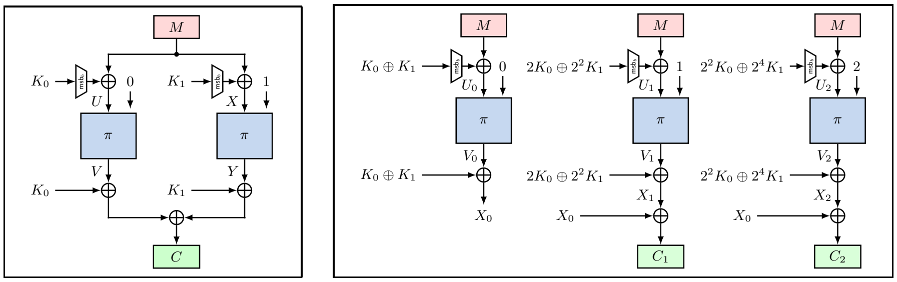
$$\mathsf{DS\text{-}XORPP}[\pi, w] : (\mathbb{F}_{2^n})^2 \times \mathbb{F}_2^{n-d} \to (\mathbb{F}_2^n)^w$$

as given in Algorithm 2. An illustration is given in Figure 4b. The input domain is $M \in \mathbb{F}_2^{n-d}$. Again, we use $(n-d)$ bits of the key in forward direction only, i.e., the domain is **not** masked.

DS-CENCPP is then defined in the natural manner. Let $\mathcal{N} =^{\text{def}} \mathbb{F}_2^{\nu+\mu}$ be a nonce space such that $\nu + \mu = n - d$. Let $N \in \mathcal{N}$ be a nonce and $M \in \mathbb{F}_2^*$ be a message. Let again $\mathbf{K} =^{\text{def}} (K_0, K_1) \in (\mathbb{F}_{2^n})^2$ and $\pi \in \mathsf{Perm}(\mathbb{F}_2^n)$. Then, the encryption and decryption algorithms $\mathcal{E}$ and $\mathcal{D}$ of $\mathsf{DS\text{-}CENCPP}[\pi, w]_{\mathbf{K}}(N, M)$ are provided in Algorithm 2.

7 Distinguishers on DS-SoEM and DS-XORPP

This section provides a distinguisher on DS-SoEM that matches our later security bound, and distinguishers on variants that mask also the domain and use only a single key, respectively. Thus, they show that our bound is tight (up to a logarithmic factor) and provide a design rationale on our constructions.



(a) $\mathsf{DS\text{-}SoEM}[\pi]_{K_0,K_1}$. **(b)** $\mathsf{DS\text{-}XORPP}[\pi, w]_{K_0,K_1}$.

Fig. 4: The domain-separated constructions, here with $\mathsf{DS\text{-}XORPP}[\pi, 2]$. The trapezoids represent truncation of the key masks at the input to their $b = n - d$ most significant bits.

The Existing Distinguisher from [CLM19, Proposition 2] on SoEM12 (one permutation, two independent keys) needed $3 \cdot 2^{n/2}$ queries:

1. For $i \leftarrow 1..2^{n/2}$, query $M^i = (\langle i \rangle_{n/2} \,\|\, 0^{n/2})$ to obtain C^i, and query $M^{*i} = M^i \oplus 1$ to obtain C^{*i}.
2. For $j \leftarrow 1..2^{n/2}$, query $M'^j = (0^{n/2} \,\|\, \langle j \rangle_{n/2})$ to obtain C'^j, and query $M'^{*j} = M'^j \oplus 1$ to obtain C'^{*j}.

After $3 \cdot 2^{n/2}$ queries, there exists one tuple $(M^i, M^{*i}, M'^j, M'^{*j})$ such that $M^i \oplus M'^j = M^{*i} \oplus M'^{*j} = K_0 \oplus K_1$, which can be seen if $C^i = C'^j$ and $C^{*i} = C'^{*j}$. Note that the fourth set of queries M'^{*j} is not new, but can be taken from the other sets. For SoEM, the distinguisher exploited that one can find two queries M and M' such that their inputs to the left and right permutation are **swapped**. For DS-SoEM, this distinguisher does **not** apply since the domain separation prevents that the permutation inputs can be swapped entirely.

A Working Distinguisher can be constructed with significant advantage and $6c \cdot 2^{2n/3}$ queries, for small constant $c \geq 1$. Let $q = c \cdot 2^{2n/3}$.

1. For $j \leftarrow 1..q$, query a random M^j (without replacement), obtain C^j. Moreover, query $M^{*j} = M^j \oplus \langle 1 \rangle_n$ to obtain C^{*j} and store (C^j, C^{*j}).
2. For $i \leftarrow 1..q$, sample $u^i \in \mathbb{F}_2^{n-d}$ (without replacement), query $U^i = (u^i \,\|\, \langle 0 \rangle_d)$ to π, and obtain V^i. Derive $U^{*i} = U^i \oplus 10^{n-1}$, query U^{*i} to π to obtain V^{*i} and store (V^i, V^{*i}).
3. For $k \leftarrow 1..q$, sample $x^k \in \mathbb{F}_2^{n-d}$ (without replacement), query $X^k = (x^k \,\|\, \langle 1 \rangle_d)$ to π, and obtain Y^k. Derive $X^{*k} = X^k \oplus 10^{n-1}$, query X^{*k} to π to obtain Y^{*k} and store (Y^k, Y^{*k}).

With high probability, there exists a tuple (M^j, U^i, X^k) such that

$$((M^j \oplus \mathsf{msb}_{n-d}(K_0)) \,\|\, \langle 0 \rangle_d) = U^i \quad \text{and} \quad ((M^j \oplus \mathsf{msb}_{n-d}(K_1)) \,\|\, \langle 1 \rangle_d) = X^k \,.$$

If this is the case, check if

$$((M^{*j} \oplus \mathsf{msb}_{n-d}(K_0) \parallel \langle 0 \rangle_d) = U^{*i} \quad \text{and} \quad ((M^{*j} \oplus \mathsf{msb}_{n-d}(K_1)) \parallel \langle 1 \rangle_d) = X^{*k}$$

also holds. If yes, return real; return random otherwise.

Why Not Also Mask The Domain? If the keys K_0 and K_1 would be XORed also to the domains, it could hold for DS-SoEM that

$$\mathsf{lsb}_d(K_0) \oplus \langle 0 \rangle_d = \mathsf{lsb}_d(K_1) \oplus \langle 1 \rangle_d \,.$$

Similarly, it could hold for DS-XORPP for any distinct pair $i, j \in [0..w]$ that

$$\mathsf{lsb}_d(2^i K_0 \oplus 2^{2i} K_1) \oplus \langle i \rangle_d = \mathsf{lsb}_d(2^j K_0 \oplus 2^{2j} K_1)) \oplus \langle j \rangle_d$$

This would invalidate the effect of the domain. Note that, while the distinguisher from [CLM19, Proposition 2] would still not be applicable, a slide attack (cf. [DKS12,DDKS13]) may become.
In the following, we consider a variant of DS-SoEM$[\pi]$ where the inputs to the permutations would be defined as

$$U^i \leftarrow (M^i \parallel \langle 0 \rangle_d) \oplus K_0 \quad \text{and} \quad X^i \leftarrow (M^i \parallel \langle 1 \rangle_d) \oplus K_1.$$

We assume that $K_0, K_1 \twoheadleftarrow \mathbb{F}_2^n$, $d = 1$ for simplicity, and that $\mathsf{lsb}_d(K_0) \oplus \mathsf{lsb}_d(K_1) = 1$, i.e., their least significant d bits differ, which holds with probability 0.5. Let $c \in \mathbb{F}_2^{n-d}$ be an arbitrary non-zero constant. Then:

1. For $i \leftarrow 1..2^{n/2}$, sample $M^i = (\langle i \rangle_{n/2} \parallel 0^{n/2-d})$, obtain C^i and store it.
2. Derive $M^{*i} = M^i \oplus c$, and obtain its corresponding ciphertext C^{*i}.
3. Similarly, for $j \leftarrow 1..2^{n/2-d}$, sample $M^j = (0^{n/2} \parallel \langle j \rangle_{n/2-d})$, obtain C^j and store it.
4. Derive $M^{*j} = M^j \oplus c$, and obtain its corresponding ciphertext C^{*j}.
5. If there exist distinct $i \neq j$ such that $C^i = C^j$ and $C^{*i} = C^{*j}$, return real; return random otherwise.

With probability one, there exists one pair such that $M^i \oplus M^j = \mathsf{msb}_{n-d}(K^0 \oplus K^1)$. In this case, it holds that $U^i = X^j$ and $U^j = X^i$, from which $C^i = C^j$ follows. A similar argument holds for $C^{*i} = C^{*j}$.

A Distinguisher on a Single-key Variant. It could furthermore appear tempting to use a single-key domain-separated variant of DS-SoEM. Since the domain differs in both permutation calls, this would ensure distinct inputs in both sides of each query. However, this construction would possess only $n/2$-bit PRF security. In the following, we sketch a distinguisher, where we assume that both keys K_0 and K_1 are replaced by a single key K. We further assume $d < n/2$ for simplicity.

1. For $i \leftarrow 1..2^{n/2}$, sample $M^i = (\langle i \rangle_{n/2} \parallel 0^{n/2-d})$ to obtain C^i and store them. Associate with each M^i a plaintext $M'^i = M^i \oplus (10^{n-1-d})$ and its corresponding output C'^i.

2. For $j \leftarrow 1..2^{n/2-d}$, ask for the primitive encryption of $U^j = (\langle 0\rangle_{n/2} \parallel \langle i\rangle_{n/2-d} \parallel \langle 0\rangle_d)$ to obtain V^j. Query $U'^j = U^j \oplus (10^{n-1})$ to obtain V'^j.
3. Similarly, for $j \leftarrow 1..2^{n/2-d}$, ask for the primitive encryption of $X^j = (\langle 0\rangle n/2 \parallel \langle i\rangle_{n/2-d} \parallel \langle 1\rangle_d)$ to obtain Y^j. Query $X'^j = X^j \oplus (10^{n-1})$ to obtain Y'^j.
4. If there exists one tuple i, j s. t. $C^i = V^j \oplus Y^j$ and $C'^i = V'^j \oplus Y'^j$, output real and output random otherwise.

With probability one, there will be one collision for the real construction, whereas the probability of the $2n$-bit event is negligible in the ideal world.

8 Security Analysis of DS-SoEM

In the following, we consider $\mathsf{DS\text{-}SoEM}[\pi]_{\mathbf{K}}$ with $d \in [n-1]$, based on $\pi \twoheadleftarrow \mathsf{Perm}(\mathbb{F}_2^n)$, $K_0, K_1 \twoheadleftarrow \mathbb{F}_2^n$, and $\mathbf{K} = (K_0, K_1)$.

Theorem 3. Let $\mathbf{A}$ be a distinguisher with at most q_c construction queries and q_p primitive queries to each of $\pi^{\pm}(\cdot \parallel \langle 0\rangle_d)$ and $\pi^{\pm}(\cdot \parallel \langle 1\rangle_d)$. Let $q_c + 2q_p < 2^{n-3}$ and $q_c, q_p > 9n$. Then,

$$\begin{aligned}\mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{DS\text{-}SoEM}[\pi]_{\mathbf{K}}}(\mathbf{A}) \leq\ & \frac{(6 \cdot 2^d + 2^{2d})q_c q_p^2}{2^{2n}} + \frac{2^{2d} q_c q_p^2}{2^{3n}} + \\ & \frac{q_c + 2 + 4q_p\sqrt{3nq_c}}{2^n} + \frac{2q_c(2q_c + 2q_p)^2}{2^{2n}} .\end{aligned}$$

Proof. Again, we follow the footsteps by Chen et al.; this time, we partition the transcript τ into $\tau = \tau_c \cup \tau_0 \cup \tau_1$, where $\tau_c = \{(K_0, K_1), (M^1, C^1), \ldots, (M^{q_c}, C^{q_c})\}$ is the transcript of construction queries. We define two primitive transcripts: τ_0 and τ_1; $\tau_0 = \{(U_j^1, V_j^1), \ldots, (U_j^{q_p}, V_j^{q_p})\}$ contains exactly the queries to and responses from permutation π for which it holds that $\mathsf{lsb}_d(U^i) = \langle 0\rangle_d$. Similarly, $\tau_1 = \{(U_1^j, V_1^j), \ldots, (U_{q_p}^j, V_{q_p}^j)\}$ contains exactly the queries to and responses from permutation π for which $\mathsf{lsb}_d(U^i) = \langle 1\rangle_d$ holds. We denote the permutation inputs of construction queries, for $j \in [q_c]$ as

$$\begin{aligned}\widehat{U}^j &=^{\mathrm{def}} (M^j \oplus \mathsf{msb}_{n-d}(K_0)) \parallel \langle 0\rangle_d \quad \text{and} \\ \widehat{X}^j &=^{\mathrm{def}} (M^j \oplus \mathsf{msb}_{n-d}(K_1)) \parallel \langle 1\rangle_d\end{aligned}$$

and their corresponding outputs as $\widehat{V}^j$ and $\widehat{Y}^j$, respectively. We also use interchangeably the notations of $\widehat{U}_0^j = \widehat{U}^j$, $\widehat{U}_1^j = \widehat{X}^j$, $\widehat{V}_0^j = \widehat{V}^j$, and $\widehat{V}_1^j = \widehat{Y}^j$, respectively. Define $\mathcal{S} =^{\mathrm{def}} \{(i, j, k) : C^i \oplus K_0 \oplus K_1 = V_0^j \oplus V_1^k\}$ for $i \in [q_c]$ and $j, k \in [q_p]$. Let $\theta = q_p^2 q_c / 2^n + q_p\sqrt{3nq_c}$ be the threshold from Lemma 2.

Bad Events. We define the following bad events:

- bad_1: There exists a construction query j and two primitive queries i and k such that $(\widehat{U}_0^j = U_0^i) \wedge (\widehat{U}_1^j = U_1^k)$.

- bad_2: It holds that $|\mathcal{S}| \geq \theta$.
- bad_3: There exists a construction query j and two primitive queries i and k such that $(\widehat{U}_0^j = U_0^i) \wedge (\widehat{V}_1^j = V_1^k)$.
- bad_4: There exists a construction query j and two primitive queries i and k such that $(\widehat{U}_1^j = U_1^i) \wedge (\widehat{V}_0^j = V_0^k)$.
- bad_5: There exists a construction query j and two primitive queries i and k such that $(\widehat{U}_0^j = U_0^i) \wedge (\widehat{V}_1^j = V_0^k)$.
- bad_6: There exists a construction query j and two primitive queries i and k such that $(\widehat{U}_1^j = U_1^i) \wedge (\widehat{V}_0^j = V_1^k)$.
- bad_7: There exist two distinct construction queries j and k and two distinct primitive queries i and ℓ such that $(\widehat{U}_0^j = U_0^i) \wedge (\widehat{U}_0^k = U_0^\ell) \wedge (\widehat{V}_1^j = \widehat{V}_1^k)$.
- bad_8: There exist two distinct construction queries j and k and two distinct primitive queries i and ℓ such that $(\widehat{U}_1^j = U_1^i) \wedge (\widehat{U}_1^k = U_1^\ell) \wedge (\widehat{V}_0^j = \widehat{V}_0^k)$.
- bad_9: There exist two distinct construction queries j and k and two distinct primitive queries i and ℓ such that $(\widehat{U}_0^j = U_0^i) \wedge (\widehat{U}_1^k = U_1^\ell) \wedge (\widehat{V}_1^j = \widehat{V}_0^k)$.
- bad_{10}: There exists a construction query j such that $C^j = K_0 \oplus K_1$.

Lemma 5. Let $q_c + 2q_p < 2^{n-3}$. It holds that

$$\Pr\left[\Theta_{\text{ideal}} \in \text{BADT}\right] \leq \frac{(6 \cdot 2^d + 2^{2d}) q_c q_p^2}{2^{2n}} + \frac{2^{2d} q_c q_p^2}{2^{3n}} + \frac{q_c + 2}{2^n} + \frac{4 q_p \sqrt{3 n q_c}}{2^n}\,. \quad (5)$$

Proof. The event bad_1 considers the probability of two input collisions of a construction and two primitive queries. Thus, the probability can be upper bounded by

$$\Pr[\mathsf{bad}_1] = \sum_{j \in [q_c]} \sum_{i \in [q_p]} \sum_{k \in [q_p]} \Pr\left[\widehat{U}_0^j = U_0^i \wedge \widehat{U}_1^j = U_1^k\right] \leq \frac{q_c q_p^2}{2^{2(n-d)}}\,.$$

The probability of bad_2 is upper bounded by Lemma 2:

$$\Pr[\mathsf{bad}_2] = \Pr\left[|\mathcal{S}_{\alpha,\beta}| \geq \theta\right] \leq \frac{2}{2^n}\,.$$

The events bad_3 and bad_4 consider an input and an output collision:

$$\begin{aligned}\Pr[\mathsf{bad}_3 | \neg \mathsf{bad}_2] &= \sum_{j \in [q_c]} \sum_{i \in [q_p]} \sum_{k \in [q_p]} \Pr\left[\widehat{U}_0^j = U_0^i \wedge \widehat{V}_1^j = V_1^k\right] \\ &\leq \frac{q_c q_p^2}{2^{n+(n-d)}} + \frac{q_p \sqrt{3 n q_c}}{2^n}\,.\end{aligned}$$

The probability $\Pr[\mathsf{bad}_4 | \neg \mathsf{bad}_2]$ can be upper bounded by a similar argument. Events bad_5 and bad_6 study an input collision between a construction and a primitive query, that leads to a conflict of the other output for that construction query. The probabilities can be upper bounded by

$$\Pr[\mathsf{bad}_5 | \neg \mathsf{bad}_2] = \sum_{j \in [q_c]} \sum_{i \in [q_p]} \sum_{k \in [q_p]} \Pr\left[\widehat{U}_0^j = U_0^i \wedge \widehat{V}_1^j = V_0^k\right]$$

$$\leq \frac{q_c q_p^2}{(2^n - 1)(2^{n-d})} + \frac{q_p \sqrt{3 n q_c}}{2^n} \, .$$

The bound of $\Pr[\mathsf{bad}_6 | \neg \mathsf{bad}_2]$ is again analogous.
The event bad_7 requires first two separate input collisions between a construction query and a primitive query each, and the output collisions between their other permutation-calls outputs. This probability can be upper bounded by

$$\Pr[\mathsf{bad}_7] \leq \sum_{1 \leq j < k \leq q_c} \sum_{1 \leq i < \ell \leq q_p} \Pr\left[\widehat{U}_0^j = U_0^i \wedge \widehat{U}_1^k = U_0^\ell \wedge \widehat{V}_1^j = \widehat{V}_1^k\right] \leq \frac{\binom{q_c}{2}\binom{q_p}{2}}{2^{2(n-d)} 2^n} \, .$$

The probabilities of bad_8 and bad_9 can be bounded in a similar manner. The probability of the latter is

$$\Pr[\mathsf{bad}_9] \leq \sum_{1 \leq j < k \leq q_c} \sum_{i \in [q_p]} \sum_{\ell \in [q_p]} \Pr\left[\widehat{U}_0^j = U_0^i \wedge \widehat{U}_1^k = U_0^\ell \wedge \widehat{V}_1^j = \widehat{V}_0^k\right] \leq \frac{\binom{q_c}{2} q_p^2}{2^{2(n-d)} 2^n} \, .$$

Note that events such as

$$(\widehat{U}_0^j = U_0^i) \wedge (\widehat{U}_0^k = U_0^\ell) \wedge (\widehat{V}_0^j = \widehat{V}_0^k)$$

can not occur since we assume that $\mathbf{A}$ does not ask trivial queries. Thus, the distinct construction queries $j \neq k$ prevent that $\widehat{U}_0^j = \widehat{U}_0^k$ would hold, which implies that $\widehat{V}_0^j \neq \widehat{V}_0^k$. A similar argument holds for

$$(\widehat{U}_1^j = U_1^i) \wedge (\widehat{U}_1^k = U_1^\ell) \wedge (\widehat{V}_1^j = \widehat{V}_1^k) \, .$$

Finally, bad_{10} represents the event that a construction query obtains equal outputs from both permutation calls, while the inputs are always distinct. Thus, $V^j \oplus Y^j = C^j \oplus K_0 \oplus K_1$ can never be zero for the real construction. The probability is upper bounded by

$$\Pr[\mathsf{bad}_{10}] = \sum_{j \in [q_c]} \Pr\left[\widehat{V}_0^j = \widehat{V}_1^j\right] \leq \frac{q_c}{2^n} \, .$$

The bound in Lemma 5 follows from

$$\sum_{i=1}^{2} \Pr\left[\mathsf{bad}_i\right] + \sum_{i=3}^{6} \Pr\left[\mathsf{bad}_i | \neg \mathsf{bad}_2\right] + \sum_{i=7}^{10} \Pr\left[\mathsf{bad}_i\right] \, . \quad \square$$

Good Transcripts. It remains to consider the interpolation probability of good attainable transcripts.

Lemma 6. It holds that

$$\frac{\Pr[\Theta_{\text{real}} = \tau]}{\Pr[\Theta_{\text{ideal}} = \tau]} \geq 1 - \frac{2 q_c (2 q_p + 2 q_c)^2}{2^{2n}} . \tag{6}$$

We note that this part is almost exactly as the part of good transcripts in the proof of $\mathsf{SoEM22}$ by Chen et al. [CLM19]. Moreover, similar results for secret permutations have been derived at several places, for example, by Jha and Nandi [JN18] and Datta et al. [DDN+17].

Proof. Again, we can write

$$\frac{\Pr[\Theta_{\mathrm{real}} = \tau]}{\Pr[\Theta_{\mathrm{ideal}} = \tau]} = \frac{|\mathsf{Comp}_{\mathrm{real}}(\tau)| \cdot |\mathsf{All}_{\mathrm{ideal}}(\tau)|}{|\mathsf{Comp}_{\mathrm{ideal}}(\tau)| \cdot |\mathsf{All}_{\mathrm{real}}(\tau)|}.$$

Three out of four terms are again easy to bound:

$$|\mathsf{All}_{\mathrm{real}}(\tau)| = 2^{2n} \cdot (2^n)!$$

since there exist 2^{2n} keys and $2^n!$ independent permutations π. A similar argument holds in the ideal world, combined with $(2^n)^{2^n}$ random functions for the answers to the construction queries:

$$|\mathsf{All}_{\mathrm{ideal}}(\tau)| = 2^{2n} \cdot (2^n)! \cdot (2^n)^{2^n}$$

Moreover, we can bound

$$|\mathsf{Comp}_{\mathrm{ideal}}(\tau)| = (2^n)^{2^n - q_c} \cdot (2^n - 2q_p)!$$

compatible oracles exist in the ideal world: there exist $(2^n)^{2^n - q_c}$ oracles that produce the correct construction-query outputs for the $2^n - q_c$ remaining non-queried inputs, and $(2^n - 2q_p)!$ compatible permutations π. So, we obtain

$$\frac{\Pr[\Theta_{\mathrm{real}} = \tau]}{\Pr[\Theta_{\mathrm{ideal}} = \tau]} \geq \frac{|\mathsf{Comp}_{\mathrm{real}}(\tau)| \cdot 2^{2n} \cdot (2^n)! \cdot (2^n)^{2^n}}{(2^n)^{2^n - q_c} \cdot (2^n - 2q_p)! \cdot 2^{2n} \cdot (2^n)!} = \frac{|\mathsf{Comp}_{\mathrm{real}}(\tau)| \cdot (2^n)^{q_c}}{(2^n - 2q_p)!}.$$

It remains to determine $|\mathsf{Comp}_{\mathrm{real}}(\tau)|$. We reuse the claim by Chen et al.:

Claim. For a good transcript, $\tau \in \textsc{GoodT}$, any construction query $(M^j, C^j) \in \tau_c$ collides with at most one primitive query (U^i_α, V^i_α) for some $\alpha \in \{0, 1\}$, but never with multiple primitive queries.

We regroup the queries from τ_c, τ_0, and τ_1 to τ_c^{new}, τ_0^{new}, and τ_1^{new}. The new transcript sets are initialized by their corresponding old parts, and reordered:

- If there exists an i such that $\widehat{U}^j_0 = U^i_0$, then (M^j, C^j) is removed from τ_c^{new} and $(U^i_1, V^i_1) = (\widehat{U}^j_1, \widehat{V}^j_1)$ is added to τ_1^{new}.
- If there exists an i such that $\widehat{U}^j_1 = U^i_1$, then (M^j, C^j) is removed from τ_c^{new} and $(U^i_0, V^i_0) = (\widehat{U}^j_0, \widehat{V}^j_0)$ is added to τ_0^{new}.

Given q_c constructions queries and q_p queries in τ_0 and τ_1 each, we denote the number of queries moved from τ_c into the primitive transcript τ_0 and τ_1 by s_0 and s_1. We define $s = s_0 + s_1$ for brevity.

The number of queries in the new construction transcript is denoted by $q' = q_c - s$. In the following, for a given transcript τ_p^{new}, it remains to count the number of permutations π that are compatible with the transcript. The set of occurred (i.e., prohibited) outputs V_0 (for some U_0 with $\mathsf{lsb}_d(U_0) = 0$) and V_1 (for some U_1 with $\mathsf{lsb}_d(U_1) = 1$) of π are denoted by V_0^{out} and V_1^{out}, respectively. For $\alpha = 0, \ldots, q' - 1$, let

$$\lambda_{\alpha+1} \overset{\text{def}}{=} \left|\left\{(V_0^1, \ldots, V_0^{\alpha+1}, V_1^1, \ldots, V_1^{\alpha+1})\right\}\right| \tag{7}$$

be the number of solutions that satisfy

(1) $\left\{(V_0^1, \ldots, V_0^{\alpha}, V_1^1, \ldots, V_1^{\alpha})\right\}$ satisfy the conditions recursively,
(2) It holds that

$$V_0^{\alpha+1} \oplus V_1^{\alpha+1} = C^{\alpha+1} \oplus K_0 \oplus K_1. \tag{8}$$

(3.0) It holds that $V_0^{\alpha+1} \notin \{V_0^1, \ldots, V_0^{\alpha}, V_1^1, \ldots, V_1^{\alpha}\} \cup V_0^{\mathsf{out}} \cup V_1^{\mathsf{out}}$.
(3.1) It holds that $V_1^{\alpha+1} \notin \{V_0^1, \ldots, V_0^{\alpha}, V_1^1, \ldots, V_1^{\alpha}\} \cup V_0^{\mathsf{out}} \cup V_1^{\mathsf{out}}$.

Then, the goal is to define a recursive expression for $\lambda_{\alpha+1}$ from λ_α such that a lower bound can be found for the expression $\lambda_{\alpha+1}/\lambda_\alpha$. It holds that

$$|\mathsf{Comp}_{\mathrm{real}}(\tau)| = \lambda_{q'} \cdot (2^n - (q_1 + q_2 + 2q'))!\,.$$

We obtain

$$\frac{\Pr[\Theta_{\mathrm{real}} = \tau]}{\Pr[\Theta_{\mathrm{ideal}} = \tau]} = \frac{\lambda_{q'} \cdot (2^n - (q_1 + q_2 + 2q'))! \cdot (2^n)^{q_c}}{(2^n - 2q_p)!}\,. \tag{9}$$

Let $\mathcal{B}_{(1,2)}$ denote the set of solutions that comply with only Conditions (1) and (2), without considering Condition (3). Moreover, let $\mathcal{B}_{(3.0:i)}$ denote the set of solutions compatible with Conditions (1) and (2), but not with $(3.0 : i)$ and define $\mathcal{B}_{(3.1:i)}$ in the natural manner. From the inclusion-exclusion principle, it follows that

$$\begin{aligned}
\lambda_{\alpha+1} &= \left|\mathcal{B}_{(1,2)}\right| - \left| \bigcup_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+|V_1^{\mathsf{out}}|} \mathcal{B}_{(3.0:i)} \cup \bigcup_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+|V_1^{\mathsf{out}}|} |\mathcal{B}_{(3.1:i)}| \right| \\
&\geq \left|\mathcal{B}_{(1,2)}\right| - \left(\sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+|V_1^{\mathsf{out}}|} \left|\mathcal{B}_{(3.0:i)}\right| \right) - \left(\sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+|V_1^{\mathsf{out}}|} \left|\mathcal{B}_{(3.1:i)}\right| \right) \\
&\quad + \left(\sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+|V_1^{\mathsf{out}}|} \sum_{i'=1}^{\alpha+|V_0^{\mathsf{out}}|+|V_1^{\mathsf{out}}|} \left|\mathcal{B}_{(3.0:i)} \cap \mathcal{B}_{(3.1:i')}\right| \right) \\
&\geq \left|\mathcal{B}_{(1,2)}\right| - \left(\sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+|V_1^{\mathsf{out}}|} \left|\mathcal{B}_{(3.0:i)}\right| \right) - \left(\sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+|V_1^{\mathsf{out}}|} \left|\mathcal{B}_{(3.1:i)}\right| \right)
\end{aligned}$$

$$\geq 2^n \cdot \lambda_\alpha - \sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+|V_1^{\mathsf{out}}|} \lambda_\alpha - \sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+|V_1^{\mathsf{out}}|} \lambda_\alpha .$$

So, it follows that

$$\begin{aligned} \lambda_{\alpha+1} &\geq 2^n \cdot \lambda_\alpha - (\alpha + q_1 + q_2) \cdot \lambda_\alpha - (\alpha + q_1 + q_2) \cdot \lambda_\alpha \\ &= 2^n \cdot \lambda_\alpha - 2(\alpha + q_1 + q_2) \cdot \lambda_\alpha . \end{aligned}$$

Therefore,

$$\frac{\lambda_{\alpha+1}}{\lambda_\alpha} \geq 2^n - 2\alpha - 2q_1 - 2q_2 \geq 1 \,,$$

with $\lambda_0 = 1$. It follows from Equation (9) that

$$\begin{aligned} (9) &= \prod_{j=0}^{s_0+s_1-1} \frac{2^n}{2^n - 2q_p - j} \cdot \prod_{i=0}^{q'-1} \frac{\lambda_{i+1}}{\lambda_i} \cdot \frac{2^n}{(2^n - q_1 - q_2 - i)(2^n - q_1 - q_2 - q' - i)} \\ &\geq \prod_{i=0}^{q'-1} \frac{(2^n - 2i - 2q_1 - 2q_2) \cdot 2^n}{(2^n - q_1 - q_2 - i)(2^n - q_1 - q_2 - q' - i)} \\ &\geq \prod_{i=0}^{q'-1} \left(1 - \frac{(q_1 + q_2 + q' + i)(q_1 + q_2 + i) - 2^n q'}{(2^n - q_1 - q_2 - i)(2^n - q_1 - q_2 - q' - i)}\right) \\ &\geq \prod_{i=0}^{q'-1} \left(1 - \frac{(q_1 + q_2 + q' + i)(q_1 + q_2 + i)}{(2^n - q_1 - q_2 - q')(2^n - q_1 - q_2 - q' - q')}\right) \\ &\geq \prod_{i=0}^{q'-1} \left(1 - \frac{(q_1 + q_2 + 2q')^2}{(2^n - q_1 - q_2 - 2q')^2}\right) \\ &\geq \left(1 - \frac{(q_1 + q_2 + 2q')^2}{(2^n - q_1 - q_2 - 2q')^2}\right)^{q'} \\ &\geq 1 - \frac{q'(q_1 + q_2 + 2q')^2}{(2^n - q_1 - q_2 - 2q')^2} \\ &\geq 1 - \frac{2q'(q_1 + q_2 + 2q')^2}{2^{2n}} \geq 1 - \frac{2q_c(2q_p + 2q_c)^2}{2^{2n}} \,, \end{aligned}$$

where we used that $q_p + q_c \ll 2^{n-3}$. □

Our claim in Theorem 3 follows from Lemma 1, 5, and 6. □

9 Security Analysis of DS-CENCPP

In what remains, we study the nE security of DS-CENCPP. As before, let $\pi \twoheadleftarrow \mathsf{Perm}(\mathbb{F}_2^n)$ and $K_0, \ldots, K_w \twoheadleftarrow \mathcal{K}$ be independent secret keys; we write $\mathbf{K} =$

$(K_0, \ldots, K_w)$ for brevity. Again, we conduct a two-step analysis, where we consider (1) the PRF security of $\mathsf{DS\text{-}XORPP}[\pi, w]$ and (2) the PRF security of $\mathsf{DS\text{-}CENCPP}[\pi, w]$.

Theorem 4. It holds that

$$\mathbf{Adv}^{\mathsf{nE}}_{\mathsf{DS\text{-}CENCPP}[\pi,w]_{\mathbf{K}}}(q_p, q_c, \sigma) \leq \mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{DS\text{-}XORPP}[\pi,w]_{\mathbf{K}}}\left(q_p, \frac{m}{w} q_c, \sigma\right).$$

The proof follows a similar argumentation as that of CENCPP.

Theorem 5. Let $v =^{\text{def}} w + 1$, $q_c + v q_p \leq 2^{n-w}$, and $q_p, q_c > 9n$. It holds that

$$\begin{aligned}\mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{DS\text{-}XORPP}[\pi,w]_{\mathbf{K}}}(q_p, q_c, \sigma) \leq\; & \frac{\left(v^2 2^{2d} + v^2 2^d + v^3 2^d\right) q_c q_p^2 + v^3 2^d q_c q_p}{2^{2n}} + \\ & \frac{v^4 2^{2d} q_c^2 q_p^2}{2^{3n}} + \frac{v^2 q_c}{2^n} + \frac{3v^2 q_c^3 + 6v^3 q_c^2 q_p + 4v^4 q_c q_p^2}{2^{2n}} + \\ & \frac{(w+1)^2 + (w+1)^2 q_p \sqrt{3nq_c}}{2^n}.\end{aligned}$$

Proof. Again, we employ the proof strategy from XORPP. Here, the adversary can query q_p primitive queries to each domain-separated primitive $\pi^{\pm}(\cdot \parallel \langle i \rangle_d)$. We define sets $\mathcal{S}_{\alpha,\beta,\gamma} =^{\text{def}} \{(i,j,k) : C^i_\alpha \oplus (1 + 2^\alpha) K_0 \oplus (1 + 2^{2\alpha}) K_1 = V^j_\beta \oplus V^k_\gamma\}$ for $i \in [q_c]$, $j, k \in [q_p]$, $\alpha \in [w]$ and $\beta, \gamma \in [0..w]$. Let $\theta = q_p^2 q_c / 2^n + q_p \sqrt{3nq_c}$ be the threshold from Lemma 2.

Bad Events. We study the following bad events:

- bad_1: There exists a construction query index $j \in [q_c]$, two primitive query indices $i, k \in [q_p]$ and distinct permutation indices $\alpha, \beta \in [0..w]$ such that $(\widehat{U}^j_\alpha = U^i_\alpha) \wedge (\widehat{U}^j_\beta = U^k_\beta)$.
- bad_2: There exist $\alpha \in [w]$ and distinct $\beta, \gamma \in [0..w]$ such that $|\mathcal{S}_{\alpha,\beta,\gamma}| \geq \theta$.
- bad_3: There exists a construction query index $j \in [q_c]$, two primitive query indices $i, k \in [q_p]$ and permutation indices $\alpha, \beta \in [0..w]$ such that $(\widehat{U}^j_\alpha = U^i_\alpha) \wedge (\widehat{V}^j_\beta = V^k_\beta)$.
- bad_4: There exists a construction query index $j \in [q_c]$, two primitive query indices $i, k \in [q_p]$ and distinct permutation indices $\alpha, \beta \in [0..w]$ as well as any $\gamma \in [0..w]$ with $\beta \neq \gamma$ such that $(\widehat{U}^j_\alpha = U^i_\alpha) \wedge (\widehat{V}^j_\beta = V^k_\gamma)$.
- bad_5: There exists a construction query index j and a primitive query index i and k and distinct permutation indices $\alpha, \beta \in [0..w]$ as well as any $\gamma \in [0..w]$ with $\beta \neq \gamma$ such that $(\widehat{U}^j_\alpha = U^i_\alpha) \wedge (\widehat{V}^j_\beta = \widehat{V}^j_\gamma)$.
- bad_6: There exist distinct construction query indices j, ℓ and primitive query indices i and k as well as distinct permutation indices $\alpha, \beta \in [0..w]$ and any $\gamma, \delta \in [0..w]$ such that $(\widehat{U}^j_\alpha = U^i_\alpha) \wedge (\widehat{U}^j_\gamma = U^i_\gamma) \wedge (\widehat{V}^j_\beta = \widehat{V}^\ell_\delta)$.
- bad_7: There exists a construction query index j and a permutation index $\alpha \in [w]$ such that $C^j_\alpha = (K_0 \oplus K_1) \oplus (2^\alpha K_0 \oplus 2^{2\alpha} K_1)$.

Our claim in Theorem 5 follows from Lemmas 7, 8, and 1. □

Lemma 7. Let $v =^{\text{def}} w + 1$ and $q_c + v \cdot q_p < 2^{n-3}$. It holds that

$$\Pr\left[\Theta_{\text{ideal}} \in \text{BadT}\right] \leq \frac{\left(v^2 2^{2d} + v^2 2^d + v^3 2^d\right) q_c q_p^2 + v^3 2^d q_c q_p}{2^{2n}} + \frac{v^4 2^{2d} q_c^2 q_p^2}{2^{3n}} + \frac{v^2 q_c + v^3 + v^3 q_p \sqrt{3nq_c}}{2^n}.$$

For space limitations, the proof is deferred to Appendix A.

Lemma 8. Let $v =^{\text{def}} w + 1$ It holds that

$$\frac{\Pr[\Theta_{\text{real}} = \tau]}{\Pr[\Theta_{\text{ideal}} = \tau]} \geq 1 - \frac{3v^2 q_c^3 + 6v^3 q_c^2 q_p + 4v^4 q_c q_p^2}{2^{2n}}.$$

The proof is deferred to Appendix B.

10 Conclusion

This work has proposed a variant of CENC from public permutations, CENCPP. Consequently, it is straight-forward to obtain a nonce-based encryption scheme or in form of its underlying component XORPP, a fixed-input-length variable-output-length PRF with security of up to $O(2^{2n/3}/w^2)$ queries. Our result can be combined with a beyond-birthday-secure MAC from public permutations in the close future to obtain an authenticated encryption scheme.
We note that the doubling-based key schedule ensures pairwise independent keys for all pairs of permutation inputs in XORPP and DS-XORPP. Although the key masks can be cached, for values of $w \leq 2$, the choice of keys can be improved in terms of computations. For $w = 1$, XORPP degenerates to the SoEM construction, and can simply use (K_0, K_1) for the permutation calls. For $w = 2$, XORPP can employ the key masks $(K_0, K_0 \oplus K_1, K_1)$ for the three calls to the permutations to ensure independent keys without the need for doubling.

Acknowledgments. Eik List has been supported by DFG Grant LU 608/9-1.

References

AJN14. Jean-Philippe Aumasson, Philipp Jovanovic, and Samuel Neves. NORX: Parallel and Scalable AEAD. In Miroslaw Kutyłowski and Jaideep Vaidya, editors, *ESORICS II*, volume 8713 of *LNCS*, pages 19–36. Springer, 2014.

BDH+17. Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Farfalle: parallel permutation-based cryptography. *IACR Trans. Symmetric Cryptol.*, 2017(4):1–38, 2017.

BDP+16. Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles van Assche, and Ronny van Keer. Ketje v2. 2016. Submission to the CAESAR competition http://competitions.cr.yp.to/caesar-submissions.html.

BGIM19. Zhenzhen Bao, Jian Guo, Tetsu Iwata, and Kazuhiko Minematsu. ZOCB and ZOTR: Tweakable Blockcipher Modes for Authenticated Encryption with Full Absorption. *IACR Trans. Symmetric Cryptol.*, 2019(2):1–54, 2019.

BL16. Karthikeyan Bhargavan and Gaëtan Leurent. On the Practical (In-)Security of 64-bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS*, pages 456–467. ACM, 2016.

CLL$^+$14. Shan Chen, Rodolphe Lampe, Jooyoung Lee, Yannick Seurin, and John P. Steinberger. Minimizing the Two-Round Even-Mansour Cipher. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO I*, volume 8616 of *LNCS*, pages 39–56. Springer, 2014.

CLM19. Yu Long Chen, Eran Lambooij, and Bart Mennink. How to Build Pseudorandom Functions from Public Random Permutations. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO I*, volume 11692 of *LNCS*, pages 266–293. Springer, 2019.

CS14. Shan Chen and John P. Steinberger. Tight Security Bounds for Key-Alternating Ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT*, volume 8441 of *LNCS*, pages 327–350. Springer, 2014. Full version at https://eprint.iacr.org/2013/222.

CS15. Benoît Cogliati and Yannick Seurin. Beyond-Birthday-Bound Security for Tweakable Even-Mansour Ciphers with Linear Tweak and Key Mixing. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT II*, volume 9453 of *LNCS*, pages 134–158. Springer, 2015.

CS18. Benoît Cogliati and Yannick Seurin. Analysis of the single-permutation encrypted Davies-Meyer construction. *Des. Codes Cryptogr.*, 86(12):2703–2723, 2018.

DDKS13. Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. Key Recovery Attacks on 3-round Even-Mansour, 8-step LED-128, and Full AES^2. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT I*, volume 8269 of *Lecture Notes in Computer Science*, pages 337–356. Springer, 2013.

DDN$^+$17. Nilanjan Datta, Avijit Dutta, Mridul Nandi, Goutam Paul, and Liting Zhang. Single Key Variant of PMAC_Plus. *IACR Trans. Symmetric Cryptol.*, 2017(4):268–305, 2017.

DDNY18. Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Kan Yasuda. Encrypt or Decrypt? To Make a Single-Key Beyond Birthday Secure Nonce-Based MAC. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO I*, volume 10991 of *Lecture Notes in Computer Science*, pages 631–661. Springer, 2018.

DEMS16. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2 Submission to the CAESAR Competition. September 15 2016. Submission to the CAESAR competition http://competitions.cr.yp.to/caesar-submissions.html.

DIS$^+$18. Patrick Derbez, Tetsu Iwata, Ling Sun, Siwei Sun, Yosuke Todo, Haoyang Wang, and Meiqin Wang. Cryptanalysis of AES-PRF and Its Dual. *IACR Trans. Symmetric Cryptol.*, 2018(2):161–191, 2018.

DKS12. Orr Dunkelman, Nathan Keller, and Adi Shamir. Minimalism in Cryptography: The Even-Mansour Scheme Revisited. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 336–354. Springer, 2012.

GJMN16. Robert Granger, Philipp Jovanovic, Bart Mennink, and Samuel Neves. Improved Masking for Tweakable Blockciphers with Applications to Authenticated Encryption. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT I*, volume 9665 of *Lecture Notes in Computer Science*, pages 263–293. Springer, 2016.

GSWG19. Chun Guo, Yaobin Shen, Lei Wang, and Dawu Gu. Beyond-birthday secure domain-preserving PRFs from a single permutation. *Des. Codes Cryptogr.*, 87(6):1297–1322, 2019.

HT16. Viet Tung Hoang and Stefano Tessaro. Key-Alternating Ciphers and Key-Length Extension: Exact Bounds and Multi-user Security. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO I*, volume 9814 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2016.

IM16. Tetsu Iwata and Kazuhiko Minematsu. Stronger Security Variants of GCM-SIV. *IACR Trans. Symmetric Cryptol.*, 2016(1):134–157, 2016.

IMPS17. Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A Fast Tweakable Block Cipher Mode for Highly Secure Message Authentication. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO, Part III*, volume 10403 of *LNCS*, pages 34–65. Springer, 2017. Full version at https://eprint.iacr.org/2017/535.

IMV16. Tetsu Iwata, Bart Mennink, and Damian Vizár. CENC is Optimally Secure. *IACR Cryptology ePrint Archive*, 2016:1087, 2016.

Iwa06. Tetsu Iwata. New Blockcipher Modes of Operation with Beyond the Birthday Bound Security. In Matthew J. B. Robshaw, editor, *FSE*, volume 4047 of *LNCS*, pages 310–327. Springer, 2006.

Iwa07. Tetsu Iwata. Tightness of the Security Bound of CENC. In Eli Biham, Helena Handschuh, Stefan Lucks, and Vincent Rijmen, editors, *Symmetric Cryptography*, volume 07021 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.

JN18. Ashwin Jha and Mridul Nandi. A Survey on Applications of H-Technique: Revisiting Security Analysis of PRP and PRF. 2018.

JNP14. Jérémy Jean, Ivica Nikolić, and Thomas Peyrin. Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT II*, volume 8874 of *LNCS*, pages 274–288. Springer, 2014.

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

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

Min14. Kazuhiko Minematsu. Parallelizable Rate-1 Authenticated Encryption from Pseudorandom Functions. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT*, volume 8441 of *LNCS*, pages 275–292. Springer, 2014. Full version at https://eprint.iacr.org/2013/628.pdf.

MMH$^+$14. Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers. In Antoine Joux and Amr M. Youssef, editors, *SAC*, volume 8781 of *LNCS*, pages 306–323. Springer, 2014.

MN17a. Bart Mennink and Samuel Neves. Encrypted Davies-Meyer and Its Dual: Towards Optimal Security Using Mirror Theory. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO, Part III*, volume 10403 of *LNCS*, pages 556–583. Springer, 2017. Full version at `https://eprint.iacr.org/2017/473`.

MN17b. Bart Mennink and Samuel Neves. Optimal PRFs from Blockcipher Designs. *IACR Trans. Symmetric Cryptol.*, 2017(3):228–252, 2017.

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

Nai15. Yusuke Naito. Full PRF-Secure Message Authentication Code Based on Tweakable Block Cipher. In Man Ho Au and Atsuko Miyaji, editors, *ProvSec*, volume 9451 of *LNCS*, pages 167–182. Springer, 2015.

Nan20. Mridul Nandi. Mind the Composition: Birthday Bound Attacks on EWCDMD and SoKAC21. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT*, Lecture Notes in Computer Science. Springer, 2020. To appear.

NIS01. NIST. Advanced Encryption Standard (AES). *Federal Information Processing Standards (FIPS) Publication*, 197, Nov 26 2001.

NIS15. NIST. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. *Federal Information Processing Standards (FIPS) Publication*, 202, 2015.

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

Pat10. Jacques Patarin. Introduction to Mirror Theory: Analysis of Systems of Linear Equalities and Linear Non Equalities for Cryptography. *IACR Cryptology ePrint Archive*, 2010:287, 2010.

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

SB15. Markku-Juhani O. Saarinen and Billy Bob Brumley. STRIBOBr2: WHIRLBOB. Aug 28 2015. Submission to the CAESAR competition http://competitions.cr.yp.to/caesar-submissions.html.

A Proof of Lemma 7

We restate the lemma to aid the reader.

Lemma 7. Let $v =^{\text{def}} w + 1$ and $q_c + v \cdot q_p < 2^{n-3}$. It holds that

$$\Pr\left[\Theta_{\text{ideal}} \in \textsc{BadT}\right] \leq \frac{\left(v^2 2^{2d} + v^2 2^d + v^3 2^d\right) q_c q_p^2 + v^3 2^d q_c q_p}{2^{2n}} + \frac{v^4 2^{2d} q_c^2 q_p^2}{2^{3n}} + \frac{v^2 q_c + v^3 + v^3 q_p \sqrt{3 n q_c}}{2^n}.$$

Proof. Again, we can go through the bad events. The first event bad_1 considers the probability of two input collisions of a construction and two primitive queries. Thus, the probability can be upper bounded by

$$\Pr[\mathsf{bad}_1] = \sum_{j\in[q_c]} \sum_{i\in[q_p]} \sum_{k\in[q_p]} \sum_{0\le\alpha<\beta\le w} \Pr\left[\widehat{U}^j_\alpha = U^i_\alpha \wedge \widehat{U}^j_\beta = U^k_\beta\right] \le \frac{\binom{w+1}{2} q_c q_p^2}{2^{2(n-d)}} .$$

The event bad_2 considers the probability of a sum set with too many elements. For fixed α, β, γ, the probability of this event is given by Lemma 2. Over the union bound of all combinations of α and β, we obtain that

$$\Pr[\mathsf{bad}_2] = \sum_{\alpha\in[w]} \sum_{0\le\beta<\gamma\le w} \Pr\left[|\mathcal{S}_{\alpha,\beta,\gamma}| \ge \theta\right] \le \frac{2w \cdot \binom{w+1}{2}}{2^n} .$$

The event bad_3 considers an input and an output collision. Given that bad_2 does not hold, we have

$$\begin{aligned}\Pr[\mathsf{bad}_3 | \neg\mathsf{bad}_2] &\le \sum_{j\in[q_c]} \sum_{i\in[q_p]} \sum_{k\in[q_p]} \sum_{\alpha,\beta\in[0..w]} \Pr\left[\widehat{U}^j_\alpha = U^i_\alpha \wedge \widehat{V}^j_\beta = V^k_\beta\right] \\ &\le \frac{(w+1)^2 q_c q_p^2}{2^{n+(n-d)}} + \frac{(w+1)^3 q_p \sqrt{3nq_c}}{2^n} .\end{aligned}$$

The bound of bad_4 considers an output collision between $\widehat{V}^j_\beta = V^k_\gamma$ for any primitive query output. Given that bad_2 does not hold, we have

$$\begin{aligned}\Pr[\mathsf{bad}_4 | \neg\mathsf{bad}_2] &\le \sum_{j\in[q_c]} \sum_{i\in[q_p]} \sum_{k\in[q_p]} \sum_{\alpha\in[0..w]} \sum_{0\le\beta<\gamma\le w} \Pr\left[\widehat{U}^j_\alpha = U^i_\alpha \wedge \widehat{V}^j_\beta = V^k_\gamma\right] \\ &\le \frac{(w+1)^3 q_c q_p^2}{2^{n+(n-d)}} + \frac{(w+1)^3 q_p \sqrt{3nq_c}}{2^n} .\end{aligned}$$

The event bad_5 studies an input collision between a construction and a primitive query, that leads to a conflict of the other output for that construction query. The probability can be upper bounded by

$$\begin{aligned}\Pr[\mathsf{bad}_5] &\le \sum_{j\in[q_c]} \sum_{i\in[q_p]} \sum_{\alpha\in[0..w]} \sum_{0\le\beta<\gamma\le w} \Pr\left[\widehat{U}^j_\alpha = U^i_\alpha \wedge \widehat{V}^j_\beta = \widehat{V}^j_\gamma\right] \\ &\le \frac{(w+1)\binom{w+1}{2} q_c q_p}{2^{n+(n-d)}} .\end{aligned}$$

The event bad_6 requires first two separate input collisions between a construction query and a primitive query each, and the output collisions between their other permutation-calls outputs. This probability can be upper bounded by

$$\Pr[\mathsf{bad}_6] \le \sum_{1\le j<k\le q_c} \sum_{i\in[q_p]} \sum_{\ell\in[q_p]} \sum_{\alpha,\beta,\gamma,\delta\in[0..w]} \Pr\left[\widehat{U}^j_\alpha = U^i_\alpha \wedge \widehat{U}^k_\gamma = U^\ell_\gamma \wedge \widehat{V}^j_\beta = \widehat{V}^j_\delta\right]$$

$$\leq \frac{(w+1)^4 \binom{q_c}{2} q_p^2}{2^{2(n-d)} 2^n}.$$

Finally, bad_7 represents the event that a construction query obtains equal outputs from both permutation calls, while the inputs are always distinct. Thus, $\widehat{V}_\alpha^j \oplus \widehat{V}_\beta^j = C_\alpha^j \oplus C_\beta^j \oplus (2^\alpha K_0 \oplus 2^{2\alpha} K_1) \oplus (2^\beta K_0 \oplus 2^{2\beta} K_1)$ can never be zero for the real construction. The probability is upper bounded by

$$\Pr[\mathsf{bad}_7] \leq \sum_{j \in [q_c]} \sum_{0 \leq \alpha < \beta \leq w} \Pr\left[\widehat{V}_\alpha^j = \widehat{V}_\beta^j\right] \leq \frac{\binom{w+1}{2} q_c}{2^n}.$$

The bound in Lemma 5 follows from the sum of probabilities of the individual bad events. □

B Proof of Lemma 8

It remains to consider the interpolation probability of good attainable transcripts. Again, we restate the lemma to aid the reader.

Lemma 8. Let $v =^{\text{def}} w + 1$ It holds that

$$\frac{\Pr[\Theta_{\text{real}} = \tau]}{\Pr[\Theta_{\text{ideal}} = \tau]} \geq 1 - \frac{3v^2 q_c^3 + 6v^3 q_c^2 q_p + 4v^4 q_c q_p^2}{2^{2n}}.$$

Proof. Given $\tau \in \textsc{GoodT}$, we compute the probability of its occurrences in both worlds. Let $\mathsf{All}_{\text{real}}(\tau)$ denote the set of all oracles in the real world, and $\mathsf{All}_{\text{ideal}}(\tau)$ the set of all oracles in the ideal world. Let $\mathsf{Comp}_{\text{real}}(\tau)$ denote the fraction of oracles in the real world that are compatible with τ and $\mathsf{Comp}_{\text{ideal}}(\tau)$ the corresponding fraction in the ideal world. It holds that

$$\frac{\Pr[\Theta_{\text{real}} = \tau]}{\Pr[\Theta_{\text{ideal}} = \tau]} = \frac{|\mathsf{Comp}_{\text{real}}(\tau)| \cdot |\mathsf{All}_{\text{ideal}}(\tau)|}{|\mathsf{Comp}_{\text{ideal}}(\tau)| \cdot |\mathsf{All}_{\text{real}}(\tau)|}.$$

We can easily bound three out of four terms:

$$|\mathsf{All}_{\text{real}}(\tau)| = (2^n)^{w+1} \cdot (2^n)!$$

since there exist $(2^n)^{w+1}$ keys and $2^n!$ possible permutations. The same argument holds in the ideal world

$$|\mathsf{All}_{\text{ideal}}(\tau)| = (2^n)^{w+1} \cdot (2^n!)^{w+1} \cdot (2^{wn})^{2^n},$$

combined with $(2^{wn})^{2^n}$ random functions for the answers to the construction queries. Moreover,

$$|\mathsf{Comp}_{\text{ideal}}(\tau)| = (2^{wn})^{2^n - q_c} \cdot (2^n - (w+1) \cdot q_p)!$$

compatible oracles exist in the ideal world, where $(2^{wn})^{2^n-q_c}$ are the oracles that produce the correct construction-query outputs for the $2^n - q_c$ remaining non-queried inputs, and for all permutations, there exist $(2^n-(w+1)q_p)!$ compatible primitives each.

It remains to determine $|\mathsf{Comp}_{\mathrm{real}}(\tau)|$. Chen et al. regrouped the queries from the transcript parts. We generalize their claim [CLM19] to the following to cover all $w+1$ permutations:

Claim. For a good transcript, $\tau \in \textsc{GoodT}$, any construction query $(M^j, C^j_\alpha) \in \tau_c$ collides with at most one primitive query (U^i_α, V^i_α) for some $\alpha \in [0..w]$, but never with multiple primitive queries.

We regroup the queries from τ_c, τ_0, $\dots$, τ_w to τ_c^{new}, τ_0^{new}, $\dots$, τ_w^{new}. The new transcript sets are initialized by their corresponding old parts, and reordered as follows:

If there exist $j \in [q_c]$, $i \in [q_p]$, and $\alpha \in [0..w]$ such that $\widehat{U}^j_\alpha = U^i_\alpha$, then (M^j, C^j_α) is removed from τ_c^{new} and $(U_\beta, V_\beta) = (\widehat{U}^j_\beta, \widehat{V}^j_\beta)$ is added to $\tau_\beta^{\mathsf{new}}$, for all $\beta \in [0..w]$ with $\beta \neq \alpha$.

Given q_c constructions queries and q_p primitive queries to each of the permutations $\pi(\cdot \parallel \langle i \rangle_d)$, for $i \in [0..w]$ in the original transcript, the numbers of queries moved from τ_c into the primitive partial transcripts τ_i is denoted by s_i. The number of queries in the new construction transcript is denoted by $q' = q_c - \sum_{i=0}^{w} s_i$. Moreover, we define $q_i = q_p + s_i$, for all $0 \leq i \leq w$. In the following, for a given transcript τ_0^{new} of q' elements, it remains to count the number of permutations π that are compatible with the transcript. The set of occurred (i.e., prohibited) outputs of $\pi^{\pm}(\cdot \parallel \langle \iota \rangle_d)$ are denoted by V_ι^{out}, for $0 \leq \iota \leq w$. For $\alpha = 0, \dots, q'-1$, let

$$\lambda_{\alpha+1} \overset{\text{def}}{=} \left|\left\{(V_0^1, \dots, V_0^{\alpha+1}, \dots, V_w^1, \dots, V_w^{\alpha+1})\right\}\right| \tag{10}$$

be the number of solutions that satisfy

(1) $\left\{(V_0^1, \dots, V_0^{\alpha}, \dots, V_w^1, \dots, V_w^{\alpha})\right\}$ satisfy the conditions recursively,

(2) It holds that

$$\begin{aligned} V_0^{\alpha+1} \oplus V_1^{\alpha+1} &= C_1^{\alpha+1} \oplus K_0 \oplus K_1 \\ &\vdots \\ V_0^{\alpha+1} \oplus V_w^{\alpha+1} &= C_w^{\alpha+1} \oplus K_0 \oplus K_w. \end{aligned} \tag{11}$$

(3.0) It holds that $V_0^{\alpha+1} \notin \{V_0^1, \dots, V_0^\alpha\} \cup V_0^{\mathsf{out}} \cup \dots \cup V_w^{\mathsf{out}}$.

– …

(3.w) It holds that $V_w^{\alpha+1} \notin \{V_w^1, \dots, V_w^\alpha\} \cup V_0^{\mathsf{out}} \cup \dots \cup V_w^{\mathsf{out}}$.

Then, the goal is to define a recursive expression for $\lambda_{\alpha+1}$ from λ_α such that a lower bound can be found for the expression $\lambda_{\alpha+1}/\lambda_\alpha$. It holds that

$$|\mathsf{Comp}_{\mathrm{real}}(\tau)| = \lambda_{q'} \cdot \left(2^n - \left(\sum_{i=0}^{w} q_i + (w+1)q'\right)\right)!$$

We obtain

$$\frac{\Pr[\Theta_{\text{real}} = \tau]}{\Pr[\Theta_{\text{ideal}} = \tau]} = \frac{\lambda_{q'} \cdot (2^n - (\sum_{i=0}^{w} q_i + (w+1)q'))!}{(2^n - (w+1)q_p)!} \cdot (2^n)^{w \cdot q_c} \,. \qquad (12)$$

Let $\mathcal{B}_{(1,2)}$ denote the set of solutions that comply with only Conditions (1) and (2), without considering Conditions (3.0) through (3.w). Moreover, let $\mathcal{B}_{(3.\iota:i)}$ denote the set of solutions compatible with Conditions (1) and (2), but not with (3.$\iota : i$), for $i = 1, \dots, \alpha + \sum_{k=0}^{w} |V_k^{\mathsf{out}}|$. From inclusion-exclusion, it follows that

$$\begin{aligned}
\lambda_{\alpha+1} &= \left|\mathcal{B}_{(1,2)}\right| - \left| \bigcup_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+\cdots+|V_w^{\mathsf{out}}} \mathcal{B}_{(3.0:i)} \right| \cup \cdots \cup \left| \bigcup_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+\cdots+|V_w^{\mathsf{out}}} |\mathcal{B}_{(3.w:i)}| \right| \\
&\geq \left|\mathcal{B}_{(1,2)}\right| - \left| \sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+\cdots+|V_w^{\mathsf{out}}} |\mathcal{B}_{(3.0:i)}| \right| - \cdots - \left| \sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+\cdots+|V_w^{\mathsf{out}}} |\mathcal{B}_{(3.w:i)}| \right| \\
&\quad + \sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|} \sum_{i'=1}^{\alpha+|V_1^{\mathsf{out}}|} \left|\mathcal{B}_{(3.0:i)} \cap \mathcal{B}_{(3.1:i')}\right| + \cdots \\
&\quad + \sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+\cdots+|V_w^{\mathsf{out}}|} \sum_{i'=1}^{\alpha+|V_0^{\mathsf{out}}|+\cdots+|V_w^{\mathsf{out}}|} \left|\mathcal{B}_{(3.(w-1):i)} \cap \mathcal{B}_{(3.w:i')}\right| \\
&\geq 2^n \cdot \lambda_\alpha - \sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+\cdots+|V_w^{\mathsf{out}}|} \lambda_\alpha - \cdots - \sum_{i=1}^{\alpha+|V_0^{\mathsf{out}}|+\cdots+|V_w^{\mathsf{out}}|} \lambda_\alpha \,.
\end{aligned}$$

So, it follows that

$$\lambda_{\alpha+1} \geq 2^n \cdot \lambda_\alpha - (\alpha + q_p + s_0) \cdot \lambda_\alpha - \ldots - (\alpha + q_p + s_w) \cdot \lambda_\alpha.$$

Therefore,

$$\begin{aligned}
\frac{\lambda_{\alpha+1}}{\lambda_\alpha} &\geq 2^n - (w+1)\alpha - (w+1)q_p - (w+1)\sum_{i=0}^{w} s_i \\
&= 2^n - (w+1)\alpha - (w+1)\sum_{i=0}^{w} q_i
\end{aligned}$$

with $\lambda_0 = 1$. It follows from Equation (12) that

$$\begin{aligned}
(12) &= \prod_{j=0}^{w \cdot s - 1} \frac{2^n}{2^n - (w+1)q_p - j} \cdot \prod_{j=0}^{q'-1} \frac{\lambda_{\alpha+1}}{\lambda_\alpha} \cdot \frac{(2^n)^w}{\prod_{i=0}^{w}(2^n - \sum_{k=0}^{w} q_k - iq' - j)} \\
&\geq \prod_{j=0}^{q'-1} \frac{(2^n - (w+1)j - (w+1)\sum_{i=0}^{w} q_i)(2^n)^w}{\prod_{i=0}^{w}(2^n - \sum_{k=0}^{w} q_k - iq' - j)} \,.
\end{aligned}$$

We use $q_{\mathsf{sum}} =^{\text{def}} \sum_{k=0}^{w} q_k$. Then

$$\prod_{j=0}^{q'-1} \frac{(2^n - (w+1)(q' + q_{\mathsf{sum}})(2^n)^w}{\prod_{i=0}^{w}(2^n - (q_{\mathsf{sum}} + q'))} \geq \prod_{j=0}^{q'-1} \frac{(2^n - (w+1)(q' + q_{\mathsf{sum}})(2^n)^w}{(2^n - (q_{\mathsf{sum}} + q'))^{w+1}}. \tag{13}$$

It holds that

$$\begin{aligned}
&\frac{1}{(2^n - (q_{\mathsf{sum}} + q'))^{w+1}} \\
&= \frac{1}{(2^n)^{w+1} - \binom{w+1}{1}(2^n)^w(q_{\mathsf{sum}} + q') + \binom{w+1}{2}(2^n)^{w-1}(q_{\mathsf{sum}} + q')^2 - \ldots} \\
&\geq \frac{1}{(2^n)^{w+1} - \binom{w+1}{1}(2^n)^w(q_{\mathsf{sum}} + q') + \binom{w+1}{2}(2^n)^{w-1}(q_{\mathsf{sum}} + q')^2}.
\end{aligned}$$

For the sake of format, we define a helping variable

$$\begin{aligned}
z \overset{\text{def}}{=} (2^n)^{w+1} - (2^n)^w(w+1)(q' + q_{\mathsf{sum}}) + \binom{w+1}{2}(2^n)^{w-1}(q_{\mathsf{sum}} + q')^2 - \\
\binom{w+1}{2}(2^n)^{w-1}(q_{\mathsf{sum}} + q')^2 \,.
\end{aligned}$$

It follows that

$$\begin{aligned}
(13) &\geq \left(\frac{z}{(2^n)^{w+1} - \binom{w+1}{1}(2^n)^w(q_{\mathsf{sum}} + q') + \binom{w+1}{2}(2^n)^{w-1}(q_{\mathsf{sum}} + q')^2} \right)^{q'} \\
&\geq \left(1 - \frac{\binom{w+1}{2}(2^n)^{w-1}(q_{\mathsf{sum}} + q')^2}{(2^n)^{w+1} - \binom{w+1}{1}(2^n)^w(q_{\mathsf{sum}} + q') + \binom{w+1}{2}(2^n)^{w-1}(q_{\mathsf{sum}} + q')^2} \right)^{q'} \\
&\geq 1 - \frac{\binom{w+1}{2}(2^n)^{w-1}(q_{\mathsf{sum}} + q')^2 \cdot q'}{(2^n)^{w+1} - \binom{w+1}{1}(2^n)^w(q_{\mathsf{sum}} + q') + \binom{w+1}{2}(2^n)^{w-1}(q_{\mathsf{sum}} + q')^2} \\
&\geq 1 - \frac{2\binom{w+1}{2}(q_{\mathsf{sum}} + q')^2 \cdot q'}{(2^n)^2} \\
&\geq 1 - \frac{(w+1)^2(q'^3 + q'^2 q_{\mathsf{sum}} + q' q_{\mathsf{sum}}^2)}{2^{2n}}.
\end{aligned}$$

Since $q' + q_{\mathsf{sum}} = s \cdot w + q' + (w+1)q_p$ and $s \leq q_p$, it follows that $q' + q_{\mathsf{sum}} \leq q_c + 2wq_p$:

$$\begin{aligned}
&1 - \frac{(w+1)^2(q_c^3 + q_c^2(q_c + 2wq_p) + q_c(q_c + 2wq_p)^2}{2^{2n}} \\
&\geq 1 - \frac{(w+1)^2(q_c^3 + q_c^3 + 2w \cdot q_c^2 q_p + q_c^3 + 4w \cdot q_c^2 q_p + 4w^2 \cdot q_c q_p^2)}{2^{2n}} \\
&\geq 1 - \frac{3(w+1)^2 q_c^3 + 6(w+1)^3 q_c^2 q_p + 4(w+1)^4 q_c q_p^2}{2^{2n}}. \quad \square
\end{aligned} \tag{14}$$