CENCPP* – Beyond-birthday-secure Encryption from Public Permutations

Arghya Bhattacharjee[1], Avijit Dutta[2], Eik List[3] and Mridul Nandi[1]

[1] Indian Statistical Institute, Kolkata, India
bhattacharjeearghya29(at)gmail.com,
mridul.nandi(at)gmail.com,
[2] Institute of Advancing Intelligence, TCG-CREST, Kolkata, 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 permutations. 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., [GJMN16]), authentication (e.g. [MMH+14]), or authenticated encryption (e.g. [AJN14, BDP+16, DEMS16]).

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.
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 a tweak as an additional public input, has allowed the construction of modes with 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 is usually tolerated, e.g. in [BDH+17, GJMN16]. This lack of security and efficiency has been compensated by using permutations with larger state sizes. Moreover, well-known generic attacks render it difficult to succumb the birthday bound when the primitive is public due, e.g. [DDKS13, DKS12]. Many approaches tried to increase the security by multiple calls to the primitive, e.g., with multi-round Even-Mansour constructions [CLL+14, CLM19, CS14, CS15]. Cogliati et al. [CDK+18] studied the security of (wide) TBCs based on SPNs with public permutations and linear or nonlinear tweaking and mixing layers. They showed $2n/3$-bit security for two rounds with nonlinear mixing layers.
Though, permutation-based modes do not have to be limited in general. Often, it is argued that the mere size of the underlying permutation renders birthday attacks infeasible – a valid and pragmatic argument. Equally pragmatically, yet, the state size of current permutations poses considerable costs either to implementation size, area, or performance. Thus, efficient permutation-based modes with higher security seem attractive, be it with some restrictions such as the need for multiple keys.
At CRYPTO'19, Chen et al. [CLM19] proposed two permutation-based PRFs, the Sum-of-Even-Mansour constructions (SoEM) and Sum-of-Key-alternating-Ciphers (SoKAC), with proofs for up to $O(2^{2n/3})$ queries. The single primitive variant was revisited by [Nan20] and [CNTY20]. Moreover, [CNTY20] proposed PDM-MAC; [DN20] introduced nEHtM$_p$, both $2n/3$-bit-secure PRFs with n-bit outputs from public permutations. Still, their constructions map only fixed-length inputs to fixed-length outputs, which left the question of designing a variable-length encryption scheme with similar security open.

This work proposes 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]. This represents a trade-off, where w can be chosen to be below the usual number of round keys e.g. for the AES [NIS01] or Deoxys-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 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 DS-SoEM and DS-CENCPP*$[w]$, and show their security. We show that two independent keys are sufficient and necessary for our security guarantees by providing distinguishers for all constructions in $O(2^{n/2})$ if single keys or simpler key-scheduling approaches would be taken. Moreover, we describe distinguishers in $O(2^{2n/3})$ queries to note that the security is effectively tight except the logarithmic factor in w. We compare our proposals with public-permutation-based beyond-birthday-secure PRFs from the literature in Figure 1.

Construction	#Prim.	#Keys	IF	Nonce	Rate	Bits In	Bits Out	Sec.
PDM-MAC [CNTY20]	1	1	–	–	1/2	n	n	$\frac{2n}{3}$
SoKAC22 [CLM19]	2	2	•	–	1/2	n	n	$\frac{2n}{3}$
SoEM22 [CLM19]	2	2	•	–	1/2	n	n	$\frac{2n}{3}$
DS-SoEM [Sect. 6]	1	2	•	–	$\frac{n-d}{2n}$	$n-d$	n	$\frac{2n}{3}$
nEHtM$_p$ [DN20]	2	2	•	•	1/2	$*$	n	$\frac{2n}{3}$
CENCPP* [Sect. 3]	w+1	2	•	•	$\frac{w}{w+1}$	$*$	$*$	$\frac{2n}{3\log(w^2)}$
DS-CENCPP* [Sect. 6]	1	2	•	•	$\frac{w(n-d)}{(w+1)n}$	$*$	$*$	$\frac{2n}{3\log(w^4)}$

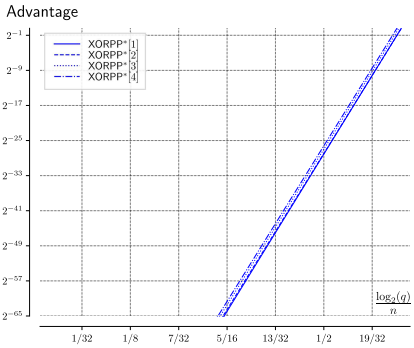


Figure 1: Left: Comparison with existing PRFs from public permutations with beyond-birthday-bound security. Prim. = primitives, IF = inverse-free, n = state size, w = word parameter, d = domain size, sec. = security, •/– = yes/no. **Right:** Security of XORPP$^*[w]$ for varying w.

The remainder is structured as follows: Section 2 recalls preliminaries before Section 3 defines CENCPP*. We employ two different keys for security and show that it is necessary to combine the keys for most primitive calls. We show that 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 DS-SoEM and DS-CENCPP*. We provide a design rationale and distinguishers on weaker variants in Section 7. We analyze the security of DS-SoEM and DS-CENCPP* in Section 8 and 9 respectively. Section 10 concludes.

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. We will use $\mathbb{F}_2^n$ and $\{0,1\}^n$ interchangeably in this paper. $X \,\|\, Y$ 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 $X = X_1 \,\|\, \cdots \,\|\, X_j$, $|X_i| = x$ for $1 \le i \le j-1$, and $|X_j| \le x$. For positive integer m, we use $\mathcal{X}^{\le m} =^{\text{def}} \bigcup_{i=0}^{m} \mathcal{X}^i$. 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 $x \le n$, 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 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 $\alpha \cdot \mathcal{V} =^{\text{def}} \{\alpha \cdot V : V \in \mathcal{V}\}$. We write $\alpha\mathcal{V}$ or $\alpha \cdot \mathcal{V}$ when the operation is clear from the context. 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 $\mathbf{D}$ is an efficient Turing machine that interacts with a set of oracles that are black boxes to $\mathbf{D}$. We write $\Delta_{\mathbf{D}}\left(\mathcal{O}^1; \mathcal{O}^2\right)$ for the advantage of $\mathbf{D}$ to distinguish between $\mathcal{O}^1$ and $\mathcal{O}^2$. All probabilities are defined over the random coins of the oracles

and those of $\mathbf{D}$ if any. $\mathbf{Adv}_F^X(q,\sigma) \stackrel{\text{def}}{=} \max_{\mathbf{D}}\{\mathbf{Adv}_F^X(\mathbf{D})\}$ denotes the maximal advantage over all X-distinguishers $\mathbf{D}$ on F that ask $\leq q$ queries of $\leq \sigma$ blocks in total to its oracles. W.l.o.g., we assume that $\mathbf{D}$ never asks queries to which it already knows the answer.
We consider information-theoretic distinguishers $\mathbf{D}$, whose resources are bounded only in terms of their maximal numbers of queries and blocks that they can ask to their available oracles. One can derive computation-theoretic counterparts in a 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. For analysis constructions 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.

PRF security refers to the maximal advantage of distinguishing the outputs of a scheme from random bits of the expected length. For primitives and schemes in general, we will often use the set $\mathcal{K} = \mathbb{F}_2^n$ 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. 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 $\mathbf{D}$ is defined as $\mathbf{Adv}_{F_K}^{\mathsf{PRF}}(\mathbf{D}) \stackrel{\text{def}}{=} \Delta_{\mathbf{D}}(F_K; \rho)$.

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 $\mathbf{D}$ is defined as $\mathbf{Adv}_{\Pi_K}^{\mathsf{nE}}(\mathbf{D}) \stackrel{\text{def}}{=} \Delta_{\mathbf{D}}(\mathcal{E}_K; \rho)$.

In the ideal-permutation model, the distinguisher has one or multiple additional oracles $\pi^{\pm}$ that provides access to the public permutation π in for- and backward directions. This work studies the security notions such as PRF and nE security in the ideal-permutation model. We write $\Pi[\pi]$ and $\mathcal{E}[\pi]$, $\mathcal{D}[\pi]$, etc. to indicate that Π is based on a primitive π.

The H-coefficient technique is a proof method by Patarin [Pat08, Pat10] that was modernized by Chen and Steinberger [CS14]. A distinguisher $\mathbf{D}$ interacts with 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 its interaction are collected in a transcript 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 attainable if $\mathbf{D}$ 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 GoodT and BadT and bound the distinguishing advantage as:

Lemma 1 ([Pat08])**.** Assume, there exist $\epsilon_1, \epsilon_2 \geq 0$ s. t. for any transcript $\tau \in \text{GoodT}$, it holds $\frac{\Pr[\Theta_{\text{real}}=\tau]}{\Pr[\Theta_{\text{ideal}}=\tau]} \geq 1 - \epsilon_1$ and $\Pr[\Theta_{\text{ideal}} \in \text{BadT}] \leq \epsilon_2$. Then, for all distinguishers $\mathbf{D}$, it holds that $\Delta_{\mathbf{D}}(\mathcal{O}_{\text{real}}; \mathcal{O}_{\text{ideal}}) \leq \epsilon_1 + \epsilon_2$.

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

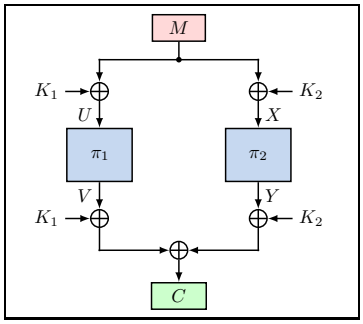


Figure 3: The construction SoEM22 by Chen et al. [CLM19].

Lemma 2 (Sum-capture Lemma [CLL+14])**.** Let $n, q \in \mathbb{N}$ s. t. $9n \leq q \leq 2^{n-1}$. Let $\mathcal{T} = \{T^1, \ldots, T^q\} \subseteq \mathbb{F}_2^n$ s. t. 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,

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

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

Lemma 3. Let $\mathbf{A}_{2\times 2} = (a_{ij}) \in \{0,1\}^n$ be a non-singular matrix. For any $b_1, b_2 \in \{0,1\}^n$

$$\Pr\left[K_0, K_1 \twoheadleftarrow \{0,1\}^n : \mathbf{A} \cdot (K_0, K_1)^\top = (b_1, b_2)^\top\right] = 2^{-2n}\,.$$

Proof. Since $\mathbf{A}$ is non-singular, $\mathbf{A}^{-1}$ exists. Therefore, $K_0 = {\mathbf{A}^{-1}}_1 \cdot [b_1\ b_2]^\top$ and $K_1 = {\mathbf{A}^{-1}}_2 \cdot [b_1\ b_2]^\top$, where ${\mathbf{A}^{-1}}_1$ and ${\mathbf{A}^{-1}}_2$ are the first and second row of $\mathbf{A}^{-1}$ respectively. Since, K_0, K_1 are uniform random variables over $\{0,1\}^n$, the result follows. □

3 The CENCPP* Mode

This section defines a generic CENC construction that we call CENCPP*. Standing on the shoulders of existing constructions, we start with the necessary details of SoEM and CENC.

3.1 SoEM

At CRYPTO'19, Chen et al. [CLM19] proposed 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; the latter is 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 3 illustrates SoEM22, which will be relevant in this work. Both modes need two calls to the independent permutations. Moreover, SoEM demanded two independent keys. Chen et al. studied 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,

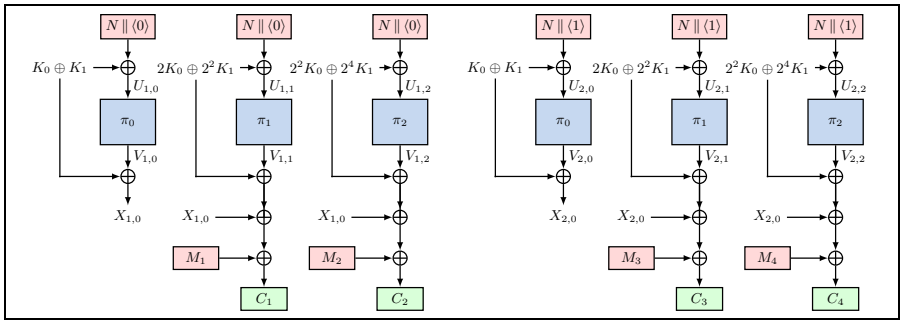


Figure 4: Encryption of a four-block message $M = (M_1, \ldots, M_4)$ with $\mathsf{CENCPP}^*[(\pi_0, \pi_1, \pi_2), 2]_{K_0,K_1}$. The final chunk is truncated if its length is less than $2n$ bits. N is a nonce, K_0 and K_1 are independent secret keys and π_0, π_1, and π_2 independent permutations.

splits each sequence of w 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 input bits 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 $S_1 \| \ldots \| S_w$ as

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

Thus, it makes $w + 1$ block-cipher calls with pairwise distinct inputs, where $E_K(X \| \langle s \rangle_\mu)$ 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 $N \| \langle i \rangle_\mu$, $N \| \langle i + 1 \rangle_\mu$, ...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. $(M_1 \| \ldots \| M_m) \xleftarrow{n} M$. Let $\ell = \lceil m/w \rceil$ denote the number of chunks. It must hold that $\ell \cdot (w + 1) < 2^\mu$. Then

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

3.3 CENCPP*

In the following, we adapt CENC to the public-permutation setting. Let $\mathbf{A} = (a_{ij})$ be a $(w + 1) \times 2$ dimensional matrix such that each of its element a_{ij} are n-bit binary strings. Let $\pi_0, \ldots, \pi_w \in \mathsf{Perm}(\mathbb{F}_2^n)$ be permutations, and let $K_0, K_1 \in \mathbb{F}_2^n$ be independent secret keys. We define $\boldsymbol{\pi} \stackrel{\text{def}}{=} (\pi_0, \ldots, \pi_w)$ 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 $(a_{i,0} \cdot K_0) \oplus (a_{i,1} \cdot K_1)$ to generate the i-th round key K'_i, for all $i \in [0..w]$. In matrix notation, we write this as follows:

$$\mathbf{A} \cdot \mathbf{K} = \begin{bmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \\ \ldots & \ldots \\ a_{w,0} & a_{w,1} \end{bmatrix} \cdot \begin{bmatrix} K_0 \\ K_1 \end{bmatrix} = \begin{bmatrix} K'_0 \\ K'_1 \\ \vdots \\ K'_w \end{bmatrix}.$$

Algorithm 1 Definition of CENCPP*.

```
101: function CENCPP*[π, w, A].E_K(N, M)
102:   (M_1, ..., M_m) ←n M
103:   ℓ ← ⌈m/w⌉
104:   for i ← 0..ℓ − 1 do
105:     j ← i · w
106:     (S_{j+1} ‖ ··· ‖ S_{j+w})
107:         ← XORPP*[π, w, A]_K(N ‖ ⟨i⟩_μ)
108:     for k ← j + 1..j + w do
109:       C_k ← msb_{|M_k|}(S_k) ⊕ M_k
110:   return (C_1 ‖ ··· ‖ C_m)

201: function CENCPP*[π, w, A].D_K(N, C)
202:   return CENCPP*[π, w, A].E_K(N, C)

301: function XORPP*[π, w, A]_K(M)
302:   (K_0, K_1) ← K
303:   (π_0, ..., π_w) ← π
304:   U_0 ← M ⊕ (a_{0,0}K_0 ⊕ a_{0,1}K_1)
305:   X_0 ← π_0(U_0) ⊕ (K_0 ⊕ K_1)
306:   for j ← 1..w do
307:     L_j ← (a_{j,0} · K_0) ⊕ (a_{j,1} · K_1)
308:     U_j ← M ⊕ L_j
309:     X_j ← π_j(U_j) ⊕ L_j
310:     C_j ← X_j ⊕ X_0
311:   return (C_1 ‖ ··· ‖ C_w)
```

We call $\mathbf{A}$ the *key-scheduling matrix*. We adapt XORP to XORPP* to note that it is based on the XOR of **public** permutations. For a key-scheduling matrix $\mathbf{A}$ of dimension $(w+1)\times 2$, we define XORPP* $[\boldsymbol{\pi}, w, \mathbf{A}] : (\mathbb{F}_2^n)^2 \times \mathbb{F}_2^n \to (\mathbb{F}_2^n)^w$, instantiated with $w+1$ permutations $\pi_0, \ldots, \pi_w$, a key space $(\mathbb{F}_2^n)^2$ and the key-scheduling matrix $\mathbf{A}$. We write XORPP* as short for XORPP* $[\boldsymbol{\pi}, w, \mathbf{A}]$ when w, key-scheduling matrix $\mathbf{A}$ and the permutations $\boldsymbol{\pi}$ are clear from the context. Given that the permutations are independent, CENCPP* uses the same input $(N \,\|\, \langle i \rangle_\mu)$ for each permutation in one call of XORPP*. We define encryption and decryption 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 CENCPP* seems very efficient. Instantiating CENC with a two-round Even-Mansour construction could be a generic approach that can provide roughly 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 AES-PRF, Mennink and Neves increased the performance of their construction [MN17b] by instantiating it with five-round AES. However, its security margin is thin [DIS+18], so that improved cryptanalysis could endanger it 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 [DDNY18] that would also possess a security bound of $O(2^{2n/3})$ but limited the input space $2n/3$ bits. SUMPIP could retain beyond-birthday-bound security 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 en- and decryption of the primitive, which is less practical than a construction that only needs a single direction. Plus, for CENCPP*, we are unaware of 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.

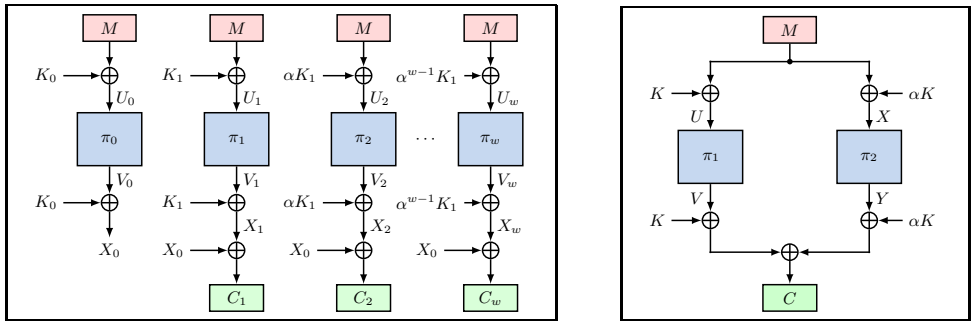


Figure 5: Example of using a weak key schedule for XORPP* (left) and SoEM′ (right).

4 Birthday-bound Distinguisher on CENCPP*

To derive the i-th round key K'_i of CENCPP*, we have $K'_i = (a_{i,0} \cdot K_0) \oplus (a_{i,1} \cdot K_1)$ for all $i \in [0..w]$, where $\mathbf{A} = (a_{ij}) \in \{0,1\}^n$ is the key-scheduling matrix of dimension $(w+1) \times 2$ and K_0, K_1 are two independent n-bit keys. Using SoEM as a base, it is tempting to use a key scheduling of K_0, K_1, αK_1, $\alpha^2 K_1$, ..., which omits the addition of K_0 for all subsequent permutation calls. In matrix form, this key scheduling would produce

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

While the latter appears much simpler, after transposing its matrix form to $w+1$ rows, it contains dependent rows. Let those two rows be $\mathbf{A}_i$ and $\mathbf{A}_j$ in the key-scheduling matrix $\mathbf{A}$ such that they are linearly dependent, i.e., $\mathbf{A}_i = \alpha \mathbf{A}_j$ for some non-zero $\alpha \in \{0,1\}^n$. Then, we have $K'_i = \alpha K'_j$ for some $\alpha \in \{0,1\}^n \setminus \{0^n\}$. We use the idea of canceling the two outputs that use dependent keys and reduce the distinguishing problem to that for single-key SoEM. Since the steps are not intuitive, we illustrate the birthday-bound distinguisher of CENCPP* in the following. First, we show that we can reduce the security of CENCPP* to the security of SoEM with the key usage of $(K'_i, \alpha K'_i)$ for some non-zero $\alpha \in \{0,1\}^n$ when $\mathbf{A}_i$ and $\mathbf{A}_j$ rows of $\mathbf{A}$ are linearly dependent. We denote this variant of SoEM as $\mathsf{SoEM}' \stackrel{\text{def}}{=} \mathsf{SoEM}[\pi_i, \pi_j]_{K'_i, \alpha K'_i}$.

4.1 Reduction to SoEM′

Suppose, $\mathbf{D}$ is an information-theoretic distinguisher on SoEM′ and $\tau = \{K\} \cup \tau_p \cup \tau_c$ is a transcript, consisting of the key, the primitive-query transcript τ_p with q_p primitive queries and their corresponding responses (U^i, V^i) to π_1 and (X^k, Y^k) to π_2 each, as well as the construction-query transcript τ_c with q_c construction queries and their corresponding responses (M^j, C^j). After the interaction, $\mathbf{D}$ is given τ, including the key $K \twoheadleftarrow \mathbb{F}_{2^n}$, and sees $C = W \oplus Z$ where $W =^{\text{def}} \pi_1(M \oplus K) \oplus K$ and $Z =^{\text{def}} \pi_2(M \oplus (\alpha \cdot K)) \oplus (\alpha \cdot K)$. In comparison, a distinguisher $\mathbf{D}'$ on CENCPP* $[\pi_0, \pi_i, \pi_j]_{K_0, K_1}$ with key schedule as above can compute $C_i \oplus C_j = (X_i \oplus X_0) \oplus (X_j \oplus X_0) = W \oplus Z = C$. Thus,

$$\mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{CENCPP}^*}(\mathbf{D}') \geq \mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{SoEM}'}(\mathbf{D}) ,$$

where $\mathbf{D}$ and $\mathbf{D}'$ ask the same number of construction queries q_c and primitive queries q_p to each of the primitives. Note that the distinguisher $\mathbf{D}'$ knows the values of i and j from the knowledge of the key-scheduling algorithm.

4.2 Birthday-bound Attack on SoEM′

Let $\mathcal{U}$ and $\mathcal{V}$ be two subspaces of $\mathbb{F}_{2^n}$. Then, for every $\alpha \in \mathbb{F}_{2^n}$, $\mathcal{U}+\mathcal{V} =^{\text{def}} \{u+v | u \in \mathcal{U}, v \in \mathcal{V}\}$ and $\alpha \cdot \mathcal{V} =^{\text{def}} \{\alpha \cdot v | v \in \mathcal{V}\}$ are also subspaces. We write $\mathbf{0}$ and $\mathbf{1}$ for the neutral elements of addition and multiplication, respectively. If $\{x_1, x_2, \cdots, x_{n/2}\}$ is a basis of $\mathcal{V}$, then $\{\alpha \cdot x_1, \alpha \cdot x_2, \cdots, \alpha \cdot x_{n/2}\}$ is also a basis of $\alpha \cdot \mathcal{V}$, where $\alpha \neq \mathbf{0}$.

Fact 1. Let $\mathcal{U}$ and $\mathcal{V}$ are two subspaces of $\mathbb{F}_{2^n}$. If their intersection contains only the zero element $\mathcal{U} \cap \mathcal{V} = \{\mathbf{0}\}$, we say that $\mathcal{U}$ and $\mathcal{V}$ have zero intersection. If both have zero intersection, it holds that $\dim(\mathcal{U}+\mathcal{V}) = \dim(\mathcal{U}) + \dim(\mathcal{V})$. Equivalently, one can say that the basis elements of $\mathcal{U}$ and $\mathcal{V}$ are linearly independent.

Theorem 1. Let $\alpha \notin \{\mathbf{0}, \mathbf{1}\}$. For every $1 \leq i \leq n/2$, there exists a subspace $\mathcal{V} \subseteq \mathbb{F}_{2^n}$ with $\dim(\mathcal{V}) = i$ such that $\mathcal{V}$ and $\alpha \cdot \mathcal{V}$ have zero intersection. In particular, there is a subspace $\mathcal{V}$ of dimension $n/2$ such that $\mathcal{V} + \alpha \cdot \mathcal{V} = \mathbb{F}_{2^n}$.

Proof. We prove Theorem 1 by induction on i. For $i = 1$, the statement is obvious by choosing non-zero x_1. For $1 \leq i < n/2$, suppose, we have picked $x_1, x_2, \cdots, x_i$ such that all elements from $\{x_1, x_2, \cdots, x_i, \alpha \cdot x_1, \alpha \cdot x_2, \cdots \alpha \cdot x_i\}$ are linearly independent. Let $\mathcal{S}_i =^{\text{def}} \mathbf{span}(\{x_1, x_2, \cdots, x_i, \alpha \cdot x_1, \alpha \cdot x_2, \cdots, \alpha \cdot x_i\})$, i.e., its span. Moreover, we define $\mathcal{T}_i$ as short form of $\mathcal{T}_i =^{\text{def}} \mathcal{S}_i \cup \big(\alpha^{-1} \cdot \mathcal{S}_i\big) \cup \big((1+\alpha)^{-1} \cdot \mathcal{S}_i\big)$. It holds that $|\mathcal{T}_i| \leq 3 \cdot 2^{n-2} < 2^n$. When we choose a new element $x_{i+1} \notin \mathcal{T}_i$, it follows from the definition of $\mathcal{T}_i$ that x_{i+1}, $\alpha \cdot x_{i+1}$ and $(1+\alpha) \cdot x_{i+1}$ are not in $\mathcal{S}_i$. Hence, $\{x_1, x_2, \cdots, x_{i+1}, \alpha \cdot x_1, \alpha \cdot x_2, \cdots, \alpha \cdot x_{i+1}\}$ are linearly independent, which concludes the proof. Note that such a basis can be constructed efficiently element by element. □

Distinguisher on SoEM′. Next, we demonstrate a distinguisher on SoEM′. Given the observation above, we can first construct a vector space $\mathcal{X}$ of dimension $n/2$ such that $\mathcal{X} + (1+\alpha) \cdot \mathcal{X} = \mathbb{F}_{2^n}$. Let $\mathcal{M} = (1+\alpha)^{-1} \cdot \mathcal{X}$. So, $\mathcal{M} + \mathcal{X} = \mathbb{F}_{2^n}$ and hence there exists $X \in \mathcal{X}$ and $M \in \mathcal{M}$ with $M + X = \alpha \cdot K$. Let $\mathcal{U} = \alpha^{-1} \cdot \mathcal{X}$. Then

$$\mathcal{U} = \alpha^{-1} \cdot (1+\alpha) \cdot \mathcal{M} = (1+\alpha^{-1}) \cdot M\,.$$

Thus, $M + K = \alpha^{-1} \cdot X + (1+\alpha^{-1}) \cdot M \in \mathcal{U}$ and there exists $M \in \mathcal{M}, U \in \mathcal{U}$, and $X \in \mathcal{X}$ such that $M \oplus U = K$ and $M \oplus X = \alpha K$.
Let $\pi_1(U) = V$ and $\pi_2(X) = Y$. Then, $C = \mathsf{SoEM}'(M) = (1 \oplus \alpha) \cdot K \oplus V \oplus Y$. We use shorthand notations $V^{\oplus c}$, $Y^{\oplus c}$ and $C^{\oplus c}$ to denote $\pi_1(U \oplus c)$, $\pi_2(X \oplus c)$ and $\mathsf{SoEM}'(M \oplus c)$ respectively for some non-zero $c \in \{0,1\}^n$. It is easy to see that for any c, it holds that

$$C^{\oplus c} = (1 \oplus \alpha) \cdot K \oplus V^{\oplus c} \oplus Y^{\oplus c}$$

and hence $C \oplus C^{\oplus c} = (V \oplus V^{\oplus c}) \oplus (Y \oplus Y^{\oplus c})$. We use this observation to complete our attack. Suppose that c and d are two distinct constants outside of $\mathcal{U}$, $\mathcal{X}$, and $\mathcal{M}$. Then, the distinguisher can proceed as follows:

1. It queries all values $U_i \in \mathcal{U}$, $U_i \oplus c$ and $U_i \oplus d$ to its primitive oracle π_1, and stores them together with the corresponding responses V_i, $V_i^{\oplus c}$ and $V_i^{\oplus d}$.
2. Similarly, it queries all values $X_i \in \mathcal{X}$, $X_i \oplus c$ and $X_i \oplus d$ to its primitive oracle π_2, and stores them together with the corresponding responses Y_i, $Y_i^{\oplus c}$ and $Y_i^{\oplus d}$.
3. Moreover, it queries all values $M_i \in \mathcal{M}$, $M_i \oplus c$ and $M_i \oplus d$ to its construction oracle, and stores them together with the corresponding responses C_i, $C_i^{\oplus c}$ and $C_i^{\oplus d}$.
4. After making all queries as described above, it looks for triple (i, j, k) such that the following two equalities hold:

4.1 $C_i \oplus C_i^{\oplus c} = (V_j \oplus V_j^{\oplus c}) \oplus (Y_k \oplus Y_k^{\oplus c})$.
4.2 $C_i \oplus C_i^{\oplus d} = (V_j \oplus V_j^{\oplus d}) \oplus (Y_k \oplus Y_k^{\oplus d})$.

5. If there exists such triple (i, j, k), it outputs real, and random otherwise.

5 Security Analysis of CENCPP*

This section studies the nE security of CENCPP*. Prior, we briefly revisit that 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] adapted this bound 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 with security for up to $2^n/w$ calls to E_K. The proof sketch by [IMV16] reduced the security of CENC to the proof of the sum of two permutations. At that time, the latter analysis relied on recursive arguments of Patarin's Mirror Theory that were subject to controversies. The work by Bhattacharya and Nandi [BN18] proved similar security for the generalized sum of permutations and CENC using the χ^2 method [DHT17].

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, $K_0, K_1 \twoheadleftarrow \mathcal{K}$ be independent secret keys and $\mathbf{A}$ be the $(w+1) \times 2$ dimensional key-scheduling matrix such that each entry is an n-bit binary string. 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* $[\boldsymbol{\pi}, w, \mathbf{A}]$ and (2) the PRF security of CENCPP*$[\boldsymbol{\pi}, w, \mathbf{A}]$. For the simplicity of the notation, we write XORPP* $[\boldsymbol{\pi}, w, \mathbf{A}]$ as XORPP* and CENCPP*$[\boldsymbol{\pi}, w, \mathbf{A}]$ as CENCPP*.

Theorem 2. It holds that $\mathbf{Adv}^{\mathsf{nE}}_{\mathsf{CENCPP}^*}(q_p, q_c, \sigma) \leq \mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{XORPP}^*}\left(q_p, \frac{m}{w}q_c, \sigma\right)$.

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*$[\boldsymbol{\pi}, w, \mathbf{A}]_{\mathbf{K}}$ consists of the application of ℓ instances of XORPP* $[\boldsymbol{\pi}, w, \mathbf{A}]_{\mathbf{K}}$. We can replace XORPP* $[\boldsymbol{\pi}, w]_{\mathbf{K}}$ by a random function ρ at the cost of

$$\mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{XORPP}^*}\left(q_p, \frac{m}{w}q_c, \sigma\right).$$

Since the resulting construction is indistinguishable from random bits, Theorem 2 follows.

Theorem 3. Let $\mathbf{A}$ be a $(w+1) \times 2$ dimensional matrix such that each of its elements are n-bit binary strings and each of its rows are pairwise linearly independent. Let $q_c + (w+1)q_p \leq 2^{n-w}$ and $q_c \geq 9n$. It holds that

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

Proof of Theorem 3. The analysis extends and adapts that by Chen et al. to more permutations. In the real world, $\mathbf{D}$ has access to a construction oracle. It can ask at most q_c tuples of nonces and messages and will receive the corresponding ciphertexts. In the ideal world, the construction queries are answered by random bits of the expected length. Moreover, in both worlds, $\mathbf{D}$ 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.
We partition τ into $\tau = \tau_c \cup \tau_0 \cup \ldots \cup \tau_w$, where each partial transcript captures the queries and responses from a particular oracle. The construction transcript contains the keys, 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 τ 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, they are sampled uniformly at random. With their help, the adversary can 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 bad events holds and $\tau \in$ GoodT otherwise. We define sets $\mathcal{S}_{\alpha,\beta} =^{\text{def}} \{(i,j,k) : \widehat{V}_\alpha^i \oplus \widehat{V}_\beta^i = V_\alpha^j \oplus V_\beta^k\}$ 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]. 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 that would go to π_j in the real construction and produce $\widehat{V}_j^i$.

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

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

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

Lemma 4. Let $q_c + (w+1)q_p \leq 2^{n-w}$. It holds that

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

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

$$M^j \oplus (a_{\alpha,0} \cdot K_0 \oplus a_{\alpha,1} \cdot K_1) = U_\alpha^i \quad \text{and} \quad M^j \oplus (a_{\beta,0} \cdot K_0 \oplus a_{\beta,1} \cdot K_1) = U_\beta^k,$$

with $[a_{i,0}\ a_{i,1}]$ as the i-th row of the key-scheduling matrix. The equations can be seen as

$$\mathbf{A}' \cdot \mathbf{K} = \begin{bmatrix} a_{\alpha,0} & a_{\alpha,1} \\ a_{\beta,0} & a_{\beta,1} \end{bmatrix} \cdot \begin{bmatrix} K_0 \\ K_1 \end{bmatrix} = \begin{bmatrix} M^j \oplus U_\alpha^i \\ M^j \oplus U_\beta^k \end{bmatrix}$$

Since all rows of $\mathbf{A}$ are pairwise linearly independent, $\mathbf{A}'$ is non-singular. Moreover, K_0 and K_1 are uniform random variables over $\{0,1\}^n$. Thus, we can apply Lemma 3 and the probability of this event for a fixed choice of indices is 2^{-2n}. Over all indices, we obtain

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

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. Similar as bad_1, bad_3 considers a collision between a construction- and a primitive-query input as well as between a construction- and primitive-query output. It needs

$$M^j \oplus (a_{\alpha,0} \cdot K_0 \oplus a_{\alpha,1} \cdot K_1) = U_\alpha^i \quad \text{and} \quad \widehat{X}_\beta^j \oplus (a_{\beta,0} \cdot K_0 \oplus a_{\beta,1} \cdot K_1) = V_\beta^k,$$

where $[a_i b_i]$ is the i-th row of the key-scheduling matrix. The first equation reveals $\widehat{V}_\alpha^j = V_\alpha^i$, which yields $\widehat{X}_\alpha^j$ and thus $\widehat{X}_0^j = C_\alpha^j \oplus \widehat{X}_\alpha^j$. Then, $\mathbf{D}$ can deduce $\widehat{X}_\beta^j$ for all $\beta \neq \alpha$.

$$\mathbf{A}' \cdot \mathbf{K} = \begin{bmatrix} a_{\alpha,0} & a_{\alpha,1} \\ a_{\beta,0} & a_{\beta,1} \end{bmatrix} \cdot \begin{bmatrix} K_0 \\ K_1 \end{bmatrix} = \begin{bmatrix} M^j \oplus U_\alpha^i \\ \widehat{X}_\beta^j \oplus U_\beta^k \end{bmatrix}$$

Since $\mathbf{A}'$ is non-singular (a similar logic as in the bound of bad_1) and K_0 and K_1 are uniform random variables over $\{0,1\}^n$, we can apply Lemma 3 and the probability of this event for a fixed choice of indices comes out to be 2^{-2n}. Since bad_2 does not hold, there are at most θ such tuples. Over all indices, we obtain that $\Pr[\mathsf{bad}_3 | \neg\mathsf{bad}_2]$ is at most

$$\sum_{j\in[q_c]} \sum_{i\in[q_p]} \sum_{k\in[q_p]} \sum_{0\leq\alpha<\beta\leq w} \Pr\left[\widehat{U}_\alpha^j = U_\alpha^i \wedge \widehat{V}_\beta^j = V_\beta^k\right] \leq \frac{\binom{w+1}{2} q_p^2 q_c}{2^{2n}} + \frac{\binom{w+1}{2} q_p \sqrt{3nq_c}}{2^n}.$$

bad$_4$. A construction-query input collides with a primitive-query input, which allows deriving $\widehat{V}_\alpha^j$ that reveals all further permutation outputs for the j-th construction query. One of them collides with another construction-query output. The probability is at most 2^{-2n} since $\{\alpha, \beta, \gamma\}$ contain at least two independent indices. W.l.o.g., assume $\beta \neq \alpha$. C_α^j and C_β^j are chosen independently uniformly at random from $\mathbb{F}_2^n$. We obtain

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

bad$_5$. Here, the permutation inputs of two distinct construction queries collide with a primitive-query input each. Both input collisions allow us to find a candidate of $\widehat{V}_\alpha^j$ and $\widehat{V}_\gamma^k$ that reveals all further permutation outputs for both construction queries. Next, an output collides between the construction-query outputs. The probability for the collisions with the primitive-query inputs is 2^{-2n} since C_α^j and C_γ^k are chosen independently uniformly at random from $\mathbb{F}_2^n$. The probability of $\widehat{V}_\beta^j = \widehat{V}_\beta^k$ is again 2^{-n}. Thus, $\Pr[\mathsf{bad}_5]$ is at most

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

The bound in Lemma 4 follows. □

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

Lemma 5. It holds that

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

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 $\tau \in \textsc{GoodT}$. 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

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

We can easily bound 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 construction queries' answers. 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 find $|\mathsf{Comp}_{\text{real}}(\tau)|$. Like Chen et al., we regroup the queries from the transcript parts. We generalize their claim [CLM19] to cover all $w+1$ permutations:

Claim. Given $\tau \in \textsc{GoodT}$, no construction query $(M^j, C^j) \in \tau_c$ collides with more than one primitive query (U_α^i, V_α^i) for some $\alpha \in [0..w]$.

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 $\widehat{U}_\alpha^j = U_\alpha^i$, then (M^j, C_α^j) is removed from τ_c^{new} and $(U_\beta, V_\beta) = (\widehat{U}_\beta^j, \widehat{V}_\beta^j)$ 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

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

be the number of solutions that satisfy

(1) $\{(V_0^1, \ldots, V_0^j, \ldots, V_w^1, \ldots, V_w^j)\}$ satisfy the conditions recursively,

(2) For all $i \in [1..w]$, it holds that

$$V_0^{j+1} \oplus V_1^{j+1} = C_i^{j+1} \oplus (K_0 \oplus K_1) \oplus (2^i K_0 \oplus 2^{2i} K_1)\,.V_0^{j+1} \oplus V_w^{j+1} \tag{3}$$

(3) For all $i \in [0..w]$, it holds that $V_i^{\alpha+1} \notin \{V_i^1, \ldots, V_i^\alpha\} \cup V_i^{\mathsf{out}}$.

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

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

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

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

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 the inclusion-exclusion principle, it follows that

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

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

$$(4) = \prod_{j=0}^{s_0-1} \frac{2^n}{2^n - q_p - j} \cdot \ldots \cdot \prod_{j=0}^{s_w-1} \frac{2^n}{2^n - q_p - j} \cdot \prod_{i=0}^{q'-1} \frac{\lambda_{\alpha+1}}{\lambda_\alpha} \cdot \frac{(2^n)^w}{\prod_{j=0}^{w} (2^n - q_p - i - s_j)}$$

Algorithm 2 Definition of CENCPP.

101: **function** CENCPP$[\boldsymbol{\pi}, w].\mathcal{E}_{\mathbf{K}}(N, M)$
102: $(M_1, \ldots, M_m) \xleftarrow{n} M$
103: $\ell \leftarrow \lceil m/w \rceil$
104: **for** $i \leftarrow 0..\ell - 1$ **do**
105: $j \leftarrow i \cdot w$
106: $(S_{j+1} \| \cdots \| S_{j+w})$
107: $\leftarrow$ XORPP* $[\boldsymbol{\pi}, w]_{\mathbf{K}}(N \| \langle i \rangle_\mu)$
108: **for** $k \leftarrow j + 1..j + w$ **do**
109: $C_k \leftarrow \mathsf{msb}_{|M_k|}(S_k) \oplus M_k$
110: **return** $(C_1 \| \cdots \| C_m)$

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

301: **function** XORPP$[\boldsymbol{\pi}, w]_{\mathbf{K}}(M)$
302: $(K_0, K_1) \leftarrow \mathbf{K}$
303: $(\pi_0, \ldots, \pi_w) \leftarrow \boldsymbol{\pi}$
304: $U_0 \leftarrow M \oplus (K_0 \oplus K_1)$
305: $X_0 \leftarrow \pi_0(U_0) \oplus (K_0 \oplus K_1)$
306: **for** $j \leftarrow 1..w$ **do**
307: $L_j \leftarrow (2^j \cdot K_0) \oplus (2^{2j} \cdot K_1)$
308: $U_j \leftarrow M \oplus L_j$
309: $X_j \leftarrow \pi_j(U_j) \oplus L_j$
310: $C_j \leftarrow X_j \oplus X_0$
311: **return** $(C_1 \| \cdots \| C_w)$

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

using the fact that $q_p + q' + s_j \ll 2^{n-w}$. The bound in Lemma 5 follows. □

Our claim in Theorem 3 follows from Lemma 1, 4, and 5. □

5.3 CENCPP: An Instantiation of CENCPP*

A natural instantiation of CENCPP* can be realized by instantiating the key-scheduling matrix $\mathbf{A}$ of dimension $(w + 1) \times 2$ of XORPP* as follows:

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

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 $\mathbf{x}^1$ for practical values of $\mathbb{F}_{2^n}$. $p(\mathbf{x})$ is an irreducible modulus polynomial in $\mathbb{F}_{2^n}$. Note that any two rows of the above matrix $\mathbf{L}$ are linearly independent. We refer to the instantiation of XORPP* with matrix $\mathbf{L}$ as XORPP. We define the concrete nonce- and public-permutation-based encryption scheme CENCPP in Algorithm 2. Since any two rows in the key-scheduling matrix of CENCPP are linearly independent, the security of CENCPP follows from Theorems 2, and 3.

6 Domain-separated Variants

DS-SoEM is 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 DS-SoEM$[\pi]_{K_0,K_1} : (\mathbb{F}_{2^n})^2 \times \mathbb{F}_2^{n-d} \to \mathbb{F}_2^n$

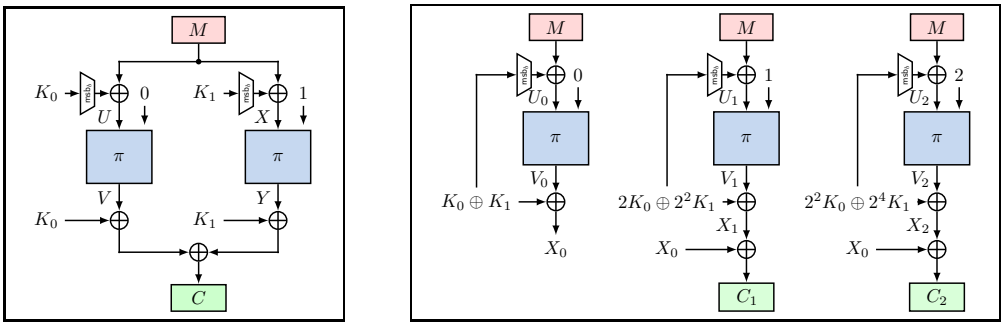
Algorithm 3 Definition of DS-CENCPP*, DS-XORPP*, and DS-SoEM.

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

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

301: **function** DS-XORPP*$[\pi, w]_{\mathbf{K}}(M)$
302: $(K_0, K_1) \leftarrow \mathbf{K}$
303: $U_0 \leftarrow (M \oplus \mathsf{msb}_{n-d}(K_0 \oplus K_1)) \| \langle 0 \rangle_d$
304: $X_0 \leftarrow \pi(U_0) \oplus (K_0 \oplus K_1)$
305: **for** $j \leftarrow 1..w$ **do**
306: $L_j \leftarrow (2^j \cdot K_0) \oplus (2^{2j} \cdot K_1)$
307: $U_j \leftarrow (M \oplus \mathsf{msb}_{n-d}(L_j)) \| \langle j \rangle_d$
308: $X_j \leftarrow \pi(U_j) \oplus L_j$
309: $C_j \leftarrow X_j \oplus X_0$
310: **return** $(C_1 \| \cdots \| C_w)$

401: **function** DS-SoEM$[\pi, w]_{\mathbf{K}}(M)$
402: $(K_0, K_1) \leftarrow \mathbf{K}$
403: $U \leftarrow (\mathsf{msb}_{n-d}(K_0) \oplus M) \| \langle 0 \rangle_d$
404: $X \leftarrow (\mathsf{msb}_{n-d}(K_1) \oplus M) \| \langle 1 \rangle_d$
405: $V \leftarrow \pi(U) \oplus K_0$
406: $Y \leftarrow \pi(X) \oplus K_1$
407: **return** $V \oplus Y$



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

Figure 6: The domain-separated constructions, here with DS-XORPP*$[\pi, 2]$. The trapezoids represent truncation of the key masks at the input to their $b = n - d$ most significant bits.

to compute DS-SoEM$[\pi]_{K_0,K_1}(M)$, as listed in Algorithm 3. Note that we use $(n-d)$ bits of the key in forward direction only, i.e., the domain is **not** masked. 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 6a.

DS-XORPP*. We can define DS-XORPP*$[\pi, w]$ similarly. 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 DS-XORPP*$[\pi, w] : (\mathbb{F}_{2^n})^2 \times \mathbb{F}_2^{n-d} \to (\mathbb{F}_2^n)^w$ as given in Algorithm 3 and shown in Figure 6b. 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 naturally. 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 DS-CENCPP*$[\pi, w]_{\mathbf{K}}(N, M)$ are provided in Algorithm 3.

7 Distinguishers on DS-SoEM and DS-XORPP*

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

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

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, get C^j. Moreover, query $M^{*j} = M^j \oplus \langle 1 \rangle_n$ to get 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. Query $U^{*i} = U^i \oplus 10^{n-1}$ 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 get Y^k. Query $X^{*k} = X^k \oplus 10^{n-1}$ to π to get 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

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

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 counter the distinct domains. While the distinguisher from [CLM19, Proposition 2] would still be inapplicable, a slide attack (cf. [DKS12, DDKS13]) could become. In the following, we consider a variant of DS-SoEM$[\pi]$ with the permutation inputs

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

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

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

Then, there exists a pair s. t. $M^i \oplus M^j = \mathsf{msb}_{n-d}(K^0 \oplus K^1)$. It follows 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 shows that the tempting approach of using a single-key domain-separated variant of DS-SoEM does not offer sufficient security in practice. Since the domain differs in both permutation calls, this would ensure distinct inputs on 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 $M^i = (\langle i \rangle_{n/2} \;\|\; 0^{n/2-d})$ to obtain C^i and store them. To each M^i, associate a plaintext $M'^i = M^i \oplus (10^{n-1-d})$ and its output C'^i.
2. For $j \leftarrow 1..2^{n/2-d}$, ask for the primitive encryption of $U^j = (\langle 0 \rangle_{n/2} \;\|\; \langle i \rangle_{n/2-d} \;\|\; \langle 0 \rangle_d)$ 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 $X^j = (\langle 0 \rangle n/2 \;\|\; \langle i \rangle_{n/2-d} \;\|\; \langle 1 \rangle_d)$ 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

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

Theorem 4. Let $\mathbf{D}$ be a distinguisher with at most q_c construction queries and q_p primitive queries each to $\pi^{\pm}(\cdot \,\|\, \langle 0 \rangle_d)$ and $\pi^{\pm}(\cdot \,\|\, \langle 1 \rangle_d)$. Let $q_c + 2q_p < 2^{n-3}$ and $q_c, q_p > 9n$. Then, $\mathbf{Adv}^{\mathsf{PRF}}_{\mathsf{DS\text{-}SoEM}[\pi]_{\mathbf{K}}}(\mathbf{D})$ is upper bounded by

$$\frac{(6 \cdot 2^d + 2^{2d}) q_c q_p^2}{2^{2n}} + \frac{2^{2d} q_c q_p^2}{2^{3n}} + \frac{q_c + 2 + 4q_p\sqrt{3nq_c}}{2^n} + \frac{2q_c(2q_c + 2q_p)^2}{2^{2n}}.$$

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

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

and their outputs as $\widehat{V}^j$ and $\widehat{Y}^j$. We also use 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 $\widehat{V}_1^j = \widehat{Y}^j$. Let $\mathcal{S} =^{\text{def}} \{(i,j,k) : C^i \oplus K_0 \oplus K_1 = V_0^j \oplus V_1^k\}$ 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 index j and two primitive query indices 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 index j and two primitive query indices 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 index j and two primitive query indices 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 index j and two primitive query indices 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 index j and two primitive query indices 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 query indices j and k and two distinct primitive query indices 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 query indices j and k and two distinct primitive query indices 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 query indices j and k and two distinct primitive query indices 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 index j such that $C^j = K_0 \oplus K_1$.

Lemma 6. Let $q_c + 2q_p < 2^{n-3}$. It holds that

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

The proof is given in Appendix A.

Good Transcripts. It remains to consider good attainable transcripts.

Lemma 7. It holds that

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

We note that this part is almost exactly as part of good transcripts in the proof of 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]. The proof is given in Appendix B.
Our claim in Theorem 4 follows from Lemma 1, 6, and 7. □

9 Security Analysis of DS-CENCPP*

We also studied 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 conducted a two-step analysis, where we consider (1) the PRF security of DS-XORPP*$[\pi, w]$ and (2) the PRF security of DS-CENCPP*$[\pi, w]$.

Theorem 5. It holds: $\mathbf{Adv}^{\mathsf{nE}}_{\text{DS-CENCPP}^*[\pi,w]_{\mathbf{K}}}(q_p, q_c, \sigma) \leq \mathbf{Adv}^{\mathsf{PRF}}_{\text{DS-XORPP}^*[\pi,w]_{\mathbf{K}}}\left(q_p, \frac{m}{w} q_c, \sigma\right)$.

The proof follows a similar argumentation as that of CENCPP*.

Theorem 6. Let $v =^{\text{def}} w + 1$, $q_c + vq_p \leq 2^{n-w}$, and $q_p, q_c > 9n$. It holds that $\mathbf{Adv}^{\mathsf{PRF}}_{\text{DS-XORPP}^*[\pi,w]_{\mathbf{K}}}(q_p, q_c, \sigma)$ is upper bounded by

$$\frac{\left(v^2 2^{2d} + v^2 2^d + v^3 2^d\right) q_c q_p^2 + v^3 2^d q_c q_p}{2^{2n}} + \frac{v^4 2^{2d} q_c^2 q_p^2}{2^{3n}} + \frac{v^2 q_c}{2^n} + \frac{3v^2 q_c^3 + 6v^3 q_c^2 q_p + 4v^4 q_c q_p^2}{2^{2n}} + \frac{(w+1)^2 + (w+1)^2 q_p \sqrt{3nq_c}}{2^n}.$$

Proof. Again, we employ the proof strategy from XORPP*. Here, the adversary can query q_p primitive queries to each domain-separated primitive $\pi^{\pm}(\cdot \,\|\, \langle i \rangle_d)$. We define sets $\mathcal{S}_{\alpha,\beta,\gamma} =^{\text{def}} \{(i,j,k) : C_\alpha^i \oplus (1 + 2^\alpha) K_0 \oplus (1 + 2^{2\alpha}) K_1 = V_\beta^j \oplus V_\gamma^k\}$ 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 $\alpha, \beta \in [0..w]$ s. t. $(\widehat{U}_\alpha^j = U_\alpha^i) \wedge (\widehat{U}_\beta^j = U_\beta^k)$.
- 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 $(\widehat{U}_\alpha^j = U_\alpha^i) \wedge (\widehat{V}_\beta^j = V_\beta^k)$.
- 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 $(\widehat{U}_\alpha^j = U_\alpha^i) \wedge (\widehat{V}_\beta^j = V_\gamma^k)$.
- 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 $(\widehat{U}_\alpha^j = U_\alpha^i) \wedge (\widehat{V}_\beta^j = \widehat{V}_\gamma^j)$.
- 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 $(\widehat{U}_\alpha^j = U_\alpha^i) \wedge (\widehat{U}_\gamma^j = U_\gamma^i) \wedge (\widehat{V}_\beta^j = \widehat{V}_\delta^\ell)$.
- bad_7: There exists a construction query index j and a permutation index $\alpha \in [w]$ such that $C_\alpha^j = (K_0 \oplus K_1) \oplus (2^\alpha K_0 \oplus 2^{2\alpha} K_1)$.

Our claim in Theorem 6 follows from Lemmas 1, 8, and 9. □

Lemma 8. Let $v =^{\text{def}} w + 1$ and $q_c + v \cdot q_p < 2^{n-3}$. Then, $\Pr[\Theta_{\text{ideal}} \in \text{BADT}]$ is upper bounded by

$$\frac{\left(v^2 2^{2d} + v^2 2^d + v^3 2^d\right) q_c q_p^2 + v^3 2^d q_c q_p}{2^{2n}} + \frac{v^4 2^{2d} q_c^2 q_p^2}{2^{3n}} + \frac{v^2 q_c + v^3 + v^3 q_p \sqrt{3nq_c}}{2^n}.$$

The proof is provided in Appendix C.

Lemma 9. Let $v =^{\text{def}} w + 1$ It holds that

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

The proof is given in Appendix D.

10 Conclusion

This work has proposed a variant of CENC from public permutations, CENCPP*. It is straightforward 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 to obtain an authenticated encryption scheme. 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 use $(K_0, K_0 \oplus K_1, K_1)$ for the calls to the permutations to ensure independent keys without the need for doubling. We see the recent summation-truncation-hybrid by Gunsing and Mennink [GM20] to be similar to the sum of permutation, although it is based on secret permutations. Adapting it to beyond-birthday-bound security with public permutations seems very interesting related future work.

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.

[BN18] Srimanta Bhattacharya and Mridul Nandi. Revisiting Variable Output Length XOR Pseudorandom Function. *IACR Trans. Symmetric Cryptol.*, 2018(1):314–335, 2018.

[CDK+18] Benoît Cogliati, Yevgeniy Dodis, Jonathan Katz, Jooyoung Lee, John P. Steinberger, Aishwarya Thiruvengadam, and Zhe Zhang. Provable Security of (Tweakable) Block Ciphers Based on Substitution-Permutation Networks. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO I*, volume 10991 of *LNCS*, pages 722–753. Springer, 2018.

[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.

[CNTY20] Avik Chakraborti, Mridul Nandi, Suprita Talnikar, and Kan Yasuda. On the Composition of Single-Keyed Tweakable Even-Mansour for Achieving BBB Security. *IACR Trans. Symmetric Cryptol.*, 2020(2):1–39, 2020.

[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 *LNCS*, 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 *LNCS*, 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.

[DHT17] Wei Dai, Viet Tung Hoang, and Stefano Tessaro. Information-Theoretic Indistinguishability via the Chi-Squared Method. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO Part III*, volume 10403 of *LNCS*, pages 497–523. Springer, 2017. Full version at http://eprint.iacr.org/2017/537, latest version 20170616:190106.

[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 *LNCS*, pages 336–354. Springer, 2012.

[DN20] Avijit Dutta and Mridul Nandi. BBB Secure Nonce Based MAC Using Public Permutations. In Abderrahmane Nitaj and Amr M. Youssef, editors, *AFRICACRYPT*, volume 12174 of *LNCS*, pages 172–191. Springer, 2020.

[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 *LNCS*, pages 263–293. Springer, 2016.

[GM20] Aldo Gunsing and Bart Mennink. The Summation-Truncation Hybrid: Reusing Discarded Bits for Free. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO I*, volume 12170 of *LNCS*, pages 187–217. Springer, 2020.

[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 *LNCS*, 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*, LNCS. 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.

A Analysis of Bad Transcripts of DS-SoEM

We restate the lemma to aid the reader.

Lemma 6. Let $q_c + 2q_p < 2^{n-3}$. It holds that

$$\Pr\left[\Theta_{\text{ideal}} \in \text{BADT}\right] \leq \frac{(6 \cdot 2^d + 2^{2d})q_c q_p^2}{2^{2n}} + \frac{2^{2d} q_c q_p^2}{2^{3n}} + \frac{q_c + 2}{2^n} + \frac{4q_p\sqrt{3nq_c}}{2^n}\,. \tag{7}$$

Proof. The event bad_1 considers the probability of two input collisions of one 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:

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

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

$$\begin{aligned}\Pr[\mathsf{bad}_5|\neg\mathsf{bad}_2] &= \sum_{j\in[q_c]} \sum_{i\in[q_p]} \sum_{k\in[q_p]} \Pr\left[\widehat{U}_0^j = U_0^i \wedge \widehat{V}_1^j = V_0^k\right] \\ &\leq \frac{q_c q_p^2}{(2^n-1)(2^{n-d})} + \frac{q_p\sqrt{3nq_c}}{2^n}\,.\end{aligned}$$

The bound of $\Pr[\mathsf{bad}_6|\neg\mathsf{bad}_2]$ is again analogous.
The event bad_7 requires 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 $\mathbf{D}$ 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 6 follows from

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

B Analysis of Good Transcripts of DS-SoEM

We restate the lemma to aid the reader.

Lemma 7. It holds that

$$\frac{\Pr[\Theta_{\text{real}} = \tau]}{\Pr[\Theta_{\text{ideal}} = \tau]} \geq 1 - \frac{2q_c(2q_p + 2q_c)^2}{2^{2n}}. \tag{8}$$

Proof. Again, we can write

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

Three out of four terms are again easy to bound:

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

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:

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

Moreover, we can bound

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

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

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

It remains to determine $|\mathsf{Comp}_{\text{real}}(\tau)|$. 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_α^i, V_α^i) 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 $\widehat{U}_0^j = U_0^i$, then (M^j, C^j) is removed from τ_c^{new} and $(U_1^i, V_1^i) = (\widehat{U}_1^j, \widehat{V}_1^j)$ is added to τ_1^{new}.
- If there exists an i such that $\widehat{U}_1^j = U_1^i$, then (M^j, C^j) is removed from τ_c^{new} and $(U_0^i, V_0^i) = (\widehat{U}_0^j, \widehat{V}_0^j)$ 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

$$\lambda_{\alpha+1} \stackrel{\text{def}}{=} \left|\left\{(V_0^1, \ldots, V_0^{\alpha+1}, V_1^1, \ldots, V_1^{\alpha+1})\right\}\right| \tag{9}$$

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

$$V_0^{\alpha+1} \oplus V_1^{\alpha+1} = C^{\alpha+1} \oplus K_0 \oplus K_1. \tag{10}$$

(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

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

We obtain

$$\frac{\Pr[\Theta_{\text{real}} = \tau]}{\Pr[\Theta_{\text{ideal}} = \tau]} = \frac{\lambda_{q'} \cdot (2^n - (q_1 + q_2 + 2q'))! \cdot (2^n)^{q_c}}{(2^n - 2q_p)!}\,. \tag{11}$$

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

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

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

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 (11) that

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

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

C Analysis of Bad Transcripts of DS-XORPP*

We restate the lemma to aid the reader.

Lemma 8. Let $v=^{\text{def}} w+1$ and $q_c+v\cdot q_p<2^{n-3}$. It holds that

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

$$\frac{v^4 2^{2d} q_c^2 q_p^2}{2^{3n}} + \frac{v^2 q_c + v^3 + v^3 q_p \sqrt{3nq_c}}{2^n}\,.$$

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

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

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

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

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

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

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

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

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

$$\begin{aligned}\Pr[\mathsf{bad}_6] &\le \sum_{1\le j<k\le q_c} \sum_{i\in[q_p]} \sum_{\ell\in[q_p]} \sum_{\alpha,\beta,\gamma,\delta\in[0..w]} \Pr\left[\widehat{U}_\alpha^j = U_\alpha^i \wedge \widehat{U}_\gamma^k = U_\gamma^\ell \wedge \widehat{V}_\beta^j = \widehat{V}_\delta^j\right] \\ &\le \frac{(w+1)^4 \binom{q_c}{2} q_p^2}{2^{2(n-d)} 2^n}\,.\end{aligned}$$

Finally, bad_7 represents the event that a construction query obtains equal outputs from both permutation calls, while the inputs are always distinct. Thus, $\widehat{V}_\alpha^j \oplus \widehat{V}_\beta^j = C_\alpha^j \oplus$

$C^j_\beta \oplus (2^\alpha K_0 \oplus 2^{2\alpha} K_1) \oplus (2^\beta K_0 \oplus 2^{2\beta} K_1)$ can never be zero for the real construction. The probability is upper bounded by

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

The bound in Lemma 6 follows from the sum of probabilities of the individual bad events. □

D Analysis of Good Transcripts of DS-XORPP*

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

Lemma 9. 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 $|\mathsf{Comp}_{\text{real}}(\tau)|$. 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, …, 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 $\widehat{U}_\alpha^j = U_\alpha^i$, then (M^j, C_α^j) is removed from τ_c^{new} and $(U_\beta, V_\beta) = (\widehat{U}_\beta^j, \widehat{V}_\beta^j)$ 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 $\pi(\cdot \,\|\, \langle i \rangle_d)$, 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 $\pi^{\pm}(\cdot \,\|\, \langle \iota \rangle_d)$ are denoted by V_ι^{out}, for $0 \leq \iota \leq w$. For $\alpha = 0, \ldots, q' - 1$, let

$$\lambda_{\alpha+1} \stackrel{\text{def}}{=} \left|\left\{(V_0^1, \ldots, V_0^{\alpha+1}, \ldots, V_w^1, \ldots, V_w^{\alpha+1})\right\}\right| \tag{12}$$

be the number of solutions that satisfy

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

(2) It holds that

$$\begin{aligned} V_0^{\alpha+1} \oplus V_1^{\alpha+1} &= C_1^{\alpha+1} \oplus K_0 \oplus K_1 \\ &\vdots \\ V_0^{\alpha+1} \oplus V_w^{\alpha+1} &= C_w^{\alpha+1} \oplus K_0 \oplus K_w. \end{aligned} \tag{13}$$

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

- ...

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

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

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

We obtain

$$\frac{\Pr[\Theta_{\text{real}} = \tau]}{\Pr[\Theta_{\text{ideal}} = \tau]} = \frac{\lambda_{q'} \cdot (2^n - (\sum_{i=0}^{w} q_i + (w+1)q'))!}{(2^n - (w+1)q_p)!} \cdot (2^n)^{w \cdot q_c}. \tag{14}$$

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.ι : i), for $i = 1, \ldots, \alpha + \sum_{k=0}^{w} |V_k^{\mathsf{out}}|$. From inclusion-exclusion, it follows that

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

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

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 (14) that

$$\begin{aligned}(14) &= \prod_{j=0}^{w\cdot s-1} \frac{2^n}{2^n - (w+1)q_p - j} \cdot \prod_{j=0}^{q'-1} \frac{\lambda_{\alpha+1}}{\lambda_\alpha} \cdot \frac{(2^n)^w}{\prod_{i=0}^{w}(2^n - \sum_{k=0}^{w} q_k - iq' - j)} \\ &\geq \prod_{j=0}^{q'-1} \frac{(2^n - (w+1)j - (w+1)\sum_{i=0}^{w} q_i)(2^n)^w}{\prod_{i=0}^{w}(2^n - \sum_{k=0}^{w} q_k - iq' - j)} \,.\end{aligned}$$

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

$$\prod_{j=0}^{q'-1} \frac{(2^n - (w+1)(q' + q_{\mathsf{sum}})(2^n)^w}{\prod_{i=0}^{w}(2^n - (q_{\mathsf{sum}} + q'))} \geq \prod_{j=0}^{q'-1} \frac{(2^n - (w+1)(q' + q_{\mathsf{sum}})(2^n)^w}{(2^n - (q_{\mathsf{sum}} + q'))^{w+1}}. \tag{15}$$

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

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

It follows that

$$(15) \geq \left(\frac{z}{(2^n)^{w+1} - \binom{w+1}{1}(2^n)^w(q_{\mathsf{sum}} + q') + \binom{w+1}{2}(2^n)^{w-1}(q_{\mathsf{sum}} + q')^2}\right)^{q'}$$

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

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

$$1 - \frac{(w+1)^2(q_c^3 + q_c^2(q_c + 2wq_p) + q_c(q_c + 2wq_p)^2}{2^{2n}}$$
$$\geq 1 - \frac{(w+1)^2(q_c^3 + q_c^3 + 2w \cdot q_c^2 q_p + q_c^3 + 4w \cdot q_c^2 q_p + 4w^2 \cdot q_c q_p^2)}{2^{2n}}$$
$$\geq 1 - \frac{3(w+1)^2 q_c^3 + 6(w+1)^3 q_c^2 q_p + 4(w+1)^4 q_c q_p^2}{2^{2n}}. \quad \square \tag{16}$$