Journal of *Cybersecurity and Privacy*

MDPI

Article

SoP-CTR: A Lightweight Stream Cipher with Beyond-Birthday-Bound Security for IoT and Embedded Devices

Andriani Adi Lestari [1], Suryadi MT [2], Kalamullah Ramli [1,*], Susila Windarta [3] and Teddy Surya Gunawan [4]

[1] Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok 16424, Jawa Barat, Indonesia; andriani.adi@ui.ac.id
[2] Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Indonesia, Depok 16424, Jawa Barat, Indonesia; yadi.mt@sci.ui.ac.id
[3] Department of Cybersecurity, Politeknik Siber dan Sandi Negara, Bogor 16120, Jawa Barat, Indonesia; susila.windarta@poltekssn.ac.id
[4] Department of Electrical and Computer Engineering, Kulliyyah of Engineering, International Islamic University Malaysia, Kuala Lumpur 50728, Malaysia; tsgunawan@iium.edu.my
* Correspondence: kalamullah.ramli@ui.ac.id

Abstract

As systems migrate toward post-quantum readiness, symmetric keys are kept longer and protect larger data volumes, a regime in which keystream security beyond the birthday bound matters. The dominant block–cipher counter mode, AES-CTR (Advanced Encryption Standard, counter mode), caps keystream security at the birthday bound regardless of key length, and, to our knowledge, no lightweight stream cipher pairs provable beyond-birthday-bound security with a permutation standardised by the National Institute of Standards and Technology (NIST). We propose SoP-CTR, a stream cipher built from the NIST-standardised Ascon-p_{12} permutation via the Sum of Permutations (SoP) paradigm. The z-split SoP framework achieves single-key beyond-birthday-bound (BBB) security by XORing a split constant into the counter word, eliminating independent key derivation. We prove an explicit two-term pseudorandom-function (PRF) advantage bound $\mathrm{Adv} \leq q^{3/2}/2^{320} + 2pq/2^{256}$, valid for pairwise-distinct counters in $[0, 2^{63})$. The online term is negligible throughout this 2^{63}-block range, and the offline term bounds adversarial key-search and is the binding constraint for post-quantum offline budgets. The SoP construction provably eliminates the birthday distinguisher that breaks a hypothetical single-call CTR-Ascon at $q \approx 2^{160}$. We also give the first quantum analysis of this domain-separated single-key variant in the Q1 model, where construction queries are classical and the public permutation may be evaluated quantumly, obtaining an attack at $2^{85.3}$ and an unconditional lower bound of $2^{64.3}$, so the post-quantum figure is the best known attack cost of $2^{85.3}$ rather than the generic 2^{128} Grover level. On a physical STM32F407 ARM Cortex-M4 (168 MHz), hardware cycle-counter measurements yield 61.44 cycles per byte (cpb) for the unrolled variant, about 4.6% faster than Ascon-128 keystream extraction (64.39 cpb), and 63.78 cpb for the memory-optimised rolled variant (3832 bytes Flash), while providing a strictly stronger BBB guarantee. An x86-64 AVX2 two-block variant achieves 2.57 cpb at 1 KiB.

Keywords: stream cipher; beyond-birthday-bound security; post-quantum cryptography; IoT security; lightweight cryptography; Ascon permutation; sum of permutations; ARM Cortex-M4; embedded systems security

Check for updates

Received: 11 June 2026
Revised: 13 August 2026
Accepted: 29 August 2026
Published: 8 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The proliferation of Internet of Things (IoT) devices, including industrial sensors, smart meters, wearable health monitors, and autonomous vehicles, has made confidential communication between constrained endpoints a central challenge in modern cybersecurity. These devices transmit sensitive operational data continuously and often operate under long-lived symmetric keys, since frequent rekeying is impractical in battery-constrained or intermittently connected deployments. At the same time, the anticipated arrival of cryptographically relevant quantum computers places additional pressure on existing symmetric primitives. Grover's algorithm [1] reduces the effective key-search cost of the Advanced Encryption Standard (AES) with a 256-bit key, AES-256, to 2^{128} quantum operations. Any birthday-bound weakness in the keystream layer compounds this threat by providing a second, data-volume-based attack surface independent of key length [2].

The core vulnerability is a structural property of counter-mode stream ciphers built from a single bare permutation call. For such designs the keystream security threshold is set by the state or block size, not the key size [2]. AES-256 in counter (CTR) mode carries a 256-bit key, but its 128-bit block means keystream security caps at roughly 2^{64} blocks per key, no better than AES-128 [3,4]. ChaCha20 avoids this birthday cap through its feed-forward block function, whose keystream security does not degrade with the number of blocks generated [5]. Its only per-nonce ceiling is counter-field exhaustion, with the Internet Engineering Task Force (IETF) variant's 32-bit counter [6] capping each nonce to 2^{32} blocks and Bernstein's original 64-bit design [7] to 2^{64} blocks. The birthday-bound keystream limit is therefore specific to bare-permutation counter modes such as AES-CTR, where under post-quantum assumptions the quantum key-search bound no longer separates from that limit. Crucially, this is not a conservative proof artifact. We show in Section 5.3 that a concrete online-only distinguisher breaks any single-call CTR cipher with advantage $\geq$0.39 at $q = 2^{n/2}$ queries, with no offline access required.

The demand for beyond-birthday-bound (BBB) stream encryption is already visible in deployed protocols and standards. The National Institute of Standards and Technology (NIST) Special Publication SP 800-38D [4] caps Galois/Counter Mode (GCM) to at most 2^{32} authenticated-encryption invocations per key, because security assurance degrades as more data is processed under a single key [4]; the underlying birthday-collision analysis is formalised in Luykx and Paterson [3]. Transport Layer Security (TLS) 1.3 [6] addresses this directly by capping AES-GCM at $2^{24.5}$ full-size records per connection with a 2^{-57} safety margin and mandating a `KeyUpdate` before those limits are reached, citing [3] as the authoritative bound. The same pressure is visible in adjacent domains, where the HCTR^{++} preprint [8] upgrades HCTR2 [9] precisely to achieve BBB security for length-preserving sector encryption. What remains absent is a counterpart at the keystream layer itself, namely a composable, single-pass BBB stream cipher that could serve as a candidate keystream component in settings now using AES-CTR or ChaCha20, such as TLS, IPsec, and constrained-IoT schemes for authenticated encryption with associated data (AEAD). A drop-in replacement would additionally require a full mode specification, which is beyond the scope of this work.

The Sum of Permutations (SoP) paradigm [10] is the natural theoretical foundation for BBB stream cipher construction. The SoP paradigm XORs two outputs of the same permutation at domain-separated inputs to obtain a stronger pseudorandom function than either call alone. Two domain-separated calls to a single permutation provably tighten the PRF advantage bound from $\mathcal{O}(q^2/2^n)$ down to $\mathcal{O}(q^{3/2}/2^n)$ [11,12], giving security up to $q \approx 2^{2n/3}$ queries. The classical construction, however, requires the two permutation calls to use independently random keys. This constraint doubles key material, forces two separate key schedules, and adds measurable initialisation overhead on constrained hardware.

https://doi.org/10.3390/jcp6050157

We resolve this tension by replacing *key* independence with *input* independence. Keeping a single master key and XORing a *split constant* c into the counter word (block i feeds counter i into one call and $i \oplus c$ into the other) guarantees disjoint inputs whenever the counters are pairwise distinct and stay within $[0, 2^z)$, provided c has the appropriate high bit set. Sequential increment is the natural usage model of a stream cipher that enforces this, so the constraint imposes no practical restriction. We formalise this as the *z-split SoP framework*: a parameterised family of BBB stream ciphers for any n-bit permutation, secret key width κ, and split parameter z.

The resulting construction, SoP-CTR, instantiates this idea over Ascon-p_{12} [13,14], the 320-bit permutation underlying the NIST Lightweight Cryptography standard (NIST SP 800-232, 2025). With $n = 320$, the online BBB security threshold reaches $q \approx 2^{213}$ blocks for small offline query budgets p (53 bits beyond the single-call bound), while the offline binding term $2pq/2^{256}$ provides strong post-quantum resistance against key-search adversaries.

The main contributions of this work are as follows:

- The z-split SoP framework (Section 4): a parameterised family of stream ciphers achieving PRF-distinguishing BBB security in the ideal-permutation model from a single master key, for any n-bit permutation, key width κ, and split parameter z, with a structural condition ($n \geq 3\kappa + 3$) for simultaneously reaching the BBB limit and post-quantum key security.
- The SoP-CTR construction (Section 4): the canonical instantiation over Ascon-p_{12} ($n = 320$, $\kappa = 256$, $z = 63$), fully specified with single-word counter-complement domain separation and no key-derivation overhead.
- Formal security analysis (Section 5): a concrete birthday distinguisher proving the necessity of BBB, followed by an explicit two-term BBB proof for the z-split framework via the H-coefficient technique [15].
- The first quantum analysis of the domain-separated single-key single-permutation sum (Section 5.6), a variant left as an open direction by Datta et al. [16]. We give a Q1 attack at $2^{85.3}$ and an unconditional Q1 lower bound of $2^{64.3}$, which together replace the generic 2^{128} Grover figure as the post-quantum statement for this construction.
- A Cortex-M4 ARMv7-M implementation (Section 6): a word-split inline-assembly back-end maintaining the full 320-bit Ascon state in ten 32-bit register pairs throughout all 12 rounds, achieving 61.44 cpb on a physical STM32F407 at 168 MHz, verified by hardware cycle-counter measurements.
- An x86-64 implementation using AVX2 single-instruction multiple-data (SIMD) vectors (Section 6.5): a two-block parallel variant packing four Ascon-p_{12} states into five YMM registers and achieving 2.57 cpb at 1 KiB, a strategy structurally unavailable to sequential sponge modes.

The five deployment-relevant properties we compare against are, first, a beyond-birthday PRF-distinguishing bound rather than a birthday-bound or heuristic one; second, a proof in the ideal-permutation model rather than the keyed-cipher model; third, operation under a single master key with no derived second key; fourth, composable keystream output usable as a bulk-encryption component; and fifth, an implementation on Cortex-M4 verified on physical hardware. Table 1 shows that no surveyed construction satisfies all five at once, a gap in SoP-CTR addresses. Section 2 examines each scheme in turn. The survey covers permutation-based or BBB-oriented primitives with a documented security bound, so the claim is made to the best of our knowledge within that scope rather than as an absolute statement.

The paper proceeds as follows. Section 2 surveys related work, and Section 3 fixes notation while recalling the Ascon permutation and the SoP paradigm. Section 4 specifies

SoP-CTR itself, whose security bound is then established in Section 5. Turning to practice, Section 6 details the Cortex-M4 implementation and Section 7 reports the hardware measurements. Section 8 weighs deployment scenarios against the limitations and open problems that remain, and Section 9 concludes.

2. Related Work

This section surveys related work on lightweight stream ciphers, Ascon-based keystream constructions, and BBB designs, and identifies the gap that SoP-CTR fills. Each family is assessed against the same five deployment-relevant properties, so that the gap stated at the end is a property of the survey taken as a whole rather than a shortcoming of any single scheme.

2.1. Lightweight Stream Ciphers

Grain-128 [17] and Trivium [18] are hardware-oriented stream ciphers from the eSTREAM portfolio. They achieve sub-1 cpb on dedicated hardware but are inefficient on software platforms and do not provide BBB guarantees. ChaCha20 [7] is the dominant software stream cipher. Its add-rotate-XOR (ARX) structure maps well to 32-bit SIMD, and its 256-bit key provides a Grover margin of roughly 2^{128}. Unlike block–cipher counter modes, its keystream security does not degrade with the number of blocks generated [5], though it provides no ideal-permutation BBB bound.

2.2. Ascon-Based Keystream

Ascon-128 and Ascon-128a support keystream mode by extracting the rate output during the squeezing phase of the duplex sponge. Both are birthday-bound with capacity c giving $2^{c/2}$ block security; neither achieves BBB. ISAP [19] uses Ascon-p in a re-keying mode to provide side-channel resistance; its PRF-distinguishing security remains birthday-bound (2^{128}, capacity $c = 256$), and the re-keying mechanism addresses a different goal from PRF-distinguishing BBB security. Farfalle [20] and its Kravatte instantiation over Keccak-p achieve parallel keystream generation from a public permutation with single-key security, but the PRF-distinguishing advantage remains birthday-bound and to our knowledge, no efficient Cortex-M4 variant has been reported for the 1600-bit Keccak-p state.

2.3. BBB Constructions

Lucks [10] established the SoP paradigm. Patarin's mirror theory [11,21] provides the combinatorial machinery (lower bounds on the number of permutation tuples satisfying systems of XOR equalities with $\xi_{\max} = 2$) underlying the beyond-birthday analysis of sums of pseudorandom permutations; Bhattacharya and Nandi [12] gave a tight reproof via the χ^2 method for the secret-permutation model; and Chen, Lambooij, and Mennink [22] extended this to the ideal-permutation model, accounting for direct adversarial queries to the public permutation.

A distinct line of work asks how far the randomness in such sums can be reduced. Sibleyras and Todo [23] simplified the sum of Even-Mansour ciphers (SoEM) into the keyed sum of permutations (KSoP) $P_1(x \oplus k_1) \oplus P_2(x \oplus k_2)$, which drops one key addition but still needs two independent keys and two independent permutations. They also proved a negative result. Deriving the second key through an invertible linear key schedule admits key recovery at the birthday bound, so that particular simplification is ruled out. Mennink [24] minimised the construction further with sirP, sirK, and sirX, which reduce the number of permutations, of keys, or of both by overlapping the key and the data input instead of adding them. That work also observes, among the alternatives it compares, that a single-permutation domain-separated sum reaches the same $2^{2n/3}$ level provided the data

input is restricted to at most $n - 1$ bits, since otherwise two queries yield the same output. SoP-CTR follows this restricted-input route and develops it into a concrete stream cipher, and Section 2.5 states the resulting novelty boundary precisely.

Several lines of work achieve beyond-birthday-bound guarantees for stream ciphers, though in different security notions or models. Hamann and Krause [25] showed that Lizard achieves a $2^{2\kappa/3}$ key-recovery bound against time-memory-data tradeoff (TMDTO) attacks; however, the PRF-distinguishing security of Lizard remains at the birthday bound $2^{\kappa/2}$, which is the stronger notion required for composable keystream use. Chen et al. [26] studied stream encryption with key erasure for forward security, proving tight multi-user bounds; their focus is compromise-resilience in a generic PRF/block–cipher model, which differs from the single-session PRF-distinguishing setting addressed by SoP-CTR. Among block–cipher-based designs, CENC [27] achieves BBB stream encryption from a single key but operates in the keyed block–cipher model rather than the public-permutation setting. HCTR2 [9] is limited to birthday-bound security ($O(2^{n/2})$) [8] and is a length-preserving wide-block cipher, not suited to composable keystream generation.

2.4. Comparison and Gap Analysis

Table 1 synthesises the limitations identified above across five properties that matter jointly for constrained IoT deployment. Each scheme discussed so far fails at least one of them, whether through the capacity-birthday cap that exposes it to the distinguishing attack of Section 5.3, the absence of an ideal-permutation BBB bound, or a security model that does not expose composable keystream from a public permutation.

Two readings of the table need stating. First, the nature of the PRF bound differs by family. For the hardware-oriented stream ciphers it is the generic key-search birthday threshold $2^{\kappa/2}$ rather than a formal PRF-distinguishing result in the ideal-permutation model, for the permutation-based designs it is capacity-based, for GIFT-COFB it is the block-cipher birthday bound, and for Romulus-N it is the standard-model security of the tweakable block cipher. TMDTO-BBB denotes beyond-birthday security only against time-memory-data tradeoff adversaries. The last row refers to the SoP-CTR instantiation over Ascon-p_{12} with $n = 320$ and $\kappa = 256$, whereas the z-split framework generalises to any n-bit permutation. Second, a "No" in the M4 column means only that no Cortex-M4 implementation has been reported or independently verified in the cited literature, and does not imply that such an implementation is infeasible.

Three NIST Lightweight Cryptography finalists with verified Cortex-M4 implementations warrant separate mention. Schwaemm (Sparkle) [28] operates in the ideal-permutation model and produces keystream, yet its 128-bit capacity places the keystream PRF-distinguishing threshold at the capacity birthday 2^{64}, below the beyond-birthday regime. GIFT-COFB [29] is a birthday-bound block-cipher AEAD scheme, and Romulus-N [30] is a tweakable-block-cipher AEAD with standard-model security. Both operate in the keyed-cipher model rather than the ideal-permutation model, and neither exposes composable keystream.

No surveyed construction therefore achieves PRF-distinguishing BBB security in the ideal-permutation model from a single master key via counter-based domain separation with a Cortex-M4 implementation verified on physical hardware. The closest analyses are theoretical. Mennink [24] notes that a single-permutation domain-separated sum can reach the same level under a restricted input, without giving a concrete instantiation, while the minimised constructions sirP, sirK, and sirX reach it by overlapping key and input rather than by restricting the counter. What remains absent is a fully specified stream cipher of this kind, proved for its own correlated-key structure and verified on constrained hardware. SoP-CTR addresses this gap.

Table 1. Comparison of related schemes across five deployment-relevant properties. IP = ideal-permutation model. KC = keyed-cipher model. Heur. = heuristic, that is cryptanalysis-based. ARX = add-rotate-XOR, also heuristic. KS = composable keystream output. M4 = verified Cortex-M4 implementation reported in the cited work or in this paper. PRF = pseudorandom function. BBB = beyond-birthday-bound. TMDTO = time-memory-data tradeoff. KSoP = keyed sum of permutations. spds-SoEM = single-permutation domain-separated sum of Even-Mansour ciphers. The PRF bound column uses each scheme's own key length κ and its state or block size n.

Scheme	PRF Bound	Model	Single Key	KS	M4
Grain-128 [17]	Birthday (2^{64})	Heur.	Yes	Yes	No
Trivium [18]	Birthday (2^{40})	Heur.	Yes	Yes	No
ChaCha20 [7]	No data bound ‡	ARX	Yes	Yes	Yes
Ascon-128 [13]	Birthday (2^{128})	IP	Yes	Yes	Yes
Ascon-128a [13]	Birthday (2^{96})	IP	Yes	Yes	Yes
ISAP [19]	Birthday (2^{128})	IP	Yes	Yes	No
Kravatte [20]	Birthday (2^{128})	IP	Yes	Yes	No
Lizard [25]	TMDTO-BBB ($2^{2\kappa/3}$)	Heur.	Yes	Yes	No
CENC [27]	BBB ($q^{3/2}/2^n$)	KC	Yes	Yes	No
Schwaemm [28]	Birthday (2^{64})	IP	Yes	Yes	Yes
GIFT-COFB [29]	Birthday (2^{64})	KC	Yes	No	Yes
Romulus-N [30]	Standard (2^{128})	KC	Yes	No	Yes
KSoP [23]	BBB ($2^{2n/3}$)	IP	No	Yes	No
spds-SoEM [24]	BBB ($2^{2n/3}$) §	IP	Yes	Yes	No
SoP-CTR (this work)	**BBB ($q^{3/2}/2^{320}$)** †	**IP**	**Yes**	**Yes**	**Yes**

† Online BBB threshold $q \approx 2^{213}$ for $p \leq 2^{42}$ offline queries. ‡ ChaCha20's keystream security does not degrade with data volume [5], unlike block-cipher counter modes. It is a dedicated ARX design without an ideal-permutation BBB bound. § Discussed as an alternative construction in [24], with a bound carried over from a neighbouring scheme and no concrete instantiation. Bold marks the scheme proposed in this article.

2.5. Novelty Boundary

Because the individual components used here are well known, we state explicitly what this work inherits and what it contributes. We inherit the SoP paradigm together with its beyond-birthday analysis in the ideal-permutation model [10,22], and we use Ascon-p_{12} [13,14] unchanged.

The contribution is the security of the most aggressive simplification of that paradigm, a single permutation driven by a single master key whose two branch masks differ by a fixed public constant. This simplification is not free. Under unrestricted inputs the construction breaks outright, because counters i and $i \oplus c$ make the two permutation inputs coincide pairwise and force the output to zero, which is a deterministic two-query distinguisher. Confining the admissible counters to $[0, 2^z)$ removes exactly those pairs, and Theorem 1 establishes beyond-birthday security once they are gone.

The choice of a constant offset is also less arbitrary than it appears. The birthday-bound key recovery of Sibleyras and Todo [23], noted in Section 2, applies exactly when the second key is derived by an invertible linear map Γ, since their query subspaces are built from $I \oplus \Gamma$ and $I \oplus \Gamma^{-1}$. A constant offset is the case $\Gamma = I$, where $I \oplus \Gamma$ is singular and that construction degenerates, so the attack does not transfer. What remains is the meet-in-the-middle path that arises because the split constant is public. That path is analysed in Section 5.6, where it is shown to match the offline term $2pq/2^{256}$ of Theorem 1, and the same public map is what the quantum attack of that section exploits.

Relative to the closest prior analyses the difference lies in depth and in scope. Mennink [24] records the restricted-input variant as one alternative among minimised constructions, with a bound carried over from a neighbouring scheme. We give a self-

contained proof for the concrete correlated-key structure, including the coset invariance of Lemma 2, and instantiate the framework as a working stream cipher with a nonce interface, a multi-nonce statement (Proposition 1), a ten-vector known-answer suite, and implementations measured on physical hardware.

What the work does not claim is equally explicit. It claims no new permutation, no proof in the standard model, and no authenticated-encryption mode. In the quantum setting it claims neither Q2 security nor a tight Q1 bound, only the attack and the unconditional lower bound reported in Section 5.6.

3. Preliminaries

This section introduces the notation, the underlying Ascon permutation, and the SoP paradigm that together form the building blocks of SoP-CTR. The material is standard and is recalled solely to fix the notation and conventions on which the construction and its security analysis depend.

3.1. Notation

We write $\{0,1\}^n$ for the set of n-bit strings and use $\oplus$ for bitwise XOR. The concatenation of strings a and b is written $a\|b$. $\langle i \rangle_k$ denotes the k-bit little-endian encoding of non-negative integer i. All quantities are in bits unless stated otherwise.

3.2. The Ascon Permutation

Ascon [13] is a family of authenticated encryption and hashing schemes standardised as NIST SP 800-232 [14], following its selection in the NIST Lightweight Cryptography standardisation process [31]. At its core is a 320-bit permutation $\mathcal{P}_r$ parameterised by the number of rounds $r \in \{1, \ldots, 12\}$. We write $\mathcal{P}$ for $\mathcal{P}_{12}$, the full-round variant used in Ascon-128 initialisation and finalisation.

Definition 1 (Ascon-p_r state)**.** *The state $S \in \{0,1\}^{320}$ is viewed as five 64-bit words $S = (x_0, x_1, x_2, x_3, x_4)$. Each round applies a round-constant addition p_C, a nonlinear substitution layer p_S that applies a 5-bit S-box to each column, and a linear diffusion layer p_L consisting of two rotation-based mixing operations per word.*

3.3. The Sum of Permutations

The SoP paradigm constructs a pseudorandom function from two evaluations of a single permutation at domain-separated inputs. The keyed-cipher version was introduced by Bellare and Impagliazzo [2] and formalised for PRFs by Lucks [10] (the TWIN^d construction, where a single keyed pseudorandom permutation (PRP) is evaluated at consecutive inputs). In the *public-permutation model* relevant to permutation-based designs such as Ascon, the analogous construction takes the form:

$$\mathrm{SoP}_{K_1,K_2}(x) = \mathcal{P}(K_1 \oplus x) \oplus \mathcal{P}(K_2 \oplus x), \tag{1}$$

where $\mathcal{P}$ is a *public* permutation and K_1, K_2 are domain-separated working keys derived from a single master key. When $K_1 \neq K_2$ the two inputs are distinct, and the construction provides *beyond-birthday-bound* PRF security. Beyond-birthday security of order $q^{3/2}/2^n$ for such public-permutation SoP constructions was established via Patarin's mirror theory [11,21] and extended to the ideal-permutation model with explicit offline-query accounting by Chen, Lambooij, and Mennink [22]. A distinguisher requires $q \approx 2^{2n/3}$ queries to achieve constant advantage, compared with $q \approx 2^{n/2}$ (birthday bound) for a single permutation call, where $n = 320$ is the state size.

4. The SoP-CTR Construction

We now specify the construction. We first describe the general z-split SoP framework, which is a parameterised recipe for building a BBB stream cipher from any permutation, and then instantiate it as SoP-CTR over Ascon-p_{12}.

Definition 2 (z-Split SoP Stream Cipher). *Let $P : \{0,1\}^n \to \{0,1\}^n$ be a permutation, $\kappa \le n$ the secret key width, and z a split parameter with $0 < z < n - \kappa$. Let $c \in \{0,1\}^n$ be a split constant with bit z set to 1. Given master key $K \in \{0,1\}^\kappa$, nonce N, and counter $i \in [0, 2^z)$, the keystream block is*

$$\mathrm{KS}[i] \;=\; P(\mathsf{enc}(K, N, i)) \oplus P(\mathsf{enc}(K, N, i \oplus c)),$$

where enc *maps (K, N, i) to an n-bit input block in which the counter occupies a dedicated $(z+1)$-bit field that is disjoint from the κ-bit secret-key region and injective in i (all other inputs fixed). Concretely, writing the n-bit state as the concatenation of a κ-bit key region, an $(n - \kappa - z - 1)$-bit zero-filled public region, and a $(z+1)$-bit counter field, the encoding is*

$$\mathsf{enc}(K, N, i) \;=\; \big(K \oplus \mathsf{pad}_\kappa(N)\big) \,\|\, \mathbf{0}^{n-\kappa-z-1} \,\|\, \langle i \rangle_{z+1},$$

with $\langle i \rangle_{z+1}$ the $(z+1)$-bit little-endian encoding of i and pad_κ the zero-extension of the nonce to κ bits. Injectivity in i is immediate because $\langle \cdot \rangle_{z+1}$ is injective and occupies its own field. Section 4.3 gives the SoP-CTR instantiation of this encoding, and Figure 1 depicts it. The nonce N is incorporated as a public tweak XORed into the key region rather than as a separate field, so the field widths sum to exactly n with no overflow; the secret key still contributes κ bits of entropy to the input. The constraint $z < n - \kappa$ guarantees that the counter field and the key region fit disjointly within the n-bit state.

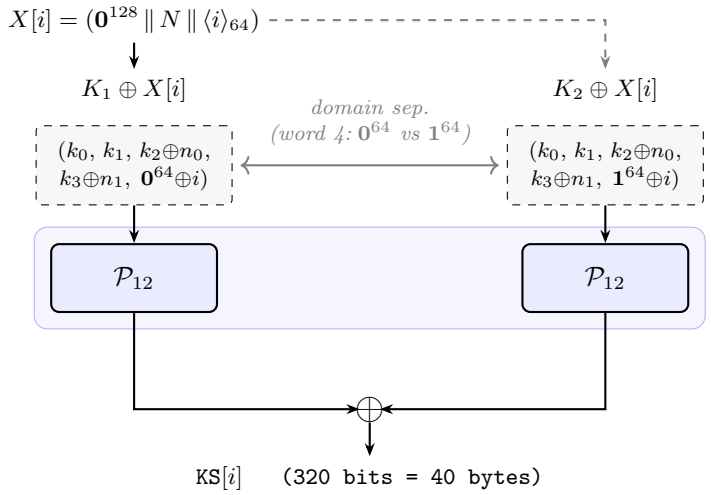


Figure 1. SoP-CTR block function. Two domain-separated Ascon-p_{12} calls with z-split inputs (word 4: $\mathbf{0}^{64}$ vs. $\mathbf{1}^{64}$, split at bit $z = 63$) are XORed to produce a 320-bit keystream block.

Why bit z suffices. The sole requirement on c is that bit z of c equals 1. For any counter $i < 2^z$, bit z of i is 0, so $i \oplus c$ has bit z equal to 1 and hence $i \oplus c \ge 2^z > i$. This guarantees that the two input families $\{\mathsf{enc}(K, N, i)\}$ and $\{\mathsf{enc}(K, N, i \oplus c)\}$ are always disjoint, which is the structural prerequisite for the BBB proof. The lower z bits of c are free. The minimal example sets only bit z, i.e., $c = 2^z$, whereas SoP-CTR uses the all-one word $c = \mathbf{1}^{64}$ (equivalently $z = 63$ with all lower bits also set), which satisfies bit z of c = 1, is fixed-point-free over the counter field $[0, 2^z)$, and additionally makes $A[i]$ and $B[i]$ differ in *every* bit of word 4. Both choices yield the same security guarantee, since the proof depends only on bit z of c being set.

4.1. Overview

SoP-CTR is a stream cipher built by instantiating the SoP pseudorandom function (1) with $\mathcal{P} = \mathcal{P}_{12}$ and applying it in counter mode. One keystream *block* is 320 bits (40 bytes), matching the natural output width of the Ascon-p state. The master key is 256 bits and the nonce is 128 bits.

Figure 1 shows the resulting block function. Reading it from the top, the same 320-bit input $X[i]$ enters both branches, each branch XORs its own key word into word 4 so that the two permutation inputs differ exactly in that word, and the two Ascon-p_{12} outputs are XORed to form one keystream block. The figure makes the single point on which the whole construction rests visible at a glance, namely that domain separation is achieved inside the permutation input rather than by deriving a second key.

4.2. Key Setup

Let $K \in \{0,1\}^{256}$ be the master key and $N \in \{0,1\}^{128}$ the nonce. Interpret K as four 64-bit little-endian words $K = (k_0, k_1, k_2, k_3)$. The two 320-bit working keys are

$$K_1 = (k_0,\ k_1,\ k_2,\ k_3,\ \mathbf{0}^{64}), \tag{2}$$
$$K_2 = (k_0,\ k_1,\ k_2,\ k_3,\ \mathbf{1}^{64}). \tag{3}$$

K_1 and K_2 share words 0–3 and differ only in word 4, where $K_1[4] = \mathbf{0}^{64}$ and $K_2[4] = \mathbf{1}^{64}$. Here $\mathbf{0}^{64}$ and $\mathbf{1}^{64}$ denote the all-zero and all-one 64-bit words ($\mathbf{1}^{64} = \texttt{0xFFFF...FFFF}$), not the integers 0 and 1. This single-word domain separation is sufficient to guarantee $K_1 \neq K_2$ for all master keys, enabling the SoP security argument.

Efficiency of single-word domain tagging. Using a single-word all-one domain tag in word 4 is more efficient than deriving K_1, K_2 via separate key-derivation function calls, and costs no extra memory or computation in the key-setup phase.

4.3. Block Function

For block counter $i \in \{0, 1, \ldots, 2^{63} - 1\}$ (usage is restricted to $[0, 2^{63})$ per Section 4.5), the *counter input* is the 320-bit value

$$X[i] = \mathbf{0}^{128} \| N \| \langle i \rangle_{64}, \tag{4}$$

viewed as words $(0,\ 0,\ n_0,\ n_1,\ i)$ where (n_0, n_1) are the two 64-bit LE halves of N. The keystream block is then

$$\mathrm{KS}[i] = \mathcal{P}_{12}(K_1 \oplus X[i]) \oplus \mathcal{P}_{12}(K_2 \oplus X[i]). \tag{5}$$

Expanding (5) using (2)–(4), the two Ascon-p_{12} inputs are

$$A[i] = K_1 \oplus X[i] = (k_0,\ k_1,\ k_2 \oplus n_0,\ k_3 \oplus n_1,\ i), \tag{6}$$
$$B[i] = K_2 \oplus X[i] = (k_0,\ k_1,\ k_2 \oplus n_0,\ k_3 \oplus n_1,\ \mathbf{1}^{64} \oplus i). \tag{7}$$

$A[i]$ and $B[i]$ differ *only* in word 4, making the two calls maximally domain-separated while sharing the expensive nonce-key mixing in words 2–3. In field terms, the 320-bit input comprises the 256-bit key region (words 0–3, with the 128-bit nonce XORed into words 2–3) and the 64-bit counter field (word 4); the nonce is a public tweak on the key, not a separate field, so the widths sum to exactly $256 + 64 = 320$ bits. For any fixed (K, N), enc is injective in the counter: word 4 equals i (resp. $\mathbf{1}^{64} \oplus i$) and uniquely identifies the block index.

4.4. Stream Encryption

Algorithm 1 gives the complete stream encryption procedure. Because SoP-CTR is a stream cipher, decryption is identical to encryption. Counter offset $i_0 > 0$ enables random access into the keystream.

Algorithm 1: SoP-CTR Stream Encryption

Input: $K \in \{0,1\}^{256}$, $N \in \{0,1\}^{128}$, $M \in \{0,1\}^*$, offset $i_0 \geq 0$
Output: C with $|C| = |M|$

1. $(K_1, K_2) \leftarrow \mathsf{KeySetup}(K)$; $i \leftarrow i_0$; pos $\leftarrow 0$
2. **while** pos $< |M|$ **do**
   (a) $\mathrm{KS}[i] \leftarrow \mathcal{P}_{12}(K_1 \oplus X[i]) \oplus \mathcal{P}_{12}(K_2 \oplus X[i])$
   (b) $\ell \leftarrow \min(320, |M| - \text{pos})$
   (c) $C[\text{pos} : \text{pos}+\ell] \leftarrow M[\text{pos} : \text{pos}+\ell] \oplus \mathrm{KS}[i][0 : \ell]$
   (d) pos $\leftarrow$ pos $+ \ell$; $i \leftarrow i + 1$
3. **return** C

The security proof (Theorem 1) assumes that all counter values i within each (K, N) session are pairwise distinct and drawn from $[0, 2^{63})$. Reusing a counter value under the same key and nonce directly exposes keystream, and querying a value $\geq 2^{63}$ breaks the proof's distinctness invariant. Implementations must ensure $i_0 + \lceil |M|/40 \rceil \leq 2^{63}$, and incrementing i by 1 per block is the standard way to satisfy both conditions. Using a fresh nonce N per session resets the counter and is always safe.

4.5. Parameters and Limits

Table 2 summarises the design parameters of SoP-CTR. The 256-bit key resists classical exhaustive search at 2^{256}. Its quantum margin is not the generic Grover level, because the Q1 attack discussed in Section 5.6 recovers the secret at $2^{85.3}$, and that figure rather than $\mathcal{O}(2^{128})$ is the post-quantum claim of this work.

Table 2. SoP-CTR parameter summary.

Parameter	Value
Key size	256 bits
Nonce size	128 bits
Block size	320 bits (40 bytes)
Max message length	$2^{63} \times 40$ bytes $\approx 3.7 \times 10^{20}$ B
Underlying perm.	Ascon-p_{12} (320-bit, 12 rounds)
Security claim	BBB, negligible advantage over the full 2^{63}-block counter range †

† Online BBB threshold $q \approx 2^{213}$ (for $p \leq 2^{42}$), capped far below by the 63-bit counter.

5. Security Analysis

This section establishes the security guarantees of SoP-CTR. We first define the security model, then state the distinctness property that underpins the analysis, motivate the need for BBB security with a concrete attack on single-call CTR, and finally prove the BBB bound via the H-coefficient technique (a standard proof method that bounds distinguishing advantage by comparing the probabilities of real and ideal transcripts).

5.1. Security Model

We analyze SoP-CTR in the *ideal-permutation model*, a standard abstraction in which the underlying permutation is treated as a uniformly random bijection that the adversary may query as a black box. In this model, Ascon-p_{12} is modeled as a uniformly random

permutation $\mathcal{P} : \{0,1\}^{320} \to \{0,1\}^{320}$. We use $\mathcal{P}$ for both the real Ascon-p_{12} and its ideal-model counterpart; the meaning is clear from context. Security is measured by the PRF advantage of a query-bounded distinguisher $\mathcal{D}$ in the ideal-permutation model that makes at most q queries. Theorem 1 is stated for a single key–nonce session (fixed public nonce N); Proposition 1 extends it to a single key used with many distinct nonces, where q denotes the total number of online queries summed over all sessions. The multi-user setting (independent keys) is outside the scope of this analysis.

Definition 3 (PRF Advantage). *Let $F_K : \{0,1\}^* \to \{0,1\}^*$ be a keyed function. The PRF advantage of $\mathcal{D}$ is*

$$\mathrm{Adv}_F^{\mathrm{prf}}(\mathcal{D}) = \left|\Pr[\mathcal{D}^{F_K(\cdot)} = 1] - \Pr[\mathcal{D}^{\$(\cdot)} = 1]\right|,$$

where K is uniform over the key space and $\$$ is a uniformly random function with the same domain/range.

5.2. Distinct Inputs Property

A prerequisite for the BBB analysis is that all permutation inputs across a query session are *distinct* [10]. Recall from (6)–(7) that for block i the two inputs are

$$A[i] = (k_0,\, k_1,\, k_2 \oplus n_0,\, k_3 \oplus n_1,\; i),$$
$$B[i] = (k_0,\, k_1,\, k_2 \oplus n_0,\, k_3 \oplus n_1,\; \mathbf{1}^{64} \oplus i).$$

Intra-block distinctness. $A[i] \neq B[i]$ because word 4 satisfies $i \neq \mathbf{1}^{64} \oplus i$ for all i (they differ in every bit).
Inter-block distinctness. For two distinct counters $i \neq j$, $A[i] \neq A[j]$ and $B[i] \neq B[j]$ trivially (different word 4). A cross-collision $A[i] = B[j]$ would require $i = \mathbf{1}^{64} \oplus j$, i.e., $j = \mathbf{1}^{64} \oplus i = \bar{i}$ (bitwise complement). For any counter $i \in [0, 2^{63})$, bit 63 of i is 0, so $\bar{i} = \mathbf{1}^{64} \oplus i$ has bit 63 equal to 1 and hence $\bar{i} \geq 2^{63}$, placing it outside $[0, 2^{63})$. Therefore, for any set of pairwise distinct counters from $[0, 2^{63})$, all $2q$ inputs $\{A[i_k]\} \cup \{B[i_k]\}$ are *pairwise distinct*. This distinct-input property is the structural prerequisite for the sum-of-PRPs theorem [10–12].

5.3. Necessity of BBB: A Concrete Attack on Single-Call CTR

Before proving that SoP-CTR achieves BBB security, we establish that birthday-bound security is *not* a conservative proof artifact. A concrete online-only distinguisher breaks any single-call CTR stream cipher at $q \approx 2^{n/2}$ queries.

Lemma 1 (Birthday distinguisher for single-call CTR). *Let $G_K(i) = P(K \oplus X[i])$ be a single-call CTR stream cipher built from an n-bit permutation P, where the encodings $X[i]$ are pairwise distinct across counters i. There exists a deterministic distinguisher $\mathcal{D}$ making q online queries and* no *offline queries with $\mathrm{Adv}_G^{\mathrm{prf}}(\mathcal{D}) \geq 1 - e^{-q^2/2^{n+1}}$. In particular,*

- *At $q = 2^{n/2}$, $\mathrm{Adv} \geq 1 - e^{-1/2} \approx 0.39$;*
- *At $q = 2^{(n+1)/2}$, $\mathrm{Adv} \geq 1 - e^{-1} \approx 0.63$.*

For $n = 320$ (Ascon-p_{12}), both thresholds lie at $q \approx 2^{160}$, i.e., approximately 2^{165} bytes of keystream under a single key.

Proof. The distinguisher and the advantage computation are given in Supplementary Material S2. □

Why the attack does not extend to SoP-CTR. Lemma 1 exploits the *injectivity* of single-call CTR output, which is a direct consequence of P being a bijection. This structural property vanishes in SoP-CTR: the output $P(A[i]) \oplus P(B[i])$ need not be injective, so the absence of collisions carries no distinguishing information. The birthday distinguisher therefore fails against SoP-CTR regardless of query count; the relevant security boundary is instead the BBB threshold of Theorem 1.

Figure 2 illustrates Lemma 1 on toy permutations of $n \in \{16, 24, 32\}$ bits over 400 independent trials each as a sanity check of the analytical prediction. This illustration does not constitute a validation for $n = 320$. The single-call CTR advantage matches the theoretical prediction $1 - e^{-q^2/2^{n+1}}$, reaching ≈ 0.39 at $q = 2^{n/2}$ and saturating near 1 beyond $q = 2^{n/2}$. SoP-CTR's advantage stays near zero throughout, because its outputs collide at the same rate as a truly random function.

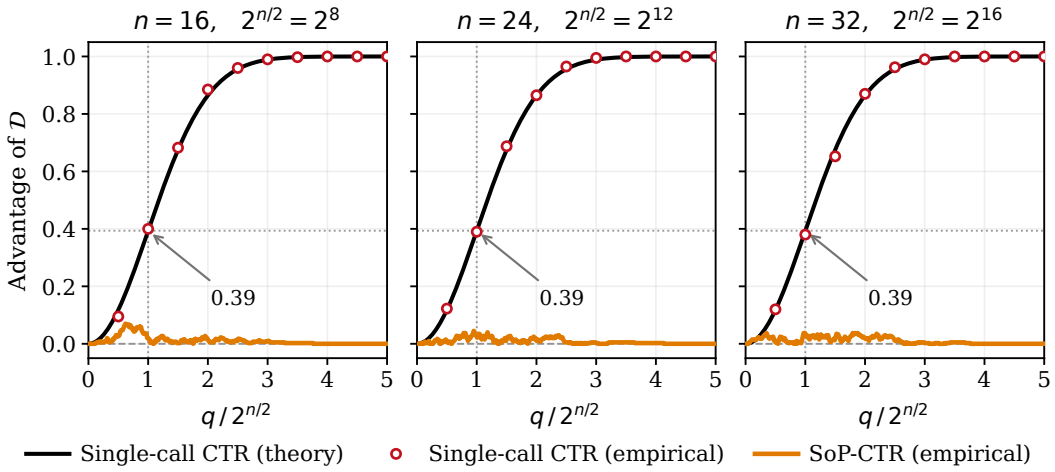


Figure 2. Advantage of birthday distinguisher $\mathcal{D}$ (Lemma 1) against single-call CTR and SoP-CTR. *Single-call CTR*: the solid line is the theoretical advantage $1 - e^{-q^2/2^{n+1}}$ and the open circles are the empirical values, which rise to ≈ 0.39 at the birthday threshold $q = 2^{n/2}$. *SoP-CTR* (orange): advantage near zero throughout. Each panel: 400 independent trials; $n \in \{16, 24, 32\}$.

5.4. Beyond-Birthday-Bound Security

In-range counter requirement. The construction is defined for counter values in $[0, 2^z)$. Suppose an adversary can submit two online queries with counter indices i and $\tilde{i} = i \oplus c$, where c is the split constant ($\tilde{i} = \bar{i}$ for SoP-CTR, since $c = \mathbf{1}^{64}$). Then $A[i] = B[\tilde{i}]$ and $B[i] = A[\tilde{i}]$, so the two block outputs XOR to zero, yielding a trivial distinguisher with advantage 1. Because bit z of c is set, $\tilde{i} = i \oplus c$ lies outside $[0, 2^z)$ whenever $i \in [0, 2^z)$, so confining all counters to $[0, 2^z)$ rules out this degenerate pair. The security proof formalises this as the hypothesis $i_k \in [0, 2^z)$ for all online queries $k \in [q]$, which deterministically ensures $\tilde{i}_k \notin [0, 2^z)$ and hence event B_3 cannot occur. Incrementing the counter internally so that the caller never supplies an arbitrary value is the recommended enforcement mechanism. Any allocation that keeps the counters pairwise distinct and within $[0, 2^z)$ is equally safe.

The proof of the main theorem invokes the mirror-theory count on the $2q$ permutation inputs of SoP-CTR. These inputs are not generic points of $\{0,1\}^n$ but all lie in a single 2^{64}-element coset, because each pair $(A[i], B[i])$ shares words 0 through 3 and differs only in word 4. The following lemma records that this coset structure does not weaken the count, and it is the formal device that closes the correlated-key concern.

Lemma 2 (Coset-invariance of the mirror-theory count)**.** *Let $P : \{0,1\}^n \to \{0,1\}^n$ be a uniformly random permutation, and let $\mathcal{C} \subseteq \{0,1\}^n$ be a fixed coset of a linear subspace, that is, a set of the form $\{v \oplus s : s \in S\}$ for a fixed offset v and a subspace S. Let $a_1, \ldots, a_q$ and $b_1, \ldots, b_q$ be elements of $\mathcal{C}$ such that the $2q$ points $\{a_k\} \cup \{b_k\}$ are pairwise distinct and disjoint from a set of p offline-queried points. For any targets $Z_1, \ldots, Z_q \in \{0,1\}^n$, the number of permutations consistent with $P(a_k) \oplus P(b_k) = Z_k$ for all k together with the p offline assignments is at least $(2^n - p)_{2q}\, 2^{-nq}\left(1 - q^{3/2}/2^n\right)$.*

Proof. Given in Supplementary Material S2. □

Theorem 1 (z-Split SoP Security)**.** *Let $P : \{0,1\}^n \to \{0,1\}^n$ be a permutation, κ the secret key width, z the split parameter, and c a split constant with bit z set to 1 (Definition 2). In the ideal-permutation model, any query-bounded distinguisher $\mathcal{D}$ making $q \le 2^z$ online PRF queries with pairwise distinct counter values $i_k \in [0, 2^z)$ and p offline queries to P or P^{-1} satisfies*

$$\mathrm{Adv}^{\mathrm{prf}}(q,p) \;\le\; \frac{q^{3/2}}{2^n} \;+\; \frac{2pq}{2^\kappa}.$$

The online BBB term $q^{3/2}/2^n$ remains below 1 for $q \lesssim 2^{2n/3}$; the offline term $2pq/2^\kappa$ is binding when $pq \gtrsim 2^{\kappa-1}$.

Proof sketch. The argument is an H-coefficient analysis [15] in four steps. Step 1 defines the transcript, comprising the q construction queries and the p offline permutation queries, augmented after the distinguisher halts with the two working keys, the usual strengthening device of the technique. Step 2 isolates four bad events, namely a collision inside either branch, a collision across the two branches, and a hit between an offline query and a branch input. The z-split confines every counter to $[0, 2^z)$, which makes the cross-branch event impossible outright, and the remaining three are bounded by birthday and hit probabilities. Step 3 bounds the good-transcript ratio below by $1 - q^{3/2}/2^n$ through the mirror-theory count, for which Lemma 2 supplies the coset invariance that the correlated keys require. Step 4 combines the two contributions into the stated bound. The full argument is given in Supplementary Materials S2. □

Corollary 1 (SoP-CTR Security)**.** *For SoP-CTR with $n = 320$, $\kappa = 256$, $z = 63$, and $c = \mathbf{1}^{64}$ in word 4:*

$$\mathrm{Adv}^{\mathrm{prf}}_{\mathsf{SoP\text{-}CTR}}(q,p) \;\le\; \frac{q^{3/2}}{2^{320}} \;+\; \frac{2pq}{2^{256}}, \qquad q \le 2^{63}.$$

The online BBB term stays negligible until $q \approx 2^{213}$ for a small offline budget $p \le 2^{42}$, far above the operational counter cap $q \le 2^{63}$. The offline term is binding when $pq \gtrsim 2^{255}$, so at a post-quantum offline budget $p = 2^{128}$ the effective threshold falls to $q \approx 2^{127}$.

Proposition 1 (Multi-nonce security under a single key)**.** *Theorem 1 fixes a single nonce N. The same bound, in the same form, holds for a single key K used with many* distinct *nonces $N^{(1)}, \ldots, N^{(d)}$ (a nonce-respecting adversary), provided each session uses pairwise distinct counters in $[0, 2^z)$. Here $q = \sum_s q_s$ is the* total *number of online queries over all sessions, which may exceed the per-session cap 2^z.*

The argument is structural. Under SoP-CTR, distinct nonces produce distinct word pairs 2–3, $(k_2 \oplus n_0^{(s)},\, k_3 \oplus n_1^{(s)})$, so every permutation input of one session differs in words 2–3 from every input of any other session. Combined with the per-session distinctness of Section 5.2, all $2q$ inputs across all sessions are pairwise distinct and each appears in exactly one XOR equation; hence the bad events of Theorem 1 still satisfy $\Pr[\mathsf{B}_1] = \Pr[\mathsf{B}_2] = \Pr[\mathsf{B}_3] = 0$ and $\Pr[\mathsf{B}_4] \le 2pq/2^\kappa$.

The mirror-theory count depends only on the $2q$ distinct inputs and their matching structure, not on the nonces, so the good-transcript ratio is again $1 - q^{3/2}/2^n$, giving

$$\mathrm{Adv}^{\mathrm{prf}}_{\mathit{SoP\text{-}CTR}}(q,p) \;\le\; \frac{q^{3/2}}{2^{320}} + \frac{2pq}{2^{256}}, \qquad q = \textstyle\sum_s q_s.$$

The 128-bit nonce space and the 2^{63} per-session counter range cap the total data under one key at $2^{128} \cdot 2^{63} = 2^{191}$ blocks, below the 2^{213} BBB threshold. Even at this structural maximum the online term is at most 2^{-33}, and it is far smaller at any realistic data volume; the offline term follows the small-p regime of Section 5.4. Nonce reuse with differing messages is excluded as the standard counter-mode failure that leaks the plaintext XOR (Section 5.6); the multi-user setting with independent keys is left to future work.

Tightness and the scope of the bound. The factor $q^{3/2}/2^n$ is the established security level for sums of public permutations in the ideal-permutation model, matching the $2n/3$-bit guarantee proved by Chen, Lambooij, and Mennink for the sum-of-Even-Mansour family [22]. It is worth delineating what this does and does not assert. In the *secret*-permutation model the sum of two permutations attains near n-bit security, with distinguishing advantage $O(q/2^n)$, as established through Patarin's mirror theory [11,21], the chi-squared analysis of Dai, Hoang, and Tessaro [32], and the tight variable-output reproof of Bhattacharya and Nandi [12]. Those results do not transfer to the present setting. Granting the adversary direct offline access to the public permutation is precisely what lowers the achievable guarantee from the n-bit regime to the $2n/3$ regime, and our bound is stated for this stronger, more realistic adversary. We therefore do not claim n-bit security. Conversely, we do not claim that $q^{3/2}/2^n$ is tight for the construction analysed here. SoP-CTR departs structurally from Even-Mansour-style sums, because the key is embedded inside the permutation input under a single master key rather than XORed as pre/post whitening, and we are not aware of a distinguisher that meets the $2^{2n/3}$ threshold against this variant. Establishing either a matching attack or an improved (up to n-bit) bound is left open, paralleling the open status of the same single-key variant in the quantum setting [16]. None of this affects the deployment-level conclusions, as Table 3 makes precise.

Structural limit on permutation size. Theorem 1 reveals a fundamental tension. Reaching the online BBB limit requires $z \ge 2n/3$. Definition 2 constrains $z < n - \kappa$, so $\kappa \le n/3 - 1$. Imposing post-quantum key security ($\kappa \ge 256$) therefore requires $256 \le \kappa \le n/3 - 1$, hence $n \ge 3\kappa + 3 \ge 771$. This condition $n \ge 3\kappa + 3$ describes the *framework* design space for a future permutation that would simultaneously reach the full online BBB limit and post-quantum key security ($\kappa = 256$). SoP-CTR itself does *not* satisfy it, since Ascon-p_{12} has $n = 320$ and $\kappa = 256$, giving $3\kappa + 3 = 771 \gg 320$. The instantiation therefore operates below the full-BBB structural ideal. The counter is restricted to $z = 63$ and the operational ceiling is $q \le 2^{63}$ per key–nonce pair, far below the theoretical online threshold 2^{213}. This restriction is not a weakness, as Table 3 shows. The 63-bit counter cap is a consequence of the parameter mismatch $n < 3\kappa + 3$, not merely a deployment choice. A future design targeting both online BBB and post-quantum key security at $\kappa = 256$ should use a permutation of at least $n = 771$ bits, for example, an 800-bit permutation such as Keccak-f[800], which admits $z = 543$ and a full online threshold near $q \approx 2^{533}$.

Operating regimes and dominant terms. The two terms of Theorem 1 dominate in different regimes, so a single threshold does not capture the security level. We therefore distinguish the *online degradation threshold*, namely the point $q \approx 2^{213}$ at which the online term $q^{3/2}/2^{320}$ alone approaches 1, from the *combined advantage*, which is negligible only when the offline term $2pq/2^{256}$ is also small. Table 3 reports the base-2 logarithm of each term across a range of (q, p), and shows that the online term is the binding constraint only

when $p \leq 2^{42}$. The headline figure $q \approx 2^{213}$ thus describes the online term in isolation and is meaningful only for adversaries with small offline permutation access, so it must not be read as a combined-advantage cap.

Table 3. Operating regimes: base-2 logarithm of each advantage term in Corollary 1. The online term $q^{3/2}/2^{320}$ is independent of p; the offline term $2pq/2^{256}$ grows with p. An entry "≥ 0" means that term reaches 1 (the bound is vacuous). The combined advantage is dominated by the larger of the two terms.

	Online	**Offline $\log_2(2pq/2^{256})$**		
q	$\log_2(q^{3/2}/2^{320})$	$p = 2^{42}$	$p = 2^{64}$	$p = 2^{128}$
2^{63}	-225.5	-150	-128	-64
2^{127}	-129.5	-86	-64	≥ 0
2^{160}	-80.0	-53	-31	≥ 0
2^{191}	-33.5	-22	≥ 0	≥ 0
2^{213}	-0.5	≥ 0	≥ 0	≥ 0

Furthermore, if an adversary queries i_k and $i_k \oplus c$ simultaneously, $A[i_k] = B[i_{k'}]$ and the XOR output is forced to 0, yielding a trivial distinguisher. Restricting all counter values to $[0, 2^{63})$ is therefore a *usage requirement* of SoP-CTR, not merely a recommendation.

AEAD composition (informal). SoP-CTR is intended to serve as the confidentiality component of an Encrypt-then-MAC AEAD scheme, paired with a strongly unforgeable message authentication code (MAC) to add strong unforgeability under chosen-message attack (SUF-CMA) on top of its BBB confidentiality under chosen-plaintext attack (IND-CPA). We sketch the expected interaction with the bound but do *not* prove it here. If an Ascon-p-based MAC is used in a black-box manner with an independent key, its internal permutation queries are bounded by the MAC's own invocation count and are not directly controlled by the distinguisher, so heuristically they do not enlarge the offline permutation budget p in Theorem 1. Turning this into a guarantee requires a formal composition proof that accounts for key separation, nonce binding, tag format, and any sharing of the permutation between the two components. We therefore treat the unchanged-bound statement as a conjecture and leave a complete AEAD proof to future work (Section 8.2).

5.5. Security Comparison

Table 4 compares the PRF-distinguishing thresholds of SoP-CTR against benchmark ciphers at the same Ascon-p state size. SoP-CTR's BBB term sustains PRF security 53 bits beyond the $q \approx 2^{160}$ point at which a hypothetical single-call CTR-Ascon baseline demonstrably loses birthday-bound security (Lemma 1), and 85 bits beyond Ascon-128's 2^{128} birthday limit, at the cost of two permutation calls per block instead of one. These are PRF-distinguishing thresholds, and SoP-CTR's own operational usage is capped at 2^{63} blocks per key–nonce pair (Section 5.4). The online column holds when the offline budget is small ($p \leq 2^{42}$). At a Grover-scale offline budget $p = 2^{128}$ the offline term becomes binding. For the 256-bit-key schemes SoP-CTR and CTR-Ascon it caps the threshold at roughly 2^{127} blocks, whereas the 128-bit-key Ascon modes instead saturate at $p \approx 2^{128}$, as the post-quantum column shows.

Table 4. PRF security comparison ($n = 320$-bit permutation state, $\kappa = 256$). All entries are theoretical PRF-distinguishing thresholds derived from the security bounds, not operational data caps. For SoP-CTR the operational cap is the counter range, which limits usage to 2^{63} blocks per key–nonce pair regardless of the 2^{213} distinguishing threshold. The online column holds when the offline budget is small ($p \leq 2^{42}$). The post-quantum column reports the offline-dominated threshold at a Grover-scale offline budget $p = 2^{128}$. For the 256-bit-key schemes the term $2pq/2^{256}$ becomes binding and caps q at about $2^{255}/p$, whereas the 128-bit-key Ascon modes use $2pq/2^{128}$ and saturate at $p \approx 2^{128}$.

Scheme	Bound	Theoretical Online Limit ($p \leq 2^{42}$, Blocks)	Theoretical PQ Offline Limit ($p = 2^{128}$, Blocks)
SoP-CTR (ours)	BBB	2^{213}	2^{127}
CTR-Ascon (1-call)	Birthday	2^{160}	2^{127}
Ascon-128	Birthday	2^{128}	Saturates ‡
Ascon-128a	Birthday	2^{96} †	Saturates ‡

† Ascon-128a has capacity $c = 192$ bits, giving birthday limit $2^{c/2} = 2^{96}$. ‡ The 128-bit-key Ascon modes use the offline term $2pq/2^{128}$, which saturates at $p \approx 2^{128}$. Here p counts queries to the public permutation, not keyed-cipher evaluations as in Grover key-search. PQ = post-quantum. BBB = beyond-birthday-bound.

5.6. Key and Nonce Security

The 256-bit master key provides 2^{256} classical key-search resistance. Generic quantum search reduces this to 2^{128} evaluations under Grover's algorithm [1], but a dedicated Q1 attack does better, as set out below. Nonce uniqueness is required, since a repeated nonce with different messages leaks the XOR of plaintexts, a standard CTR-mode vulnerability. The 128-bit nonce space makes random-nonce collision negligible for practical session counts.

Exhaustive key search. The offline term $2pq/2^{256}$ of Theorem 1 is precisely the key-guessing component of the bound, because p counts the adversary's own evaluations of the public permutation and 2^{256} is the size of the master-key space. Read at $q = 1$, the term reaches unity only at $p \approx 2^{255}$, which is exhaustive search up to a factor of two. Both branches are driven by the same master key, and the working keys K_1 and K_2 differ only by a public constant in word 4, so recovering one branch is recovering the entire key rather than half of it, and the two branches do not decompose into independently searchable halves. There is likewise no key schedule to invert, since the key enters only through the XOR $K_j \oplus X[i]$ that precedes a single fixed public permutation.

Tightness of the offline term. Exhaustive search is only the $q = 1$ corner of that term. Because the split constant c is public, writing $y = X[i] \oplus S$ for the secret mask S gives $\mathrm{KS}[i] = P(y) \oplus P(y \oplus c) =: D_c(y)$ with D_c a public map, so an adversary holding q blocks can match them against offline evaluations of D_c. The query variable reaches the nonce and the counter but never the state field carrying the upper key half, so the match guesses that field and resolves the remainder, and the mask falls once pq approaches 2^{256}, which is exactly where the offline term becomes vacuous. At $q = 1$ this is exhaustive search at $p \approx 2^{255}$, whereas at $q = 2^{63}$ it costs $p \approx 2^{193}$. We present it as a matching strategy rather than a theorem, since we give no formal reduction, but it indicates the offline term is essentially tight. The quantum attack described next beats generic Grover search by more than forty bits.

SoP-CTR is a stateless stream cipher and therefore provides no forward secrecy by itself; this is a standard limitation of keyed-PRF-based CTR modes [26] and can be addressed at the protocol layer. The online BBB bound is a *classical* distinguishing result, so we do not claim it as a quantum-distinguishing guarantee, and its quantum scope is discussed below. For a classical offline budget $p \leq 2^{42}$ the online BBB term is the bind-

ing constraint, while at a Grover-scale offline budget $p = 2^{128}$ the offline term becomes binding and caps usage at $q \approx 2^{127}$.

Quantum ideal-permutation model: scope of the proof. Theorem 1 is proved in the *classical* ideal-permutation model, where the adversary queries P and P^{-1} classically. Security established in idealised models does not automatically transfer to adversaries that make superposition queries to the primitive, as shown for the quantum random-oracle model by Boneh et al. [33]. We therefore do not claim the BBB distinguishing bound as a post-quantum guarantee. The closest available analysis is the recent study of the keyed sum of permutations and its siblings by Datta et al. [16], a preprint that is not yet peer-reviewed. That work separates two quantum models. In the Q2 model the adversary makes superposition queries to the keyed construction itself, which is not a realistic threat for a stream cipher that exposes only classical keystreams. In the realistic Q1 model, where construction queries are classical while the public permutation may be evaluated quantumly, those authors identify the domain-separated single-key single-permutation variant, which is exactly the variant instantiated by SoP-CTR, as an open direction.

Q1 security of SoP-CTR. We give the first analysis of that direction, bracketing the level from both sides, and we separate the two sides because only one is unconditional.

The upper side follows from the domain separator being public. Writing $y = X[i] \oplus S$ for the secret mask S, the keystream equals $D_c(y) = P(y) \oplus P(y \oplus c)$, which needs no secret to evaluate, so the two-call structure collapses to a single public map. The offline Simon technique of Bonnetain et al. [34], used for related targets by Sun et al. [35] and Li et al. [36], then recovers S under the trade-off $D \cdot T^2 = 2^s$, balancing at $T = 2^{s/3}$. Here s is the size of the secret rather than the permutation width, so for $s = 256$ this gives $2^{85.3}$ and not the $2^{n/3} \approx 2^{107}$ that the width alone would suggest. The lower side is an unconditional bound of $2^{64.3}$ Q1 queries, and closing the remaining gap needs a gentle-measurement argument that we state as a conjecture rather than a theorem. Both derivations, and the cost conventions they follow, are in Supplementary Materials S1.

Two consequences follow. The 2^{128} Grover margin of a 256-bit key is a generic key-search bound and not a proven floor here, so the post-quantum claim of this work is $2^{85.3}$. Theorem 1 itself is unaffected, since the attack is quantum and the classical offline effort remains 2^{256}.

Scope of the exposure, and how it could be removed. The attack above uses one structural feature and nothing else, namely that the branch separator is published, which is what makes D_c evaluable without the key. Any modification preserving a public separator therefore leaves the reduction intact, however elaborate the surrounding key derivation, whereas a separator that depends on the secret removes it. Two minimal changes illustrate the point. The first gives the two branches different key words, so that $A[i] = (K_{\mathrm{hi}}, K_{\mathrm{lo}} \oplus N, i)$ and $B[i] = (K_{\mathrm{lo}}, K_{\mathrm{hi}} \oplus N, \bar{\imath})$. The branch difference is then $(\delta h, \delta h, \mathbf{1}^{64})$ with $\delta h = K_{\mathrm{hi}} \oplus K_{\mathrm{lo}}$, which is secret and independent of nonce and counter. The second leaves the branch inputs in place and derives the separating constant from the key instead of publishing it. In both cases Theorem 1 is unchanged, because its hypothesis is pairwise distinctness alone, and in both cases D is no longer a public map, so the attack of this section does not apply. We deliberately do not state the level either variant reaches. Quantifying it requires analysis beyond the regime validated by the sources we rely on, and adopting either would invalidate the test vectors and require fresh hardware measurements. What this section establishes is therefore the boundary rather than a replacement design. The Q1 level of SoP-CTR is a property of the public separator, not of the SoP paradigm.

6. Implementation

We now turn from theory to practice and describe how SoP-CTR is realised on a representative IoT-class microcontroller. This section reports the target platform, the word-split implementation strategy on ARMv7-M, and the resulting code-size and stack-usage figures.

6.1. Target Platform

We implemented SoP-CTR on the ARM Cortex-M4F (STM32F407VGT6, 168 MHz, Thumb-2 instruction set, single-precision FPU, no cache). The STM32F407 is representative of the constrained IoT microcontrollers for which lightweight ciphers are designed.

6.2. ARMv7-M Word-Split Strategy

The Ascon-p_{12} state consists of five 64-bit words. The Cortex-M4 has only 32-bit general-purpose registers; a 64-bit word therefore occupies a *register pair* $(r_{\text{lo}}, r_{\text{hi}})$. We maintain the full 320-bit state as ten 32-bit registers throughout the permutation, avoiding memory spills for the hot path.

Each round of Ascon-p_{12} requires the following:

- Round constant addition (1 instruction per word pair): XOR into the low half of x_2.
- S-box layer (p_S): five 5-bit S-boxes applied bitwise across the state columns. Each S-box is a sequence of AND/OR/NOT operations over the ten register pairs, totalling approximately 33 instructions.
- Linear diffusion layer (p_L): two rotation-based 64-bit mixing operations per word. A 64-bit rotation by r bits on a register pair requires two `ROR`/`LSL`/`ORR` instructions in Thumb-2. The full diffusion layer costs approximately 40 instructions.

One round therefore requires approximately 74 instructions. Over 12 rounds, P_{12} costs $\approx$ 888 instructions. Two P_{12} calls per block give $\approx$ 1776 static instructions for the keystream generation. The measured cost is 62,916 cycles over 26 blocks $\approx$ 2420 cycles per 40-byte block, roughly 640 cycles above the static instruction count. This gap reflects an effective IPC near 0.73 rather than the single-cycle ideal, and is attributable to load/store latency on the register-pair spills at the P_{12} boundary, data dependencies inside the S-box layer that introduce pipeline stalls on the single-issue Cortex-M4 pipeline, and round-constant fetch overhead. The static instruction count and the measured cycle count are therefore consistent once these pipeline effects are accounted for.

6.3. Ascon-p Implementation and Inline Assembly

The open-source `ascon-c` [37] library provides word-split ARMv7-M assembly for Ascon-128, Ascon-128a, and Ascon-Hash. We reuse this `armv7m` back-end, which exposes the `P12ROUNDS` macro as a block of GCC inline assembly. SoP-CTR's block function calls `P12ROUNDS` twice, once for $A[i]$ and once for $B[i]$, then XORs the ten output word pairs to produce the 320-bit keystream block. Prior M4 benchmarks of Ascon [13] report throughput figures consistent with ours under comparable compilation settings.

6.4. Memory Footprint

Table 5 reports both dimensions of the on-device footprint. Flash figures are the `.text` sizes extracted from the linked ELF via `arm-none-eabi-nm`, compiled with `arm-none-eabi-gcc 13.2.1 -O2 -mcpu=cortex-m4 -mthumb`. Worst-case stack depth was obtained from the same ELF by inspecting the `PUSH`/`STMDB` and `SUB SP` instructions at each function prologue with `arm-none-eabi-objdump` and accumulating the call chain.

On the Flash side, the unrolled SoP-CTR permutation function (`sopctr_block_m4`, 7194 B) is larger than the Ascon-128 equivalent (3644 B) because it fully unrolls all

12 rounds in Thumb-2 word-split form. The rolled variant addresses this by placing one copy of Ascon-p_{12} in a callable function, `ascon_p12_call` at 3568 B, which `sopctr_block_m4_rolled` invokes twice via the `BL` instruction, eliminating the second inlined copy. The result is a 3832-byte total footprint, a 47.6% reduction (3476 B saved) relative to the unrolled variant, at a throughput cost of only 2.34 cpb (63.78 vs. 61.44 cpb).

Stack behaviour follows the same pattern. The rolled variant reduces the worst-case depth by 20 B (220 B vs. 240 B), because `sopctr_block_m4_rolled` saves only three registers (`r4`, `r5`, `lr`; 12 B) rather than nine (36 B), the callee-save burden having shifted to `ascon_p12_call` which manages its own register window. Both SoP-CTR M4 variants use less stack than scalar ChaCha20 (280 B), whose large local frame (244 B) accommodates the 512-bit working state of a 64-byte block alongside temporary variables. The Ascon-128 and Ascon-128a figures (80 B each) are substantially lower because their single-call permutation inlines the 320-bit state into ten general-purpose registers, leaving only 44 B of local frame for a partial-block buffer. By contrast, SoP-CTR must hold two domain-separated $\mathcal{P}_{12}$ input states (20 register-words = 80 B minimum) simultaneously, making the additional stack usage an inherent cost of the two-call SoP construction. All figures are compatible with the small per-task stack budgets common on embedded real-time operating systems, where the exact budget is application dependent.

Table 5. Memory footprint on Cortex-M4 (bytes).

	Flash			
Primitive	**Permutation**	**Cipher fn**	**Total**	**Stack**
SoP-CTR M4 unrolled	7194	114	7308	240
SoP-CTR M4 rolled	3568	264	3832	220
SoP-CTR C ref	11,260	180	11,440	248
ChaCha20 (scalar)	N/A	768	768	280
ChaCha20 (optimised)	2866	232	3098	276
Ascon-128	3644	1958	5602	80
Ascon-128a	3644	2596	6240	80

6.5. x86-64 AVX2 Implementation

The two $\mathcal{P}_{12}$ calls per SoP-CTR block are fully domain-separated, and consecutive blocks share no state. These properties enable a natural two-level SIMD parallelism on x86-64 processors with AVX2 (256-bit integer SIMD) support. Our AVX2 variant processes *two consecutive blocks per call* by packing four 320-bit Ascon-p_{12} states into five YMM registers. Each YMM register holds word k of four states across its four 64-bit lanes, namely $\text{ymm}_k = [A[i],\ A[i{+}1],\ B[i],\ B[i{+}1]]_k$. A single 12-round AVX2 permutation pass simultaneously advances all four states. The two output blocks are then recovered by lane-wise XOR, namely $\text{KS}[i] = \text{lane}_0 \oplus \text{lane}_2$ and $\text{KS}[i+1] = \text{lane}_1 \oplus \text{lane}_3$. This produces 80 bytes of keystream in one batched AVX2 $\mathcal{P}_{12}$ evaluation that processes the four packed states in parallel. Ascon sponge modes cannot benefit from this strategy because each absorption block depends on the previous sponge state.

7. Experimental Evaluation

This section reports our measurements of SoP-CTR on physical hardware and on a desktop-class processor. We describe the experimental setup, verify correctness against known-answer tests, and then compare throughput, length scaling, rounds-per-byte efficiency, and AVX2 acceleration against ChaCha20 and Ascon variants.

7.1. Experimental Setup

We ran benchmarks on an STM32F407VGT6 (ARM Cortex-M4F, 168 MHz, AHB bus, 1 MB Flash, 128 KB SRAM) mounted on an STM32F4 Discovery board. We measured cycle counts using the ARM Data Watchpoint and Trace (DWT) `CYCCNT` register, which increments on every core clock cycle. No interrupts were configured or enabled, and no real-time operating system (RTOS) or OS scheduler was present.

We compiled the firmware using `arm-none-eabi-gcc 13.2.1` at `-O2`, targeting Cortex-M4 with Thumb-2 instructions and hardware floating-point ABI. Each figure is the *median* cycle count over 64 measured runs (after 8 warmup iterations). Both platforms use the median of 64 runs as the primary statistic, providing a consistent methodology across the embedded and desktop evaluations.

We compare SoP-CTR against ChaCha20 [7] in two forms. The first is a portable RFC 8439 scalar reference and the second is an optimised Cortex-M4 assembly permutation from the ARMed SPHINCS implementation of Hülsing et al. [38], wrapped in the same stream framing. We also compare against Ascon-128 (P6 absorption, 8 B/call rate) and Ascon-128a (P8 absorption, 16 B/call rate) [13] in keystream (encryption-only, no tag) mode. All Ascon targets use the `armv7m` back-end from `ascon-c`.

7.2. Correctness Verification

Correctness is established by a known-answer test suite of ten keystream vectors that spans the relevant key, length, and counter cases. Appendix A gives one fully worked plaintext–ciphertext pair and summarises the suite, so that an implementer can reproduce SoP-CTR without running the reference code. The keys are an all-zero key, an incremental key (bytes 1–32 and 1–16), a non-trivial key (`de:ad:be:ef:00...` with nonce `ca:fe:00...`), and a repeated-pattern key (`ca:fe:ba:be` repeated, used for the 1 KiB vector). The message lengths are the empty string, a single byte, 39 bytes one short of a block, one 40-byte block, two 80-byte blocks, and 1 KiB. The counter offsets are 0, 5, and the upper boundary $i_0 = 2^{63} - 1$. Together these exercise partial-block handling, multi-block streaming, and the counter range. The golden keystream vectors are generated by the C reference and provided in the Supplementary Materials as `sopctr_kats.txt`. Every implementation reproduces these vectors byte-for-byte across the full suite: the x86-64 AVX2 variant on the host, and both the unrolled `sopctr_crypt_m4` and the rolled `sopctr_crypt_m4_rolled` variants on the physical STM32F407. As an internal consistency check, the seventh known-answer test (KAT) keystream, KAT-7, equals the KAT-1 keystream, and the KAT-5 and KAT-6 outputs are prefixes of it.

7.3. Throughput at 1 KiB Messages

Table 6 reports cycles and cycles-per-byte (cpb) for encrypting 1 KiB (1024 B) of data. The unrolled M4 implementation achieves 61.44 cpb, which is 2.11× faster than the C reference (129.32 cpb) and faster than Ascon-128 (64.39 cpb), the optimised `armv7m` lightweight standard. Against ChaCha20, SoP-CTR is within 44% of the portable scalar reference (42.76 cpb) but about 2.4× the cost of an optimised assembly ChaCha20 permutation (25.54 cpb) [38]. This gap reflects the two-permutation construction and is the price of the beyond-birthday 2^{213} margin that none of these birthday-bound ciphers provide (Section 5). The rolled variant achieves 63.78 cpb, a 3.8% penalty (2.34 cpb) relative to the unrolled version, where the overhead arises from two `BL/BX LR` round-trips per block, including callee-saved register `PUSH/POP` overhead (approximately 92 cycles out of roughly 2512 cycles per block). All figures in Table 6 are raw primitives, and the production hardening cost is discussed in Section 8.2. Because clock frequency and supply voltage are fixed across all measurements, energy consumption per byte is approximately proportional to

cycle count to first order, so the cpb figures in Table 6 serve as a first-order proxy for energy efficiency. We did not measure current or energy directly, so these figures are an estimate rather than a power measurement.

Table 6. Throughput on Cortex-M4, 1 KiB message (median of 64 runs, 8 warmup). Enc. = encryption only, no tag. Optimised = published Cortex-M4 assembly permutation [38]. cpb = cycles ÷ 1024, rounded to two decimals.

Primitive	Cycles	cpb
SoP-CTR M4 unrolled	62,916	61.44
SoP-CTR M4 rolled	65,308	63.78
SoP-CTR C ref	132,422	129.32
ChaCha20 (scalar ref)	43,789	42.76
ChaCha20 (optimised)	26,152	25.54
Ascon-128 (enc.)	65,938	64.39
Ascon-128a (enc.)	43,858	42.83

7.4. Cycles per Byte vs. Message Length

Table 7 reports cpb at four message lengths spanning two orders of magnitude. All stream ciphers converge to their steady-state cpb by 256 B. Both SoP-CTR M4 variants incur a 23% overhead at 64 B due to partial last-block handling when the message is not a multiple of 40 B. This overhead is negligible for messages longer than 256 B. The raw median cycle counts behind each entry are provided in the supplementary reproduction material.

Table 7. cpb vs. message length on Cortex-M4 (median of 64 runs, 8 warmup). SoP M4 = SoP-CTR unrolled (raw), SoP rol = rolled (raw), SoP ref = C reference, CC20 = ChaCha20 scalar ref, CC20o = ChaCha20 optimised asm, A128 = Ascon-128, A128a = Ascon-128a.

Length	SoP M4	SoP rol	SoP ref	CC20	CC20o	A128	A128a
64 B	74.06	76.94	159.22	44.59	27.05	79.64	58.08
256 B	65.55	68.06	139.19	43.13	25.84	67.44	45.88
1024 B	61.44	63.78	129.32	42.76	25.54	64.39	42.83
4096 B	60.91	63.22	128.07	42.67	25.46	63.62	42.06

7.5. Rounds-per-Byte Efficiency

Definition 4 (Rounds-per-Byte)**.** *For a stream cipher whose keystream primitive performs r_c rounds on a block of b bytes, the steady-state rounds-per-byte is* $\mathrm{r/B} = r_c/b$*. For constructions with two parallel calls (like SoP-CTR) the total round count is the sum across both calls.*

A cipher's *rounds-per-byte* (r/B) metric captures how many permutation rounds are consumed per byte of keystream produced. Among the Ascon-p-based schemes, SoP-CTR occupies an intermediate position at 0.60 r/B, as Table 8 collects side by side. Against Ascon-128, which runs at 0.75 r/B under a birthday bound, SoP-CTR uses 20% *fewer* rounds per byte and still provides BBB security, raising the block limit from 2^{128} to 2^{213}. Against Ascon-128a, which runs at 0.50 r/B under the same kind of bound, SoP-CTR spends 20% *more* rounds per byte, but in exchange the distinguishing limit rises from 2^{96} to 2^{213} blocks, a gain of 117 bits in PRF security.

Table 8. Rounds-per-byte vs. security bound (steady-state, long messages).

Cipher	Rounds	B/Call	r/B	PRF-Distinguishing Limit
SoP-CTR (ours)	2×12=24	40	0.60	BBB: 2^{213} blocks
Ascon-128	6	8	0.75	Bday: 2^{128} blocks
Ascon-128a	8	16	0.50	Bday: 2^{96} blocks
ChaCha20 *	20	64	0.31	Key: 2^{256}/2^{128}(quantum)

* ARX rounds, not directly comparable to Ascon rounds.

Figure 3 plots the same comparison, with rounds per byte on the horizontal axis and the PRF-distinguishing limit on the vertical axis, so that the preferred direction is down and to the left. SoP-CTR sits below and to the left of Ascon-128 on both axes at once, which is the concrete meaning of the claim that the beyond-birthday guarantee costs no throughput against that baseline. Ascon-128a reaches a lower rounds-per-byte figure, but only by enlarging the rate, which is exactly what lowers its distinguishing limit to 2^{96}.

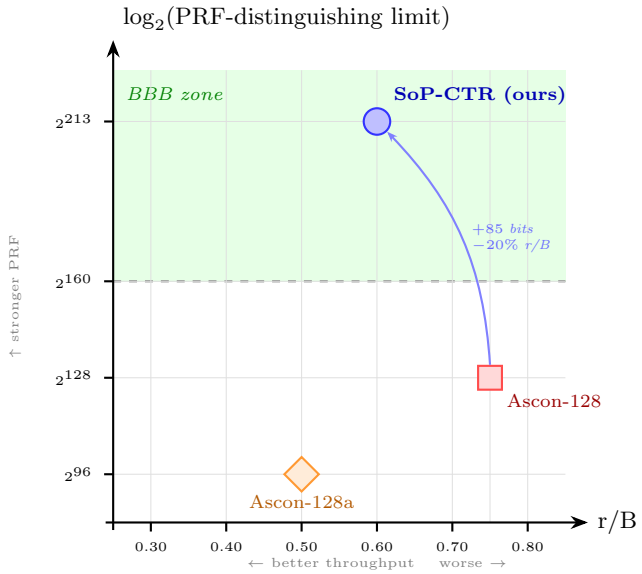


Figure 3. Rounds-per-byte vs. PRF-distinguishing limit (Ascon-p-based schemes). A lower r/B on the horizontal axis means better throughput, and a higher distinguishing limit on the vertical axis means stronger PRF security. SoP-CTR dominates Ascon-128 on both axes (BBB zone shaded).

7.6. x86-64 Throughput

Benchmarks were run on an Intel Core i7-10510U (Comet Lake, 4.9 GHz max single-core turbo, AVX2 support) under Windows 11 Pro (version 25H2, build 26200), compiled with MinGW GCC 13.2.0 at `-O3 -std=c11 -mavx2`, execution pinned to core 0 at high process priority. To stabilise the CPU turbo frequency, all ciphers were run continuously for 3 s before measurement, with the minimum processor state locked to 100% and boost mode set to aggressive (Windows `powercfg` equivalent of `cpupower -g performance`). Timing uses the `rdtsc/rdtscp + cpuid` fence technique. Each reported value is the *median* cpb over 64 timed runs after eight warmup iterations, averaged across three independent sessions. Between-session variation is approximately ±3% due to dynamic CPU frequency scaling and does not affect relative comparisons.

Table 9 compares all ciphers using plain C scalar implementations compiled with `-O3`. At 1024 B, SoP-CTR (5.01 cpb) is 21% faster than Ascon-128 (6.36 cpb) and carries a 21% throughput overhead relative to Ascon-128a (4.14 cpb), representing the cost of BBB security that Ascon-128a does not provide.

The last two rows of Table 9 isolate the effect of the two-block AVX2 acceleration. Ascon sponge modes are omitted because they are inherently sequential across blocks within a single message stream, so their two-block speedup is 1.0×. This concerns block-chaining within one stream and not multi-message batching, where independent streams could be packed across SIMD lanes for any cipher.

The speedup converges to ≈2× at 1024 B (1.95× measured), closely matching the theoretical 2× from processing two blocks per call. At 4096 B the AVX2 variant achieves 2.54 cpb, a 1.94× speedup over the scalar baseline, and the marginal difference from the 1024 B figure (2.57 cpb) lies within the ±3% between-session variance documented in Section 8.2, consistent with steady-state throughput. What the variant exploits is the structural block independence of the SoP construction, which chaining-based sponge modes cannot offer.

For context, a dedicated AVX2 implementation of ChaCha20 reaches about 1.43 cpb on Haswell [39], against 2.54 cpb at 4 KiB for SoP-CTR AVX2 on our Comet Lake core. The two figures come from different processors, so we read no precise ratio into them, but optimised vectorised ChaCha20 is clearly faster. That is expected rather than a deficiency of the implementation, since ChaCha20 is an ARX design whose quarter-round maps directly onto vector add-rotate-XOR instructions with high instruction-level parallelism, whereas SoP-CTR evaluates the bit-sliced Ascon permutation twice per block to obtain its beyond-birthday guarantee. The AVX2 result therefore demonstrates intra-cipher parallelism, not competitiveness with optimised ChaCha20.

Table 9. x86-64 throughput (cpb, MinGW GCC 13.2.0 `-O3`, median of 64 runs, 3 s thermal pre-warmup). Baselines are plain C scalar implementations.

Cipher	64 B	256 B	1024 B	4096 B
ChaCha20 (RFC 7539 ref)	4.08	3.46	3.30	3.26
Ascon-128 (P6, r = 64b)	6.78	6.41	6.36	6.35
Ascon-128a (P8, r = 128b)	4.62	4.23	4.14	4.12
SoP-CTR scalar	**6.62**	**5.42**	**5.01**	**4.92**
SoP-CTR AVX2 (2-block)	**3.66**	**3.23**	**2.57**	**2.54**

Bold marks the scheme proposed in this article.

8. Discussion

8.1. Deployment Scenarios

SoP-CTR is a composable keystream primitive intended as the bulk-encryption component of an AEAD scheme, composed with a SUF-CMA MAC via Encrypt-then-MAC (Section 5). The scenarios below describe this role.

Where the advantage actually lies. It is worth stating plainly what SoP-CTR does and does not buy over its closest baselines, because the beyond-birthday headline is easy to over-read. Against a hypothetical single-call CTR-Ascon, the BBB guarantee is a genuine asymptotic improvement, but its value at realistic data volumes is negligible. The 320-bit Ascon-p state places the single-call birthday threshold at $q \approx 2^{160}$ blocks (about 2^{165} bytes), unreachable by any device, and at the operational cap $q \leq 2^{63}$ both constructions already exhibit a distinguishing advantage below 2^{-190}. BBB security over CTR-Ascon should therefore be read as a formal assurance margin rather than a defence against a reachable attack, relevant chiefly where keys are long-lived and data accumulates under a single key across many nonces, as on aggregating gateways or long-lived record-layer connections.

Against the deployed Ascon-128 keystream mode, by contrast, the more consequential difference is not the birthday threshold but the key length. SoP-CTR carries a 256-bit master key, and although the Q1 attack of Section 5.6 lowers its quantum margin from

the generic 2^{128} to $2^{85.3}$, that still sits about twenty bits above the 2^{64} quantum key-search margin of the 128-bit Ascon modes. In a post-quantum deployment it is this key-search separation, not the keystream birthday bound, that materially distinguishes SoP-CTR. The BBB analysis then serves the complementary role of certifying that the doubled key length is not silently undone by a data-volume distinguisher in the keystream layer.

8.1.1. Constrained-Device Deployment

The rolled variant of SoP-CTR occupies 3832 bytes of Flash at 63.78 cpb on Cortex-M4, fitting within the budget of MQTT or CoAP sensor networks and OTA firmware update channels. At 168 MHz, SoP-CTR encrypts a 256 KB firmware image in approximately $\frac{256 \times 1024 \times 63.78}{168 \times 10^6} \approx 100$ ms. Because it derives from hardware DWT measurements on the physical STM32F407, this figure already incorporates the pipeline, Flash-latency, and accelerator behaviour of our bare-metal configuration with the ST Adaptive Real-Time (ART) accelerator enabled and no configured interrupts. Other configurations differ. An RTOS scheduler, a lower core frequency, or disabling the ART accelerator (which imposes seven wait states at 168 MHz) can raise encryption time by 20 to 40 percent, so practitioners should re-measure on their target before committing to timing budgets. The 3832-byte figure is for the keystream component only, and an Encrypt-then-MAC composition adds the MAC's own Flash and RAM.

8.1.2. Post-Quantum Migration in TLS-like Protocols

In record-layer protocols such as TLS 1.3 [6] or QUIC, the keystream primitive is architecturally decoupled from the handshake and authentication layers, which is exactly the role described above. Record layers are also the long-lived single-key accumulation regime identified there [3], so the beyond-birthday guarantee is of more than formal interest in this setting.

8.1.3. High-Throughput Edge and Gateway Servers

Gateway nodes aggregating encrypted telemetry may process sustained traffic under a shared session key, the aggregating single-key regime identified above. Here the x86-64 AVX2 two-block variant applies, at 2.57 cpb for 1 KiB messages (Section 7.6).

8.2. Limitations and Open Problems

8.2.1. Security Model Assumptions

The PRF security proof holds in the ideal-permutation model, treating Ascon-p_{12} as a random permutation. Security in the standard model against adversaries exploiting the concrete structure of Ascon-p_{12} is not addressed and remains open. A related concern is that the correlated keys K_1, K_2 place all $2q$ permutation inputs in a single 2^{64}-element coset rather than in generic position. This concern is resolved by Lemma 2, which shows that the mirror-theory count is invariant to the coset structure, so the bound of Theorem 1 holds without modification. The proof further requires the counter discipline specified in Section 4.3, namely pairwise-distinct counters confined to $[0, 2^{63})$, which caps usage at $q \leq 2^{63}$ per key–nonce pair. What invalidates the $B_3 = 0$ argument is a counter outside this range, in particular a complement pair i and $\bar{i} = i \oplus \mathbf{1}^{64}$ of which one member necessarily lies at or above 2^{63}, or else a reused counter. Either case may admit a trivial distinguisher.

8.2.2. Related-Key and Fault Attacks

Theorem 1 is a single-key result, so the adversary is given no oracle under any key other than the challenge key, and related-key security is not claimed. Two observations delimit the gap. First, the internal pair K_1, K_2 is itself a related-key pair with a fixed public difference in word 4, and that deliberate internal correlation is exactly what Lemma 2

covers. Second, an externally imposed key difference confined to words 2 and 3 is not a new attack surface, because the nonce is XORed into precisely those words. For any difference δ supported on words 2 and 3, the keystream generated under key $K \oplus \delta$ with nonce N equals the keystream generated under key K with nonce $N \oplus \delta$, so such a query is already available to a nonce-respecting adversary and is covered by Proposition 1. Key differences that touch words 0 or 1 fall outside the model, and a dedicated related-key analysis of Ascon-p_{12} under such differences remains open.

Fault attacks lie outside the formal model as well and call for countermeasures at the implementation level. The sum structure is specifically sensitive to a fault that suppresses or zeroes one of the two branches, because the surviving output is then a single permutation call on a publicly known input offset, and inverting the public permutation recovers the working key directly as $K_1 = P^{-1}(a) \oplus X[i]$. This is the same key-equivalence of the intermediate states noted in the side-channel discussion below. Deployments in physically exposed settings should therefore confirm that both branches actually executed, for example through redundant computation or a control-flow integrity check, in addition to the zeroisation already performed by the production library.

8.2.3. Threats Outside the Claimed Scope

Related-key security is not claimed. Theorem 1 is a single-key result, and key differences that touch words 0 and 1 cannot be simulated by nonce queries, so they fall outside the model. This scoping matches standard practice for nonce-based stream ciphers, where keys are assumed independent and no related-key oracle is available. Establishing related-key security would require both a related-key model for the construction and a dedicated analysis of Ascon-p_{12} under fixed input differences, neither of which is attempted here. Security against superposition queries to the construction is likewise not claimed, for the reason given in Section 5.6. Multi-key security is not analysed.

8.2.4. Implementation Scope

Benchmarks were conducted on a single Cortex-M4 device (STM32F407, 168 MHz) and one x86-64 laptop (Intel Core i7-10510U). Results may differ on other microcontroller unit (MCU) families, clock speeds, or compiler versions. The x86-64 comparison is confined to a single compilation environment (MinGW GCC 13.2.0, `-O3`). SIMD-optimised baselines for ChaCha20 and Ascon were not evaluated, so the AVX2 speedup figures reflect intra-cipher parallelism and do not constitute a cross-cipher SIMD comparison.

8.2.5. Benchmark Reproducibility

x86-64 measurements use dynamic CPU frequency scaling (turbo boost). Despite a 3 s thermal pre-warmup and fixed power settings, between-session variation is approximately $\pm 3\%$. Reported values are the median of 64 timed runs per session, averaged across three independent sessions. Within-session interquartile range (IQR) is ≤ 0.1 cpb for all ciphers at 1024 B. M4 measurements use the hardware DWT cycle counter in a bare-metal environment with no configured interrupts, providing deterministic single-run counts. All throughput tables report raw primitives without state zeroisation, matching the ChaCha20 and Ascon reference measurements. Absolute M4 counts carry an approximately 1% sensitivity to code layout and flash wait-states (removing one buffer wipe shifted a hot loop by 744 cycles). The production hardening cost is therefore measured within a single binary, where the code layout is fixed, rather than across separately compiled images.

8.2.6. AEAD Composition

SoP-CTR provides keystream confidentiality only. A complete AEAD scheme requires a separately specified MAC, tag format, and nonce binding. Their security composition is not proven here and is left to future work.

8.2.7. Security Bound Regime

The two terms of the PRF advantage bound play different roles, as Table 3 tabulates. The online term $q^{3/2}/2^{320}$ captures BBB resistance against online distinguishers and is independent of the offline budget, while the offline term $2pq/2^{256}$ captures key-recovery resistance and grows with p. Both must be small simultaneously, so applications assuming a large offline budget, for instance $p = 2^{128}$ under quantum key search, should verify that the operational data volume satisfies $q \ll 2^{127}$.

8.2.8. Nonce Uniqueness

SoP-CTR requires a unique nonce per encryption session under the same key, for the reason set out in Section 5.6. Secure nonce management, whether by random generation or a monotonic counter, is a deployment requirement that lies outside the scope of this paper.

8.2.9. Side-Channel Resistance

The Ascon-p_{12} permutation operates exclusively with bitwise logical operations and rotations, which are generally amenable to constant-time implementation. Neither the M4 assembly nor the C reference implementation has been formally verified for timing, power, or electromagnetic side-channel resistance. Deployments in adversarial physical environments should conduct independent side-channel analysis before reliance on these implementations. The two intermediate states $a = P(K_1 \oplus X)$ and $b = P(K_2 \oplus X)$ are key-equivalent. Since P is public and invertible, a disclosed a recovers the master key as $K_1 = P^{-1}(a) \oplus X$ with X public. The production library therefore zeroises a, b, and the keystream buffer after use, and clears the master-key context via `sopctr_clear`. Measured within a single binary, this word-wise zeroisation adds about 2.5% on the M4 (unrolled 61.44 → 62.97, rolled 63.78 → 65.56 cpb). On x86-64 the same zeroisation adds about 2.0% on the scalar path (5.01 → 5.11 cpb) and 5.4% on the two-block AVX2 path (2.58 → 2.72 cpb), where the larger fraction reflects the smaller absolute cost of the faster AVX2 baseline. The raw figure here is the twin-binary hardening baseline, one unit in the last place above the 2.57 of Table 9 and within the 0.01 run-to-run deviation. Even the hardened build remains faster than Ascon-128.

8.3. Future Work

Four directions follow from this work. First, a complete AEAD specification with a tag format, nonce binding, and a SUF-CMA MAC would turn SoP-CTR into a standalone authenticated cipher, replacing the Encrypt-then-MAC conjecture of Section 5 with a proven composition. Second, our Q1 analysis leaves a gap between the $2^{64.3}$ unconditional lower bound and the $2^{85.3}$ attack, and closing it requires a gentle-measurement argument for coset-adaptive resampling that does not yet exist in the literature. Third, the structural condition $n \geq 3\kappa + 3$ identifies an explicit design target for a future wider permutation that simultaneously reaches the online BBB threshold and the targeted post-quantum key-security level. Fourth, a masked or otherwise side-channel-hardened implementation, together with measured power and electromagnetic analysis, would extend the bare-metal timing results to physically adversarial deployments.

https://doi.org/10.3390/jcp6050157

9. Conclusions

We have presented SoP-CTR, a lightweight BBB stream cipher built from two domain-separated calls to a single Ascon-p_{12} permutation via the z-split SoP framework. The construction achieves PRF advantage $\mathrm{Adv}^{\mathrm{prf}}_{\mathsf{SoP\text{-}CTR}}(q,p) \leq q^{3/2}/2^{320} + 2pq/2^{256}$ with pairwise-distinct counters in $[0, 2^{63})$. The online term never binds operationally, since it remains below 2^{-225} over the 2^{63}-block per-key–nonce counter range and below approximately 2^{-33} even at the structural maximum of 2^{191} blocks accumulated across 2^{128} distinct nonces under one key. The offline term is the key-guessing component, and a meet-in-the-middle matching attains it. The term is therefore essentially tight, a conclusion we reach by argument rather than by formal reduction. In the Q1 quantum model we give the first analysis of this variant, with an attack at $2^{85.3}$ and an unconditional lower bound of $2^{64.3}$, so the post-quantum figure is the best known attack cost of $2^{85.3}$ rather than the generic 2^{128} Grover level.

On ARM Cortex-M4, our ARMv7-M word-split implementation achieves 61.44 cpb (unrolled) and 63.78 cpb (rolled, 3832 bytes Flash), outpacing Ascon-128 (64.39 cpb) on the same platform while providing a strictly stronger BBB guarantee. The two-block AVX2 variant achieves 2.57 cpb at 1 KiB via SIMD parallelism structurally unavailable to chained sponge modes.

The z-split framework implies $n \geq 3\kappa + 3$ for a future permutation to simultaneously reach the full online-BBB threshold and post-quantum key security at $\kappa \geq 256$, providing an explicit design target ($n \geq 771$ bits) for follow-on work. SoP-CTR is deployable as the confidentiality-only keystream component of IoT AEADs, TLS record layers, and firmware-update channels on Cortex-M4 class devices within a 3832-byte Flash budget. From a cybersecurity and privacy perspective, it adds a beyond-birthday-bound confidentiality component to the keystream layer of IoT and embedded protocols, complementing rather than replacing the existing cryptographic stack.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/jcp6050157/s1: Supplementary Material S1: Quantum Security in the Q1 Model; Supplementary Material S2: Proofs; Supplementary Material S3: Known-Answer Test Vectors; together with the accompanying source-code archive, containing the C reference, ARMv7-M assembly and AVX2 implementations, the benchmark scripts, a README, a REPRODUCTION guide, a LICENSE file and a SHA256SUMS manifest. All references cited in the Supplementary Materials also appear in the reference list of the main text: S2 cites [11,12,15,21,22], and S1 cites [16,34–36]. S3 contains no references.

Author Contributions: Conceptualisation, A.A.L., S.M., K.R., S.W. and T.S.G.; methodology, A.A.L., S.M., K.R. and T.S.G.; software, A.A.L. and S.W.; validation, A.A.L., S.W. and T.S.G.; formal analysis, A.A.L. and S.M.; investigation, A.A.L.; writing—original draft preparation, A.A.L.; writing—review and editing, A.A.L., S.M., K.R., S.W. and T.S.G.; visualisation, A.A.L.; supervision, K.R., S.M. and T.S.G.; project administration, A.A.L. and K.R.; funding acquisition, K.R. All authors have read and agreed to the published version of the manuscript.

Funding: This publication was supported by Universitas Indonesia through the Hibah Publikasi Terindeks Internasional (PUTI) Q1 Scheme under Contract PKS-252/UN2.RST/HKP.05.00/2025 and through the Innovation Program under Contract PKS-566/UN2.INV/HKP.05/2023. The work of A.A.L. was supported by Lembaga Pengelola Dana Pendidikan (LPDP), Ministry of Finance of the Republic of Indonesia under Contract SKPB-4400/LPDP/LPDP.3/2023.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The implementation source code (C reference, ARMv7-M assembly, and AVX2 variants) and benchmark scripts supporting the results of this study are available as Supplementary Materials accompanying this article, released under the MIT License. The archive carries a `SHA256SUMS` manifest for integrity checking.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Test Vectors

This appendix gives one fully worked plaintext–ciphertext pair and a summary of the ten known-answer tests, so that an implementer can reproduce SoP-CTR without running the reference code. All values are hexadecimal, byte order as transmitted.

Appendix A.1. Worked Example

Take the 256-bit key and 128-bit nonce below, and start at counter $i_0 = 0$. The first keystream block is $\mathrm{KS}[0] = \mathcal{P}_{12}(K_1 \oplus X[0]) \oplus \mathcal{P}_{12}(K_2 \oplus X[0])$, the 40-byte ASCII message is `SoP-CTR test vector, block zero example.`, and the ciphertext is $C = M \oplus \mathrm{KS}[0]$. Each value is one unbroken hexadecimal string.

key	0102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f20
nonce	0102030405060708090a0b0c0d0e0f10
keystream	4416d842ba801d34d9636c44035233c04d76a4a4cdf77bdb4dd9c75d120cd9bd6c743cabff5f171e
plaintext	536f502d43545220746573742076656374f722c20626c6f636b207a65726f206578616d706c652e
ciphertext	1779886ff9d44f14ad061f30232456a33919d688ed9517b42eb2e727777eb69d090c5dc68f337230

Decryption is the same operation, and XORing C with $\mathrm{KS}[0]$ returns M exactly. The message is one block long, so no counter increment occurs.

Appendix A.2. Known-Answer Tests

Ten known-answer vectors cover the empty message, single-byte and sub-block lengths, exact and multiple block lengths, a 1 KiB message, a non-zero starting counter, and the boundary case $i_0 = 2^{63} - 1$, which is the last admissible counter value under the usage requirement of Section 4.5. The vectors are listed in full in Supplementary Materials S3, together with the generator that produced them and the cross-checks that verify the AVX2 and Cortex-M4 back-ends against the same frozen constants.

References

1. Grover, L.K. A Fast Quantum Mechanical Algorithm for Database Search. In Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC 1996), Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219. [CrossRef]
2. Bellare, M.; Impagliazzo, R. A Tool for Obtaining Tighter Security Analyses of Pseudorandom Function Based Constructions, with Applications to PRP to PRF Conversion. IACR Cryptol. ePrint Arch. Paper 1999/024, 1999. Available online: https://eprint.iacr.org/1999/024 (accessed on 4 May 2026).
3. Luykx, A.; Paterson, K.G. Limits on Authenticated Encryption Use in TLS. Cryptol. ePrint Arch. Paper 2024/051, 2024. Available online: https://eprint.iacr.org/2024/051 (accessed on 4 May 2026).
4. Dworkin, M.J. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. In *Technical Report NIST Special Publication 800-38D*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2007. [CrossRef]
5. Degabriele, J.P.; Govinden, J.; Günther, F.; Paterson, K.G. The Security of ChaCha20-Poly1305 in the Multi-User Setting. In *CCS '21: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*; ACM: New York, NY, USA, 2021; pp. 1981–2003. [CrossRef]
6. Rescorla, E. *The Transport Layer Security (TLS) Protocol Version 1.3*; RFC 8446; Internet Engineering Task Force (IETF): Wilmington, DE, USA, 2018. [CrossRef]
7. Bernstein, D.J. ChaCha, a variant of Salsa20. In Proceedings of the Workshop on the State of the Art of Stream Ciphers (SASC 2008), Lausanne, Switzerland, 13–14 February 2008.
8. Öztürk, G.; Koçak, O.; Yayla, O. HCTR^{++}: A Beyond Birthday Bound Secure HCTR2 Variant. *Cryptol. ePrint Arch.* Paper 2026/383, 2026. Preprint, Not Yet Peer-Reviewed. Available online: https://eprint.iacr.org/2026/383 (accessed on 4 May 2026).

9. Crowley, P.; Huckleberry, N.; Biggers, E. Length-preserving encryption with HCTR2. *Cryptol. ePrint Arch.* Paper 2021/1441, 2021. Available online: https://eprint.iacr.org/2021/1441 (accessed on 4 May 2026).
10. Lucks, S. The Sum of PRPs Is a Secure PRF. In *Proceedings of the Advances in Cryptology—EUROCRYPT 2000*; Preneel, B., Ed.; Lecture Notes in Computer Science; Springer: Berlin, Heidelberg, 2000; Volume 1807, pp. 470–484. [CrossRef]
11. Patarin, J. Introduction to Mirror Theory: Analysis of Systems of Linear Equalities and Linear Non Equalities for Cryptography. *IACR Cryptol. ePrint Arch.* Paper 2010/287, 2010. Available online: https://eprint.iacr.org/2010/287 (accessed on 4 May 2026).
12. Bhattacharya, S.; Nandi, M. Revisiting Variable Output Length XOR Pseudorandom Function. *IACR Trans. Symmetric Cryptol.* **2018**, *2018*, 314–335. [CrossRef]
13. Dobraunig, C.; Eichlseder, M.; Mendel, F.; Schläffer, M. Ascon v1.2: Lightweight Authenticated Encryption and Hashing. *J. Cryptol.* **2021**, *34*, 33. [CrossRef]
14. Turan, M.S.; McKay, K.; Kang, J.; Kelsey, J.M.; Chang, D. *Ascon-Based Lightweight Cryptography Standards for Constrained Devices: Authenticated Encryption, Hash, and Extendable Output Functions*; NIST SP 800-232; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2025. [CrossRef]
15. Patarin, J. The "Coefficients H" Technique. In *Proceedings of the Selected Areas in Cryptography (SAC 2008)*; Avanzi, R., Keliher, L., Sica, F., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5381, pp. 328–345. [CrossRef]
16. Datta, N.; Dutta, A.; Mandal, S.; Nandi, H.; Sinha, A. Post-Quantum Security of Keyed Sum of Permutations and Its Siblings. *Cryptol. ePrint Arch.* Paper 2026/426, 2026. Preprint, Not Yet Peer-Reviewed. Available online: https://eprint.iacr.org/2026/426 (accessed on 4 May 2026).
17. Hell, M.; Johansson, T.; Meier, W. Grain: A Stream Cipher for Constrained Environments. *Int. J. Wirel. Mob. Comput.* **2007**, *2*, 86–93. [CrossRef]
18. De Cannière, C.; Preneel, B. Trivium. In *New Stream Cipher Designs: The eSTREAM Finalists*; Robshaw, M., Billet, O., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2008; Volume 4986, pp. 244–266. [CrossRef]
19. Dobraunig, C.; Eichlseder, M.; Mangard, S.; Mendel, F.; Mennink, B.; Primas, R.; Unterluggauer, T. ISAP v2.0. *IACR Trans. Symmetric Cryptol.* **2020**, *2020*, 390–416. [CrossRef]
20. Bertoni, G.; Daemen, J.; Hoffert, S.; Peeters, M.; Van Assche, G.; Van Keer, R. Farfalle: Parallel Permutation-Based Cryptography. *IACR Trans. Symmetric Cryptol.* **2017**, *2017*, 1–38. [CrossRef]
21. Patarin, J. Mirror Theory and Cryptography. *Appl. Algebra Eng. Commun. Comput.* **2017**, *28*, 321–338. [CrossRef]
22. Chen, Y.L.; Lambooij, E.; Mennink, B. How to Build Pseudorandom Functions from Public Random Permutations. In *Proceedings of the Advances in Cryptology—CRYPTO 2019*; Boldyreva, A., Micciancio, D., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2019; Volume 11693, pp. 266–293. [CrossRef]
23. Sibleyras, F.; Todo, Y. Keyed Sum of Permutations: A Simpler RP-Based PRF. In *Proceedings of the Topics in Cryptology—CT-RSA 2023*; Rosulek, M., Ed.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2023; Volume 13871, pp. 573–593. [CrossRef]
24. Mennink, B. Minimized PRFs from Public Permutations. *IACR Trans. Symmetric Cryptol.* **2025**, *2025*, 230–260. [CrossRef]
25. Hamann, M.; Krause, M. On Stream Ciphers with Provable Beyond-the-Birthday-Bound Security against Time-Memory-Data Tradeoff Attacks. *Cryptogr. Commun.* **2018**, *10*, 959–1012. [CrossRef]
26. Chen, Y.L.; Luykx, A.; Mennink, B.; Preneel, B. Systematic Security Analysis of Stream Encryption with Key Erasure. *IEEE Trans. Inf. Theory* **2021**, *67*, 7518–7534. [CrossRef]
27. Iwata, T. New Blockcipher Modes of Operation with Beyond the Birthday Bound Security. In *Proceedings of the Fast Software Encryption (FSE 2006)*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4047, pp. 310–327. [CrossRef]
28. Beierle, C.; Biryukov, A.; Cardoso dos Santos, L.; Großschädl, J.; Perrin, L.; Udovenko, A.; Velichkov, V.; Wang, Q. Lightweight AEAD and Hashing using the Sparkle Permutation Family. *IACR Trans. Symmetric Cryptol.* **2020**, *2020*, 208–261. [CrossRef]
29. Banik, S.; Chakraborti, A.; Iwata, T.; Minematsu, K.; Nandi, M.; Peyrin, T.; Sasaki, Y.; Sim, S.M.; Todo, Y. GIFT-COFB. Cryptol. ePrint Arch. Paper 2020/738, 2020; Finalist, NIST Lightweight Cryptography Standardization. Available online: https://eprint.iacr.org/2020/738 (accessed on 4 May 2026).
30. Iwata, T.; Khairallah, M.; Minematsu, K.; Peyrin, T. Duel of the Titans: The Romulus and Remus Families of Lightweight AEAD Algorithms. *IACR Trans. Symmetric Cryptol.* **2020**, *2020*, 43–120. [CrossRef]
31. Turan, M.S.; McKay, K.; Chang, D.; Bassham, L.E.; Kang, J.; Waller, N.D.; Kelsey, J.M.; Hong, D. *Status Report on the Final Round of the NIST Lightweight Cryptography Standardization Process*; NIST IR 8454; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023. [CrossRef]
32. Dai, W.; Hoang, V.T.; Tessaro, S. Information-Theoretic Indistinguishability via the Chi-Squared Method. In *Proceedings of the Advances in Cryptology–CRYPTO 2017*; Katz, J., Shacham, H., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; Volume 10403, pp. 497–523. [CrossRef]

33. Boneh, D.; Dagdelen, Ö.; Fischlin, M.; Lehmann, A.; Schaffner, C.; Zhandry, M. Random Oracles in a Quantum World. In *Proceedings of the Advances in Cryptology–ASIACRYPT 2011*; LNCS; Springer: Berlin/Heidelberg, Germany, 2011; Volume 7073, pp. 41–69. [CrossRef]
34. Bonnetain, X.; Hosoyamada, A.; Naya-Plasencia, M.; Sasaki, Y.; Schrottenloher, A. Quantum Attacks without Superposition Queries: The Offline Simon's Algorithm. In *Proceedings of the Advances in Cryptology—ASIACRYPT 2019*; Galbraith, S.D., Moriai, S., Eds.; Springer: Cham, Switzerland, 2019; Lecture Notes in Computer Science; Volume 11921, pp. 552–583. [CrossRef]
35. Sun, H.W.; Gao, F.; Xu, R.X.; Li, D.D.; Li, Z.Q.; Zhang, K.J. Quantum Key-Recovery Attacks on Permutation-Based Pseudorandom Functions. *Cryptol. ePrint Arch.* Paper 2025/501, 2025. Preprint, Not Yet Peer-Reviewed. Available online: https://eprint.iacr.org/2025/501 (accessed on 4 May 2026).
36. Li, Z.; Fan, S.; Gao, F.; Hao, Y.; Sun, H.; Hu, X.; Li, D. Quantum Attacks on Sum of Even-Mansour Construction Utilizing Online Classical Queries. *Cryptol. ePrint Arch.* Paper 2025/472, 2025. Preprint, Not Yet Peer-Reviewed. Available online: https://eprint.iacr.org/2025/472 (accessed on 4 May 2026).
37. Dobraunig, C.; Eichlseder, M.; Mendel, F.; Schläffer, M. ascon-c: Ascon Reference and Optimized Implementations in C. 2024. Available online: https://github.com/ascon/ascon-c (accessed on 4 May 2026).
38. Hülsing, A.; Rijneveld, J.; Schwabe, P. ARMed SPHINCS – Computing a 41 KB Signature in 16 KB of RAM. In *Proceedings of the Public-Key Cryptography–PKC 2016*; LNCS; Optimised Cortex-M ChaCha permutation; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9614, pp. 446–470. Available online: https://github.com/joostrijneveld/chacha-arm-cortex-m (accessed on 4 May 2026). [CrossRef]
39. Goll, M.; Gueron, S. Vectorization of ChaCha Stream Cipher. In *Proceedings of the 11th International Conference on Information Technology: New Generations (ITNG 2014)*; Full Version: Cryptology ePrint Archive, Paper 2013/759; IEEE: Piscataway, NJ, USA, 2014.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.