

Cryptographic Permutation-Based Primitives: Constructing Large, Secure and Efficient Building Blocks for Next-Generation Symmetric Cryptography

Linh Hoang Dinh, Luong Tran Thi*, Long Nguyen Van

Vietnam Government Information Security Commission, Hanoi, Vietnam

Abstract: In the era of big data, cloud computing, and the Internet of Things (IoT), the demand for secure and efficient cryptographic solutions has significantly increased. Permutation-based cryptography has emerged as a versatile and robust paradigm for next-generation symmetric cryptography, offering provable security, high performance, and resistance against both classical and quantum attacks. This paper provides a comprehensive survey of permutation-based cryptography, focusing on public large-state permutations (greater than 256 bits) to meet the demands of post-quantum security and high-throughput applications. We explore key structures such as the Sponge construction (used in SHA-3), the Duplex construction (for authenticated encryption schemes such as Ascon), and the Farfalle construction (enabling parallel processing in pseudorandom functions, e.g., Kravatte), highlighting their security guarantees and efficiency trade-offs. In addition, we analyze the core components of cryptographic permutations, including nonlinear layers (S-boxes and modular addition) and linear diffusion layers (MDS matrices and Column Parity Mixers (CPMs)), together with common design strategies such as Feistel networks, Substitution–Permutation Networks (SPNs), and Addition–Rotation–XOR (ARX) structures. A comparative study of notable permutations, including Keccak-f, Ascon-p, Xoodoo, Gaston, and Gimli, demonstrates their applications in hashing, authenticated encryption, message authentication codes, and pseudorandom number generation. Finally, we discuss emerging challenges related to standardization, quantum resistance, and performance optimization for future cryptographic designs. This work serves as a reference for researchers and practitioners seeking to understand and advance permutation-based cryptography in the post-quantum era.

Keywords: Cryptographic Permutations, Permutation-based Cryptography, Sponge construction, Post-quantum security, Authenticated Encryption, Hash functions, Lightweight Cryptography

1. INTRODUCTION

Today, the exponential growth of data, the ubiquity of cloud computing, and the proliferation of Internet of Things (IoT) devices are placing increasingly stringent demands on cryptographic solutions that are both secure and efficient [2, 67, 109]. The massive volume of data requires cryptographic algorithms capable of handling high-throughput workloads without compromising security. In cloud environments, secure computation over outsourced data is essential, as demonstrated by homomorphic encryption schemes [32, 91]. IoT devices, which are constrained by limited resources such as memory, processing power, and energy, necessitate lightweight cryptographic solutions [59, 191], often relying on encryption modes that do not require an inverse decryption function (e.g., CTR and OFB modes, which use only the block cipher encryption function E_K for both encryption and decryption).

*Corresponding author: luongtran@actvn.edu.vn

The advent of quantum computing presents a significant threat to current public-key cryptography (e.g., RSA and ECC), as quantum algorithms such as Shor's algorithm can efficiently solve the underlying mathematical problems far faster than classical computers [2, 67, 109]. Even symmetric-key algorithms are not immune, as Grover's algorithm can effectively halve their security level in terms of key length [78, 105, 141, 176]. This has led to the strategy known as "Harvest Now, Decrypt Later" (HNDL), where adversaries collect encrypted data today with the aim of decrypting it using future quantum computers [162, 181]. The convergence of these challenges—from massive-scale data processing and resource-constrained environments to the looming quantum threat—necessitates a fundamental re-evaluation of cryptographic primitives. Traditional block ciphers, while robust, often lack the flexibility or performance configurability required to address the wide spectrum of emerging application contexts.

1.1. Challenges of Traditional Block Cipher-Based Primitives

Until recent years, symmetric cryptography has been predominantly shaped by block ciphers. Except for a few dedicated stream ciphers, most standards and commercial cryptographic systems for encryption and authenticated encryption have relied on block ciphers. For example, the Advanced Encryption Standard (AES) [54] is the most widely adopted block cipher algorithm, and even hash functions or pseudorandom bit generators have been constructed using block cipher-based modes of operation [12, 102].

However, as cryptographic research continues to advance, several challenges associated with block cipher-based designs have come to light. In [46], J. Daemen identified fundamental issues inherent to block cipher-based cryptography that are difficult to resolve.

The first challenge concerns the security proofs of block cipher modes of operation. In practice, proving the security of such modes is often nontrivial and difficult to verify, and several errors have been identified in existing proofs. For instance, the OCB mode, known for its efficiency, was originally proven secure [6] with no known flaws in its proof. However, a standardized variant, OCB2, was later found not to satisfy key assumptions required for the original proof, rendering the proof invalid [101]. This flaw went unnoticed for nearly a decade due to the subtle and hard-to-detect nature of the assumption violations. A more recent example is the GCM-SIV mode, whose original security proof was later found to be flawed. While the authors attributed the issue to a "typographical error," the reality was that the proof overstated the achieved security level [103]. These examples demonstrate that proving and validating the security of block cipher modes is both complex and error-prone, even for expert readers or reviewers.

A second issue noted by Daemen is the so-called birthday bound, which limits the security of block ciphers when the amount of processed data approaches $2^{n/2}$ blocks, where n is the block size. While the security of block ciphers is often measured by the key size k , with exhaustive key search requiring 2^k operations, the block size n also plays a crucial role in determining the maximum amount of data that can be securely encrypted under a single key. In most block cipher modes (such as CBC, CTR, GCM, and OCB), the security guarantees no longer hold once the number of encrypted blocks exceeds $2^{n/2}$ due to the birthday paradox. For a block size of 64 bits, this threshold corresponds to only 32 GB of data, an amount easily reached in practical systems. Beyond this limit, security assurances begin to degrade, prompting the need for key rotation. For example, the older DES standard ($n = 64$) and the modern AES standard ($n = 128$) have respective birthday bounds of 2^{32} blocks (approximately 32 GB) and 2^{64} blocks.

A third challenge emphasized by Daemen concerns efficiency. In practice, performance plays a central role in determining the deployability of cryptographic algorithms. In block cipher design, efficiency must be ensured not only for encryption but also for decryption. However, many modern block cipher modes do not use the decryption function at all, resulting in wasted implementation effort. Examples of such modes include CTR, OFB, and

CFB, as well as MAC constructions based on block ciphers. Moreover, NIST hash function standards such as SHA-1 and SHA-2 are built using block cipher-based constructions but do not use the cipher's inverse mapping. This raises a fundamental design question: when the desired cryptographic primitive is a pseudorandom function (PRF), as is sufficient for many applications such as CTR mode, why insist on designing a full permutation (PRP), which includes reversibility?

Additionally, designing a secure and efficient key schedule for block ciphers remains a challenging task. In traditional block cipher architectures, inputs of size $n + k$ bits (with k the key size and n the block size) are processed asymmetrically: the key is transformed through the key schedule, while the data passes through the round function. This separation limits potential diffusion from the data path to the key schedule and may reduce the overall mixing strength. In light of these issues, permutation-based cryptography has emerged as a promising alternative approach.

1.2. Provable Security

Permutation-based cryptography offers an attractive model in which cryptographic schemes can be proven secure based on the properties of a pseudorandom permutation (PRP) or a public random permutation [5, 20–22]. This analytical framework simplifies security evaluation by shifting the focus toward the design and analysis of a single, strong permutation rather than multiple separate primitives. The threat posed by Harvest Now, Decrypt Later (HNDL) attacks [162, 181] implies that data encrypted today must remain secure well into the future, even before the actual advent of quantum computers. This significantly elevates the importance of provable security. If a scheme's security relies solely on heuristic arguments or incomplete cryptanalysis, confidence in its long-term resilience—especially against unknown quantum attacks—remains low. As such, an emphasis on provable security in permutation-based cryptography becomes critical for ensuring long-term assurance. It transforms the design paradigm from ad-hoc constructions to mathematically rigorous frameworks, providing a stronger foundation for defending against unforeseen attack vectors, including those introduced by quantum adversaries.

1.3. Resistance to Cryptanalysis

Well-designed permutations, particularly those with large internal state sizes, can offer strong resistance against a wide range of cryptanalytic attacks, including differential [33], linear [144], higher-order differential [120], zero-sum, and invariant subspace attacks [13, 44, 123]. The iterative nature of such designs, combined with the careful selection of nonlinear and linear layers, contributes to rapid diffusion and confusion, rendering these attacks computationally infeasible against full-round versions.

1.4. Implementation Efficiency

Permutation-based designs are highly amenable to optimization in both software and hardware. They can effectively exploit modern processor features such as SIMD instructions and efficient cache utilization [92]. Their reliance on simple bitwise operations (e.g., XOR, AND, NOT, and rotations) avoids costly table lookups or complex arithmetic, making them suitable for both resource-constrained environments and high-performance computing platforms [114, 156].

1.5. Enhanced Security in the Post-Quantum Era

Although quantum computing reduces the effective security of symmetric-key algorithms—for example, halving brute-force resistance via Grover's algorithm [78, 105, 141, 176]—permutation-based designs, particularly those with larger state sizes and greater security margins, are inherently more resilient than their public-key counterparts. As a result, the focus has shifted toward designing permutations with sufficient quantum security margins;

for instance, achieving 256-bit classical security may require 512-bit keys or internal states in the quantum setting [78, 105, 141, 176].

The growing demand for lightweight cryptographic solutions in IoT environments [59, 191] and high-performance solutions in cloud and big-data systems [92] necessitates primitives that are both secure and efficient to implement. Designing separate, highly optimized primitives for hashing, encryption, and authentication across all platforms is often impractical and economically inefficient. A single permutation capable of supporting multiple cryptographic functionalities, such as hashing, encryption, authenticated encryption (AEAD), and pseudorandom number generation (PRNG), through various constructions (e.g., Sponge, Duplex, and Farfalle) [5, 20–22], offers a compelling alternative. This “build once, deploy many” approach reduces implementation footprint, simplifies integration, and accelerates deployment across heterogeneous environments, directly addressing challenges related to performance, resource constraints, and maintainability.

1.6. Scope and Organization of This Survey

This paper aims to provide a comprehensive survey of permutation-based cryptography, going beyond formal descriptions to explore the underlying principles, design methodologies, and real-world instantiations. The primary focus is on permutations with state sizes of 256 bits or more, which are increasingly relevant for achieving higher security levels (e.g., 128-bit or 256-bit security in the post-quantum context) and for efficiently processing large data blocks. This includes investigating how to achieve full diffusion over large states and the inherent complexity of balancing security and performance [55, 148].

Although permutations such as Ascon-p [59] (320-bit) and Xoodoo/Gimli [19, 50] are significantly larger than traditional AES-128 block ciphers, they are still considered “lightweight” when compared to Keccak-f [23]. This observation illustrates that “large” does not merely refer to bit-size expansion but also to optimization for large yet constrained deployment contexts. It implies distinct design trade-offs, such as the use of simpler S-boxes or linear layers for large lightweight permutations, and highlights ongoing research aimed at striking an optimal balance across different application domains.

A central part of this survey is a comparative analysis of notable permutation-based primitives, including Ascon, Gimli, Xoodyak, Keccak/SHA-3, and BLAKE2/BLAKE3. The analysis covers their design rationale, performance characteristics, and cryptanalytic results, including recent attacks and their limitations, while also identifying open research challenges and future directions. Cryptanalysis is not merely about breaking designs; it is an essential component of the design lifecycle. Attacks often lead to improvements, new versions, or the retirement of weak designs, underscoring the iterative nature of cryptographic research in which the strength of a primitive is continually refined through adversarial analysis.

1.7. Cryptographic Permutations as Building Blocks

At its core, a cryptographic permutation is a bijective mapping that transforms a fixed-size input block into an output block of the same size, with the goal of being indistinguishable from a random permutation [172]. Unlike traditional block ciphers, which are keyed pseudorandom permutations (PRPs) [172], many modern permutation-based primitives—such as Keccak and Ascon—employ unkeyed, public permutations. The security of the overall scheme then depends on how this public permutation is integrated into a higher-level construction, such as Sponge, Duplex, or Farfalle [5, 20–22].

This building-block approach simplifies the design and analysis of complex symmetric cryptographic systems. Rather than designing separate algorithms for hashing, encryption, and authentication, a single well-vetted permutation can support all these functionalities through integration into different modes of operation. This provides a unified cryptographic backbone. The shift from keyed PRPs (block ciphers) to public unkeyed permutations represents a fundamental conceptual departure from traditional symmetric cryptography. By removing the key schedule from the permutation itself, the core primitive can be simpler

Table 1.1. Overview of Permutation-Based Cryptographic Structures and Cryptographic Permutations

Authors	Applications	Construction	Permutation Size	Main Contribution	Year
Bertoni et al. [21]	Hash functions, MACs, stream ciphers	Sponge	Generic n -bit	Proposed the Sponge construction, a finite-state structure, and showed that a random sponge can only be distinguished from a random oracle due to internal collisions. Analyzed attack success probabilities and advocated Sponge-based designs for hash functions, MACs, and stream ciphers.	2007
Bertoni et al. [2]	AEAD, hashing, random number generation	Sponge, Duplex	Generic n -bit	Introduced the Duplex construction, closely related to Sponge, enabling authenticated encryption with one permutation call per block and supporting key wrapping. First mode directly based on permutations rather than block ciphers.	2011
Daemen et al. [24]	Hashing, MACs, stream cipher, PRNG	Sponge, Duplex	1600-bit	Proposed KECCAK, the winner of the NIST SHA-3 competition. Introduced the Keccak- f family (200, 400, 800, 1600 bits) and demonstrated versatile use via Sponge and Duplex constructions.	2012
Mouha et al. [150]	MAC	Even–Mansour	128-bit	Proposed CHASKEY, a lightweight ARX-based MAC for 32-bit microcontrollers, achieving 128-bit security and outperforming AES-CMAC on ARM Cortex-M platforms. Based on provable Even–Mansour security.	2014
Bertoni et al. [20]	PRF, MAC, encryption, AEAD	Farfalle	Generic n -bit	Proposed Farfalle, a highly parallel permutation-based PRF structure. Introduced KRAVATTE, using Keccak- p [1600], with security claims against classical and quantum adversaries.	2016
Bernstein et al. [19]	Block cipher, hashing, stream cipher, AEAD, MAC	Farfalle, Duplex	384-bit	Introduced GIMLI, a 384-bit permutation optimized for diverse platforms including CPUs, microcontrollers, FPGA, and ASICs, with emphasis on performance and side-channel resistance.	2017
Daemen et al. [50]	PRF, encryption, AEAD, MAC	Farfalle	384-bit	Proposed XOODOO, a compact permutation inspired by Keccak- p , operating on three 128-bit planes and designed for efficiency on low-end processors.	2018
Daemen et al. [52]	Hashing, MAC, stream cipher, AEAD	Duplex	257-bit	Introduced SUBTERRANEAN 2.0, a permutation-based suite targeting resource-constrained and low-power hardware environments.	2020
Dobraunig et al. [59]	AEAD, hashing, PRF	Duplex	320-bit	Proposed ASCON, selected as the primary lightweight AEAD in the CAESAR competition. Also introduced Ascon-Hash and Ascon-XOF.	2021
Fuchs et al. [69]	MAC, stream cipher, hashing	Farfalle, Pelican-MAC	n -bit	Generalized Farfalle compression layers to study universality and differential probabilities, showing the superiority of parallel compression over serial designs.	2022
Norica et al. [10]	AEAD	Deck-based	n -bit	Introduced deck functions and several AE modes (Deck-PLAIN, BO, BOREE, JAMBO, JAMBOREE), enabling incremental and analyzable symmetric constructions.	2022
Gunsing et al. [83]	Wide-block cipher modes, keyed hashing	Farfalle-WBC	n -bit	Proposed Double-Decker and Docked-Double-Decker wide-block modes, enhancing security for in-place updates and constrained-memory systems.	2022
Lefevre et al. [125]	PRF, stream cipher	Kirby	Generic n -bit	Proposed Kirby, a VIL-PRF with strong provable bounds, especially efficient for small permutations (e.g., $n = 256$), and robust against state leakage.	2023
Ghosh et al. [73]	Hashing	Farfalle, parallel	128-bit	Introduced Multimixer-128, a highly parallel keyed hash optimized for SIMD architectures such as AVX2, SSE, and NEON.	2023
Ghosh et al. [74]	Hashing	Farfalle, parallel	156-bit	Proposed Multimixer-156, enhancing security over Multimixer-128 using 32-bit multiplication and rotations.	2023
El Hirsch et al. [64]	Block cipher, hashing, AEAD	Even–Mansour, Farfalle	320-bit	Introduced the GASTON permutation with improved differential and linear trail bounds compared to Ascon.	2023
Belkheyar et al. [14]	Block cipher	Even–Mansour	256, 512-bit	Introduced Sonic and SuperSonic, low-latency permutation-based designs for high-throughput environments such as SSDs and DRAM encryption.	2023
Daemen et al. [49]	Authenticated encryption	Deck-BO, SHA-3 based	1600-bit	Proposed AE schemes combining SHAKE and TurboSHAKE with Deck-BO, supporting partial decryption and committing properties.	2024
Müller [152]	AEAD	Sponge	1600-bit	Proposed ASCON-KECCAK AEAD schemes using Keccak permutations, achieving up to 512-bit classical and 256-bit quantum security.	2024
Parisa et al. [3]	Tweakable block cipher, AEAD	Double-Decker	512-bit	Introduced MYSTRUM, a tweakable wide-block cipher optimized for low-end processors using a custom 512-bit permutation.	2024
Lei et al. [127]	Block cipher, AEAD, hashing	Farfalle, Duplex	320-bit	Introduced improved Gaston variants using Symmetric Dual Parity Shuffle (STCPM) to enhance differential and linear resistance.	2024
Parisa et al. [4]	PRF, encryption, hashing, MAC	Kirby	257-bit	Proposed KOALA, a low-latency ASIC-oriented PRF based on the Kirby structure and a new permutation Koala-P.	2024
Lefevre et al. [126]	PRF, encryption, hashing, MAC	Full-state keyed Sponge	n -bit	Introduced MacaKey, enabling full-state absorption and squeezing while preserving provable security bounds.	2025

and potentially faster, while complexity and security are delegated to the mode of operation, enabling more flexible and provably secure multi-purpose designs.

The remainder of this paper is organized as follows. Section 2 introduces the fundamental concepts and formal definitions related to cryptographic permutations. Section 3 describes the primary permutation-based constructions, including the Sponge and its variants, the Duplex construction, and the Farfalle framework, and provides a detailed comparison highlighting their structural characteristics and implementation properties. Section 4 examines the core design components of cryptographic permutations, with particular attention to nonlinear and linear layers, common permutation construction strategies, and essential design criteria for large-state permutations. Section 5 presents an overview of prominent cryptographic permutations and their applications in authenticated encryption, hashing, and pseudorandom number generation, accompanied by a comprehensive comparative table summarizing technical features and design contributions. Finally, Section 6 concludes the paper by

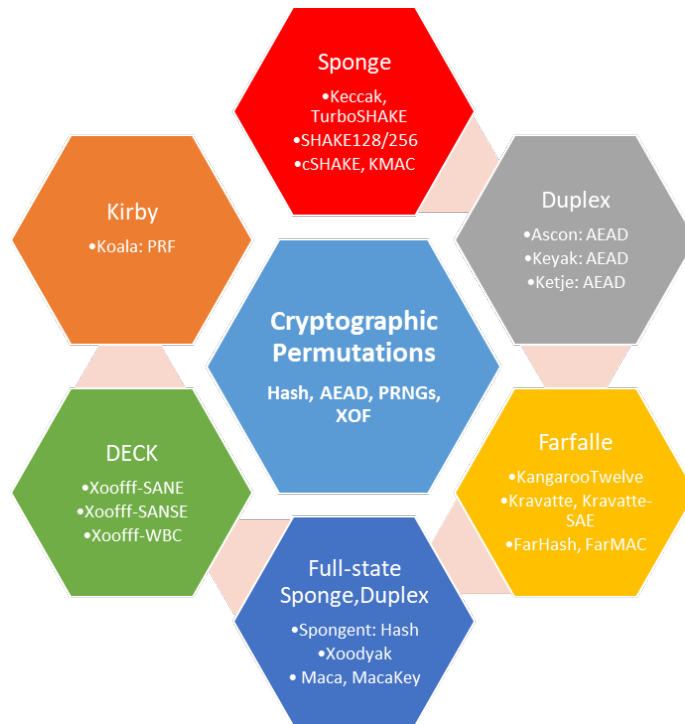


Fig. 2.1. Cryptographic permutations and their applications in symmetric cryptography

summarizing the main findings and outlining open research directions in permutation-based cryptography.

2. BACKGROUNDS

2.1. Definition and Properties of Cryptographic Permutations

A cryptographic permutation is formally defined as a bijective function, meaning it is both injective and surjective, that maps a finite set onto itself. In cryptographic contexts, the term typically refers to a function $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$ that is efficiently invertible and computationally indistinguishable from a random permutation [172]. For a cryptographic permutation, efficient invertibility is generally required, particularly in the context of block ciphers (i.e., keyed permutations), where the decryption function is defined as the inverse of the encryption function [172]. In conventional block cipher design, the encryption function $E_K : \{0, 1\}^n \rightarrow \{0, 1\}^n$ is constructed as a permutation over the set of 2^n possible n -bit inputs.

A cryptographic primitive is said to be *pseudorandom* if its output cannot be efficiently distinguished from that of a truly random function of the same size [172]. A *pseudorandom permutation* (PRP) refers to a function that is computationally indistinguishable from a random permutation under realistic adversarial effort [172]. Traditional block ciphers are typically modeled as PRPs [172]. A *pseudorandom function* (PRF), in contrast, is one that cannot be distinguished from a random function [172].

This property serves as a fundamental security notion for both PRFs and PRPs. A cryptographic scheme is considered secure if no probabilistic polynomial-time adversary can distinguish its output from that of an ideal (i.e., random) scheme [172]. A key result that formalizes the relationship between the indistinguishability of random functions and random permutations is the well-known *PRP/PRF Switching Lemma* [15]. This lemma states that

a secure PRF over a sufficiently large domain (e.g., λ -bit input/output) is indistinguishable from a secure PRP with the same block size λ [172]. As a consequence, for large domains, the practical difference in distinguishability between PRFs and PRPs becomes negligible.

While a permutation is mathematically defined as a bijective function, in cryptographic settings the term “permutation” may refer to both keyed functions (e.g., PRPs, such as block ciphers) and unkeyed public permutations (e.g., Keccak- f). Security notions such as pseudorandomness and indistinguishability are then interpreted according to the specific context. The switching lemma [15, 172] effectively blurs the distinction between PRFs and PRPs in large domains, suggesting that the invertibility aspect of permutations is often less critical in security proofs than their pseudorandom behavior. This distinction is essential when interpreting claims about permutation-based cryptography: it must be clarified whether a design refers to a PRP (where the key induces the permutation) or a public permutation (where security is derived from the mode of operation). This directly impacts how provable security arguments are structured and what assumptions are placed on the underlying primitive.

When used in appropriate constructions, a function is said to be *collision resistant* if it is computationally infeasible to find two distinct inputs that yield the same output [45]. For hash functions built using the Sponge construction, collision resistance is typically limited to $c/2$ bits, where c denotes the capacity [25, 157]. Security in this case stems from the pseudorandom properties of the internal permutation used in the Sponge [21, 26]. The *collapsing* property, a stronger notion particularly relevant in the post-quantum setting, has been shown to hold for the Sponge construction under suitable assumptions on the underlying primitive (e.g., an ideal random function or an unkeyed random permutation) [193].

A consistent finding in the literature is that the capacity parameter c is the main determinant of security in Sponge-based constructions, particularly in relation to collision and preimage resistance [157]. The capacity serves not merely as a parameter, but as a control knob that quantitatively connects the width b of the permutation to the desired security level of the hash function. This enables designers to make explicit trade-offs between efficiency (i.e., rate $r = b - c$) and security, which is particularly important in lightweight cryptography and in adapting to future quantum threats (e.g., doubling the capacity to resist Grover’s algorithm).

2.2. Multi-Purpose Primitives

The key appeal of permutation-based cryptography lies in its ability to derive multiple symmetric cryptographic functionalities from a single, well-designed permutation [5, 20–22]. This stands in contrast to traditional approaches, which typically require chaining distinct primitives—such as pairing a stream cipher with a separate message authentication code (MAC) to provide authenticated encryption. This flexibility simplifies the design of cryptographic systems, reduces implementation overhead, and potentially streamlines security analysis, as the core security properties are concentrated in a single underlying function and an overarching structural framework. The primary structural frameworks that enable this multi-purpose capability are the Sponge construction and its variants, the Duplex construction and its extensions, and the Farfalle construction.

2.2.1. The Sponge Construction and Its Variants The Sponge construction [21, 28] is a foundational mode of operation that builds a function with variable-length input and arbitrary-length output from a fixed-length permutation. It operates in two distinct phases: *absorption*, where input blocks are XORed into a b -bit state (interleaved with applications of the core permutation f), and *squeezing*, where output blocks are extracted from the r -bit portion of the state (again interleaved with f). The Keccak/SHA-3 family [24] is the most prominent example of a cryptographic hash function based on this construction.

Donkey Sponge [27] is a variant of the original Sponge design, specifically tailored for MAC construction. It uses the full internal state during both initialization and absorption phases, allowing for reduced round counts while maintaining security.

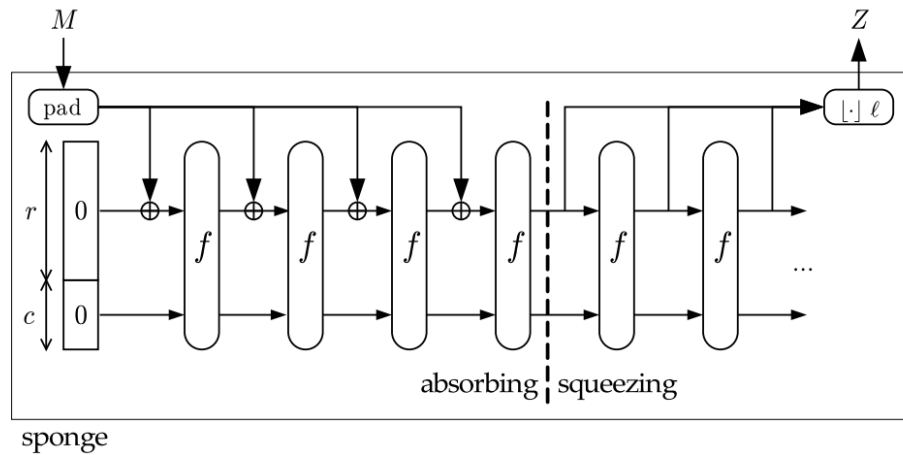


Fig. 2.2. Sponge construction [27]

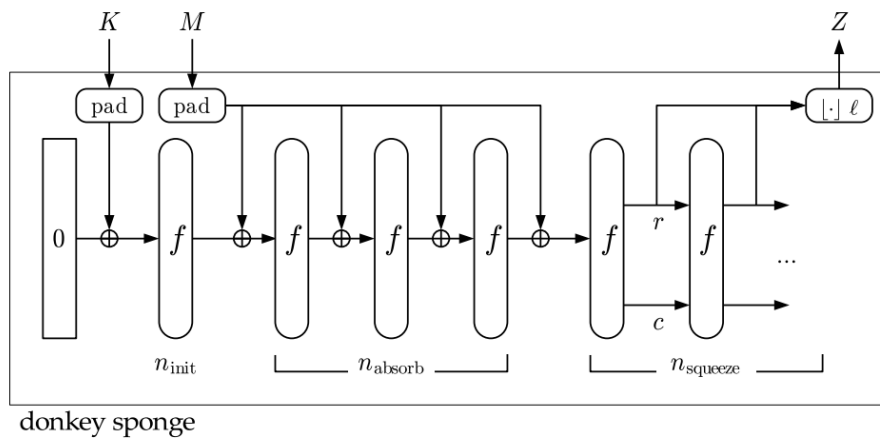


Fig. 2.3. Donkey Sponge construction [27]

2.2.2. The Duplex Construction and Its Variants The Duplex construction [?, 27] extends the Sponge design by enabling alternating input and output processing, effectively supporting two data flows. It is particularly suited for building authenticated encryption with associated data (AEAD) schemes and seeded pseudorandom number generators (PRNGs). A representative instance of this structure is Ascon [59], a lightweight AEAD algorithm based on a Duplex-like mode.

Monkey Duplex [?, 27] is another variant designed for efficient authenticated encryption and keystream generation, offering round reduction in the underlying permutation calls. Full-State Keyed Duplex (FSKD) [53] is a generalization of keyed Duplex [27] intended to enable flexible multi-purpose designs with enhanced security levels. FSKD allows the use of the full state for both absorption and squeezing, including key material, thereby strengthening resistance against side-channel and distinguishing attacks.

2.2.3. The Farfalle Construction The Farfalle construction [20] is a permutation-based framework specifically designed for building pseudorandom functions (PRFs) with explicit support for parallelism. It features distinct compression and expansion layers, each involving

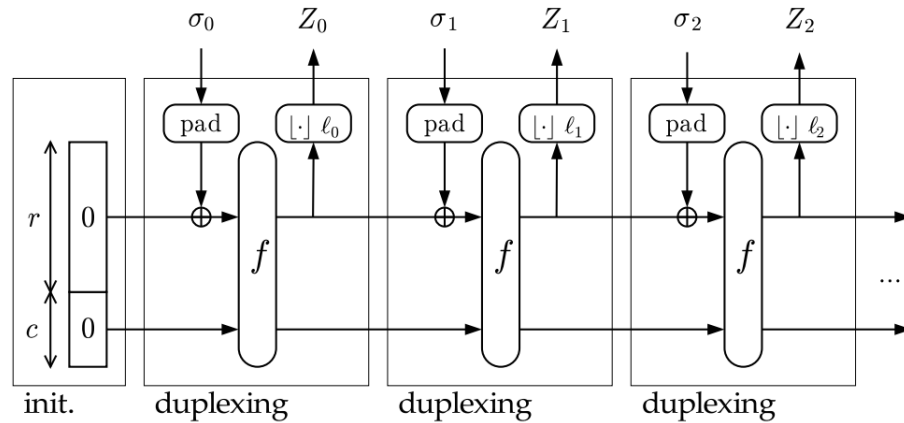


Fig. 2.4. Duplex construction [27]

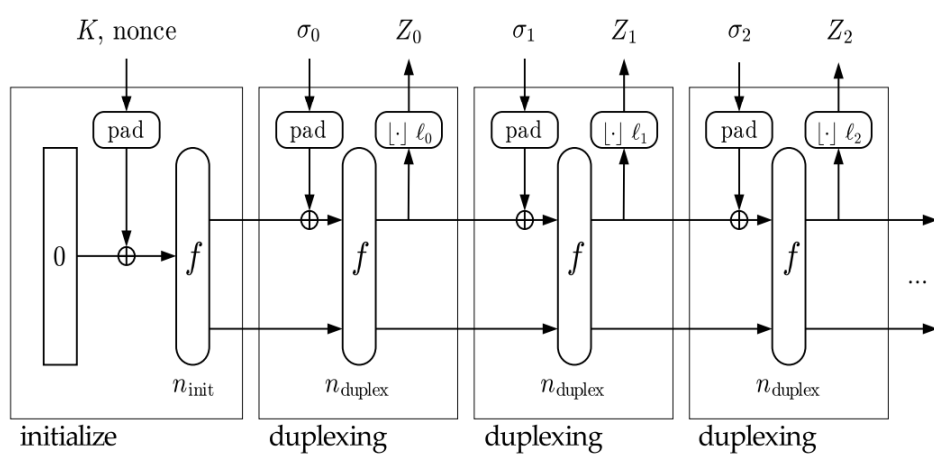


Fig. 2.5. Monkey Duplex construction [27]

parallel applications of a base permutation, accompanied by rolling functions (often LFSR-based) that generate input and output masks. Farfalle is highly versatile, supporting a wide range of use cases, including MACs, stream ciphers, key derivation functions (KDFs), and authenticated encryption with associated data (AEAD). A notable instantiation of this construction is Kravatte [51].

The “hermetic sponge strategy” [29, 38] advocates for designing the core permutation f to resist structural distinguishers—i.e., ensuring it behaves as closely as possible to an ideal random permutation. When this property is combined with provably secure modes (e.g., Sponge and Duplex being as secure as ideal oracles under suitable assumptions) [29], it leads to the following principle: *a strong permutation yields strong primitives*. This strategy significantly simplifies the trust model: rather than separately analyzing the security of a hash function, AEAD scheme, or PRNG, one primarily needs to trust the underlying permutation and the proven security of the composition framework. This philosophy strongly supports wide adoption and standardization, as exemplified by the Keccak/SHA-3 family.

While Sponge and Duplex constructions [29] are inherently sequential, Farfalle [20] is explicitly designed for parallel execution. This architectural shift is not merely a design detail—it directly responds to the capabilities of modern hardware (e.g., SIMD instructions and multi-core processors) [66]. The evolution from fully sequential constructions to parallel-friendly ones reflects the growing importance of hardware parallelism in achieving

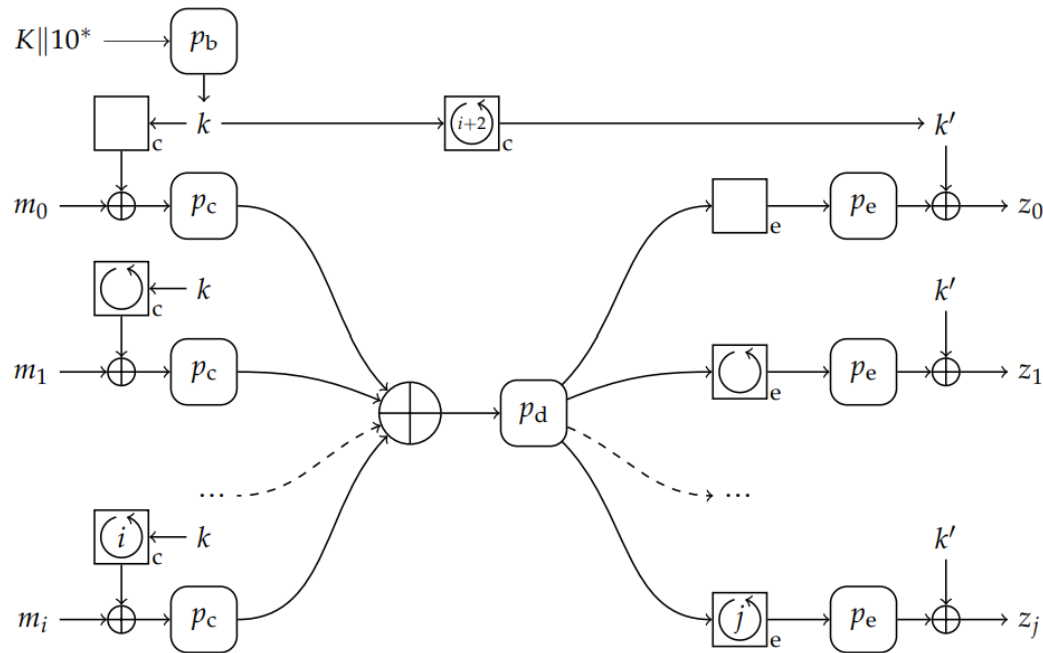


Fig. 2.6. Farfalle construction [20]

high performance. This trend suggests that future permutation-based designs are likely to increasingly prioritize parallelizability, possibly resulting in hybrid frameworks that combine the advantages of both sequential and parallel structures.

2.3. Common Permutation Design Paradigms

Several high-level design paradigms have been widely adopted for constructing cryptographic permutations:

- **Iterated Ciphers.** Permutations are typically constructed as iterated structures, where a fixed number of rounds are applied [121]. Each round is parameterized either by a round constant (in public permutations) or a round key (in block ciphers) [43].
- **Substitution–Permutation Networks (SPNs).** In SPN-based designs, the plaintext and key are processed through alternating layers of nonlinear substitution (via S-boxes) and linear diffusion (often referred to as P-boxes or linear layers) [43]. Prominent examples include AES and Keccak- p , which utilize this structure to achieve both confusion and diffusion.
- **Feistel Networks.** The input block is split into two halves, and a round function—not necessarily invertible—is applied to one half and XORed with the other. The halves are then swapped at each round [43]. Classical ciphers such as DES and Blowfish follow this approach.
- **ARX Designs (Addition–Rotation–XOR).** Round functions are constructed solely from modular addition, bitwise rotation, and XOR operations, providing good software efficiency and resistance to timing attacks [65]. Notable ARX-based constructions include ChaCha20 and BLAKE [43].
- **Lai–Massey Schemes.** Similar to Feistel networks, the Lai–Massey scheme [76, 178] achieves security without requiring invertible round functions and is employed in ciphers such as IDEA.

The continued use of SPN and ARX structures in modern designs such as Keccak- p [24] and BLAKE [7] demonstrates the robustness and longevity of these foundational

strategies. Rather than being rendered obsolete, these paradigms have been adapted and scaled to meet modern cryptographic requirements, including large state sizes and higher performance targets. This suggests that future large-state permutations will likely continue to build upon well-understood paradigms, focusing on optimizing components (e.g., S-boxes and linear layers) and round functions rather than inventing entirely new top-level models. The evolution of permutation design is thus often characterized by incremental refinements of proven concepts rather than radical departures from established frameworks.

2.4. Design Advantages

Simplified Security Analysis By consolidating multiple symmetric cryptographic functionalities around a single, well-defined permutation, security analysis can be significantly streamlined and more tightly scoped. Provable security arguments can often be established for general frameworks (e.g., the Sponge construction under the assumption of a random oracle as the underlying function) [5, 20–22]. This modular approach enables cryptanalysts to focus their efforts on analyzing the permutation itself, with findings subsequently extended to various operational modes and functionalities.

Hardware and Software Optimization Cryptographic permutations are typically constructed from simple bitwise operations (such as XOR, AND, NOT, and bit rotations), which are highly amenable to efficient implementation across a wide range of platforms, from low-end 8-bit microcontrollers to modern 64-bit CPUs with SIMD capabilities. These operations can be optimized for low area and low power consumption in hardware (e.g., Ascon for IoT devices) [59, 191], or for high throughput in software, while avoiding expensive instructions such as lookup tables, which are often vulnerable to cache-timing attacks [168].

Since permutations are built using only bitwise instructions, they are not only fast and compact, but also inherently more resilient to side-channel attacks, particularly timing-based attacks, as they avoid data-dependent memory accesses. This presents a critical advantage in embedded and IoT environments, where secure implementations remain a practical challenge [68, 75, 81]. Such a design philosophy—favoring internal simplicity—serves a dual purpose: enabling efficient performance across both constrained and high-end platforms, while also offering a fundamental level of resistance against implementation-level attacks. This positions permutation-based primitives as strong candidates for secure-by-design deployments in sensitive environments.

Enhanced Quantum Resistance Compared to Traditional Block-Cipher-Based Primitives While symmetric cryptography is generally considered more quantum-resistant than public-key cryptography—due to Grover’s algorithm providing only a quadratic speed-up for exhaustive key search, thus requiring a doubling of the key or state size to maintain classical security levels [78, 105, 141, 176]—block-cipher-based constructions often suffer from structural inefficiencies, complex key schedules, and difficulties in achieving tight security proofs under quantum adversaries [?, 5].

In contrast, permutation-based designs offer several advantages. Their inherent flexibility allows for straightforward increases in state size to accommodate higher levels of post-quantum security (e.g., Keccak- f configurations enabling 256-bit quantum security) [?]. The threat of “harvest now, decrypt later” [162, 181] underscores the need for robust primitives that remain secure even when classical data collected today may be attacked by quantum computers in the future [188]. Permutation-based PRFs are already being analyzed in this context, highlighting potential vulnerabilities when adversaries are equipped with quantum resources.

While classical attacks (e.g., differential [33] and linear [144]) are well understood, the advent of quantum computing—particularly algorithms such as Grover’s and Simon’s [78, 105, 141, 176]—introduces new dimensions in the analysis of symmetric primitives. Simply doubling classical security levels may prove insufficient if quantum algorithms can exploit

structural properties (e.g., Simon's algorithm applied to periodic functions). As a result, the design focus is shifting from merely enlarging key sizes to explicitly addressing quantum attack models. Permutation-based designs, by virtue of their scalable state sizes and configurable round functions, are well suited to adapt to evolving post-quantum security requirements, whether by enlarging capacity or state sizes or by modifying internal structures to resist specific quantum attacks.

2.5. The Sponge Construction and Its Variants

The Sponge construction [21, 29], introduced by Bertoni *et al.*, is a permutation-based framework originally proposed for the design of secure, quantum-resistant hash functions. It operates on an internal state of fixed size b bits, which is partitioned into a *rate* part r and a *capacity* part c , such that $b = r + c$. The construction proceeds in two main phases: *absorption* and *squeezing*.

In the absorption phase, the input message is first padded and divided into blocks of r bits. Each block is XORed into the r -bit portion of the internal state, interleaved with the application of a fixed permutation f over the full b -bit state. This process continues until all input blocks have been absorbed. Once absorption is complete, the squeezing phase begins. In this phase, r -bit output blocks are extracted from the rate portion of the state, again interleaved with applications of the permutation f . The output process continues until the desired number of bits has been produced. Notably, the c -bit capacity portion is never directly exposed to either input or output, serving as a security buffer that protects the internal state from information leakage.

The Sponge framework is a generalized construction for hash functions that can process inputs of arbitrary length and produce outputs of arbitrary length, commonly referred to as extendable-output functions (XOFs). This flexibility enables its use in a wide range of symmetric cryptographic functionalities, including stream ciphers, message authentication codes (MACs), pseudorandom number generators (PRNGs), and key derivation functions (KDFs). The most prominent instantiation of the Sponge construction is Keccak, which was selected as the winner of the NIST SHA-3 competition and subsequently standardized [23, 25]. Keccak employs a 1600-bit permutation, denoted Keccak- f , iterated for 24 rounds, and underlies the SHA-3 family, including SHA3, SHAKE128, and SHAKE256. In these constructions, security is primarily determined by the capacity c , with classical collision resistance of approximately $c/2$ bits. For example, SHAKE128 and SHAKE256 [107] are XOFs based on Keccak- f that allow outputs of arbitrary length.

The security bounds of Sponge-based functions are directly linked to the capacity c [21]. Under classical adversarial models, collision resistance is approximately $c/2$. However, in the presence of quantum adversaries (e.g., those employing Grover's algorithm), the effective security may be reduced [45, 104, 193]. Consequently, achieving a target quantum security level requires a significantly larger capacity, and therefore a larger overall permutation state. This observation leads to an important design implication for future permutation-based primitives in the post-quantum era: to maintain desired security levels against quantum-capable adversaries, the state size of the underlying permutation—and hence the capacity of the Sponge construction—must be substantially increased. This requirement has driven growing interest in the design of large-state permutations as fundamental building blocks for next-generation cryptographic systems.

2.6. The Duplex Construction and Its Variants

The Duplex construction [22, 23, 25] is a variant of the Sponge framework that enables the interleaving of input and output operations. Functionally, it realizes two concurrent streams of the Sponge structure, making it particularly well suited for applications requiring authenticated encryption with associated data (AEAD) and pseudorandom number generation (PRNG). One notable instantiation is Ascon, the winning algorithm of the NIST Lightweight Cryptography (LWC) competition [59, 191], which employs the Duplex mode in its AEAD

scheme, Ascon-AEAD-128. The construction is built on a 320-bit permutation, Ascon- p , optimized for lightweight implementation. Another prominent example is Xoodyak, which utilizes a Duplex-based mode known as *Cyclist* to support AEAD, hashing, and PRNG functionalities, leveraging the 384-bit Xoodoo permutation [47].

While the Sponge construction offers significant flexibility in terms of variable-length input and output [29], its inherently sequential nature can limit performance on architectures that support high degrees of parallelism. The Duplex construction [22] addresses some of the state-handling demands in these contexts; however, it also operates sequentially, which can introduce performance bottlenecks in throughput-critical applications. This observation reveals a fundamental design trade-off: the elegance and compactness of relying on a single permutation for multiple cryptographic functionalities versus the need to optimize for specific performance characteristics.

Several variants of the Duplex construction have been proposed to address these limitations. For example, the Donkey Sponge [29] employs full-state processing to construct fast and secure message authentication codes (MACs). The Monkey Duplex [147] modifies the Duplex scheduling to reduce the number of permutation rounds per call, making it particularly suitable for streaming and key derivation settings.

Although the basic Sponge framework remains a powerful and general-purpose structure, its specialized variants and alternative parallel architectures, such as Farfalle [20], have been introduced to overcome specific throughput bottlenecks. These developments highlight the inherent challenge of designing a single “ideal” permutation capable of efficiently supporting all cryptographic use cases. As a result, a growing ecosystem of tailored permutations and structural frameworks has emerged, each optimized for distinct security and performance requirements.

2.7. The Farfalle Construction

The Farfalle construction [20], introduced by Bertoni *et al.*, is a permutation-based framework designed to achieve high parallelism, offering significant performance advantages over inherently sequential designs such as Sponge and Duplex. Farfalle is structured to build pseudorandom functions (PRFs) while explicitly enabling efficient parallel execution [29].

The construction consists of three main stages: initialization, compression, and expansion. Both the compression and expansion phases involve parallel applications of the underlying permutation. Input and output masks are generated using rolling functions, typically based on linear feedback shift registers (LFSRs), which also contribute to state updates during the expansion phase, thereby enhancing diffusion and permutation of the data. A key feature of the design is its *incrementality*: common input prefixes can be eliminated during the compression stage, leading to improved efficiency in practical applications.

One of the primary strengths of Farfalle is its inherent parallelizability, which makes it particularly well suited for modern processors equipped with SIMD units or multi-core architectures. Once the corresponding input masks have been generated, permutation calls can be executed independently and in parallel. Unlike Duplex or traditional block-cipher-based modes, Farfalle imposes fewer structural constraints on the underlying permutation, potentially allowing the use of a reduced number of rounds. This flexibility enables support for a broad range of symmetric cryptographic functionalities, including message authentication codes (MACs), stream ciphers, key derivation functions (KDFs), authenticated encryption with associated data (AEAD), and pseudorandom number generation.

A prominent instantiation of the Farfalle construction is Kravatte, which is built on top of KECCAK- p permutations. In response to subsequent cryptanalysis, the design was revised to increase the number of rounds in all four permutation instances and to replace the original linear rolling function in the expansion layer with a nonlinear alternative in order to mitigate structural attacks [31]. This evolution highlights the responsiveness of the Farfalle framework to cryptanalytic findings and underscores the importance of continuous security evaluation throughout the design lifecycle.

In contrast to the inherently sequential nature of Sponge and Duplex, Farfalle was designed from the outset to support parallel execution, reflecting a forward-looking design philosophy that anticipates the trajectory of modern computing architectures. A key performance–security trade-off is exploited: the underlying permutation can be instantiated with fewer rounds because the adversary never obtains simultaneous access to both the input and output of any individual permutation call. This insight enables more efficient constructions while preserving strong security guarantees.

The revision of Kravatte [31], prompted by timely third-party cryptanalysis—including block-based attacks and analyses targeting the expansion layer—illustrates how cryptographic research actively shapes the evolution of emerging primitives. The transition from linear to nonlinear rolling functions represents not merely an implementation change but a broader shift in design philosophy. It emphasizes that even carefully structured designs may harbor unforeseen weaknesses and that independent, rigorous cryptanalysis is essential for strengthening cryptographic constructions. Overall, Farfalle represents a significant evolutionary step in permutation-based cryptography, addressing the throughput limitations of earlier designs and charting a path toward future frameworks optimized for parallel processing. The structure also raises intriguing questions regarding new proof models in which adversaries are granted only partial access to permutation inputs or outputs, thereby enabling reduced-round constructions without compromising overall security.

2.8. Towards Extendable-Output Function (XOF) Constructions

All of the previously discussed constructions ultimately aim to leverage a fixed-size permutation to produce outputs of arbitrary length. An extendable-output function (XOF) is a cryptographic primitive that allows an arbitrary amount of output to be generated from a fixed input [?, 165, 190]. Within Sponge-based designs, the permutation plays a central role during the expansion phase, where it is repeatedly applied to the internal state while output bits are iteratively extracted. This property makes XOFs extremely versatile, enabling their use as building blocks for key derivation functions (KDFs), stream ciphers, message authentication codes (MACs), and pseudorandom number generators (PRNGs).

SHAKE128 and SHAKE256 are canonical examples of standardized XOFs [157], introduced as part of the SHA-3 family. These constructions are based on the Keccak- f permutation [23] and follow the Sponge structure [29]. Unlike their fixed-output SHA-3 counterparts, SHAKE functions employ a distinct domain separator and support continuous output expansion. Furthermore, cSHAKE , as specified in NIST SP 800-185, introduces customization strings to prevent output collisions across different applications, thereby enabling explicit domain separation in a manner conceptually similar to context separation in HKDF [107].

While the Sponge framework already supports a wide range of symmetric cryptographic primitives, XOFs further elevate this flexibility by enabling truly unbounded output lengths. As a result, a single XOF instance can effectively replace multiple specialized primitives, including KDFs, stream ciphers, and PRNGs. The significance of this capability lies not only in structural flexibility but also in the simplification of cryptographic system design. XOFs embody the culmination of the “*permutation-as-a-universal-primitive*” philosophy, enabling arbitrary-length pseudorandom bit generation from compact seeds or messages. This makes XOFs particularly attractive for modern protocols that demand flexible output lengths, reduced implementation complexity, and streamlined engineering processes.

The existence of cSHAKE , with its use of customization strings, highlights a critical security consideration for multipurpose permutation-based primitives: *domain separation*. Without explicit separation mechanisms, reusing the same underlying permutation across different applications (e.g., hashing and key derivation) may lead to unintended interactions or security vulnerabilities. For highly flexible permutation-based designs, domain separation—achieved through customization strings, distinct padding rules, or separate invocation parameters—is not optional but essential. It ensures that the cryptographic properties of one

Table 2.2. Comparison of Major Permutation-Based Cryptographic Structures

Feature	Sponge [29]	Duplex [29]	Farfalle [20]	XOF
Operational Principle	Absorb–then–squeeze (sequential)	Interleaved absorption and squeezing (sequential)	Parallel compression and expansion with rolling/masking	Continuous expansion (sequential)
Advantages	High flexibility (hashing, XOF, PRNG); provable security in the random oracle model; conceptual simplicity	Stateful applications; efficient reseeding for PRNG and AEAD; security equivalent to Sponge	High parallelizability; incremental property; fewer rounds required; versatile across modes	Arbitrary-length output; unified primitive for KDFs, stream ciphers, and PRNGs; strong collision and preimage resistance
Drawbacks	Inherently sequential; performance bounded by rate r	Sequential; inherits performance limitations of Sponge	More complex design (multiple permutations, rolling functions); newer and less cryptanalyzed than Sponge	Extended output does not increase security; risks of related-output issues without careful domain separation
Typical Use Cases	Hash functions, XOFs	AEAD, stateful PRNGs, MACs	PRFs, MACs, stream ciphers, AEAD, wide-block encryption	KDFs, stream ciphers, PRNGs, hashing
Example Permutations Used	Keccak- f (SHA-3) [23]	Ascon- p (Ascon) [59], Xoodoo (Xoodyak) [50]	KECCAK- p (Kravatte) [51]	Keccak- f (SHAKE) [107]

application do not inadvertently compromise another, thereby reinforcing the robustness of the “one primitive, many purposes” design paradigm.

3. CORE DESIGN COMPONENTS OF CRYPTOGRAPHIC PERMUTATIONS

3.1. Non-linear Layer

The non-linear layer plays a pivotal role in the design of cryptographic permutations by providing resistance against a wide range of cryptanalytic techniques. Two primary classes of non-linear operations are widely employed: substitution boxes (S-boxes) and modular addition, with several alternative approaches emerging to meet specific architectural or security requirements.

Substitution Boxes Substitution boxes (S-boxes) [117] are fundamental non-linear components in symmetric cryptography, essential for introducing confusion in the sense of Shannon’s definition [202]. An S-box transforms a fixed-size input bit string into a different output bit string, typically via a bijective mapping. Classical designs such as DES and AES employ fixed lookup tables, whereas ciphers like Blowfish use key-dependent, dynamically generated S-boxes [171]. In general, the size of an S-box, denoted as an $m \times n$ bit mapping, plays a crucial role in determining both the efficiency of its implementation and the level of security it provides [57, 71].

The effectiveness and security of an S-box are determined by a set of well-established criteria developed in response to known cryptanalytic attacks, including differential [144], linear [33], and algebraic attacks [120]. Notably, the DES S-boxes were confidentially tuned to resist differential cryptanalysis prior to the public disclosure of the technique [158], illustrating the deep interdependency between S-box design and advances in cryptanalysis. Key design criteria for S-boxes include nonlinearity [119, 144], differential uniformity [33, 119], the strict avalanche criterion (SAC) and bit independence criterion (BIC) [119], algebraic degree [119, 120], balancedness [119], and multiplicative complexity [119]. Together, these criteria serve as practical guidelines for constructing robust S-boxes. However, due to the evolving landscape of cryptanalysis, the notion of an “optimal” S-box remains dynamic and is continually reshaped by the discovery of new attacks and evaluation metrics.

Modular Addition Modulo 2^n Modular addition modulo 2^n is a non-linear operation primarily employed in ARX (Addition–Rotation–XOR) constructions [65], where nonlinearity arises from carry propagation, a phenomenon that is inherently non-linear over $\text{GF}(2)$ [58]. When combined with bitwise rotation and XOR, modular addition contributes to both confusion and diffusion in a lightweight and efficient manner.

ARX-based permutations, such as BLAKE [7], exploit this simplicity to achieve strong security margins while maintaining high performance across a wide range of platforms. Their reliance on arithmetic and logical operations enables constant-time implementations, which are inherently more resilient to timing and cache-based side-channel attacks [110, 128, 129, 168]. As a result, ARX designs are particularly well suited for software-oriented implementations and post-quantum cryptographic primitives that require larger state sizes and strong resistance to side-channel attacks [183, 197].

Alternative Non-linear Components Beyond S-boxes and ARX-based operations, a variety of alternative non-linear mechanisms have been explored. These include non-linear feedback shift registers (NLFSRs), non-linear Boolean functions, and combinational logic circuits designed to satisfy specific cryptographic properties. The choice of a non-linear component must be carefully aligned with the overall security and implementation objectives of the permutation, as a weak non-linear layer can significantly compromise the robustness of the entire cryptographic primitive.

3.2. Linear Diffusion Layer

The linear diffusion layer is a fundamental component of cryptographic permutations, responsible for rapidly spreading local changes across the entire internal state. Effective diffusion ensures that small differences introduced by the non-linear layer propagate globally within a few rounds, thereby increasing resistance to differential, linear, and related cryptanalytic attacks.

MDS Matrices and Their Variants Maximum Distance Separable (MDS) matrices [180] are linear transformations defined over finite fields that offer optimal diffusion properties. They are closely related to MDS codes, which achieve the Singleton bound in coding theory. An MDS matrix provides perfect diffusion, meaning that a change in any k input components affects at least $n - k + 1$ output components, where n denotes the total number of components [179]. This property ensures that local differences in the state rapidly propagate across the entire state space, significantly increasing the complexity of cryptanalysis.

A canonical example is the *MixColumns* transformation [79] in AES, which is instantiated as a circulant MDS matrix. Each output byte of a column depends on all four input bytes of that column; after only two rounds, every bit of the state depends on all bits from two rounds earlier, achieving full diffusion [54].

Despite their strong cryptographic diffusion properties [139], MDS matrices often incur considerable implementation costs, particularly for large state sizes [88]. It is well known that a binary matrix (i.e., defined over $\text{GF}(2)$) cannot be MDS unless it has only a single row or a single column, implying that practical MDS matrices typically operate over larger finite fields. This characteristic can complicate bit-oriented and lightweight implementations. As a result, alternative constructions such as near-MDS matrices and circulant MDS matrices [80, 88, 89, 118, 137, 174] have been extensively studied to balance diffusion strength and implementation efficiency.

Column Parity Mixers (CPMs) and Their Variants Column Parity Mixers (CPMs), introduced by Stoffelen *et al.* [186], are a class of linear diffusion layers that rely exclusively on XOR operations and bitwise rotations. They generalize the θ -mixing layer used in Keccak and

operate naturally on two-dimensional state arrays, making them well suited for both software and hardware implementations.

Many CPM constructions are based on circulant matrices [115, 161], which have a regular structure in which each row is a cyclic rotation of the preceding one. Such matrices can be compactly specified by their first row, resulting in minimal storage overhead [90, 134, 140, 151]. Moreover, CPMs derived from circulant matrices can be designed as involutions (i.e., self-inverse mappings), which is advantageous for cryptographic permutations that require efficient inversion.

A limitation of the original CPM design is that its differential and linear branch numbers are both bounded by 4, regardless of the state size [186, 187]. To overcome this restriction, several enhanced variants have been proposed. The Twin Column Parity Mixer (TCPM) [64] improves the differential branch number to 12 while maintaining a linear branch number of 4. Subsequently, the Symmetric Twin Column Parity Mixer (STCPM) [127] achieves both differential and linear branch numbers equal to 12, representing a significant improvement in diffusion strength while preserving the structural simplicity of CPM-based designs.

Other Linear Diffusion Layers Beyond matrix-based constructions, generalized bit permutations (P-boxes) are widely employed in Substitution–Permutation Networks (SPNs) to redistribute the output bits of S-boxes across different inputs in subsequent rounds [34, 60, 135]. A prominent example is the *ShiftRows* transformation in AES, which cyclically shifts the rows of the state matrix. When combined with the *MixColumns* operation, *ShiftRows* plays a crucial role in achieving global diffusion across the internal state.

3.3. Structures for Constructing Cryptographic Permutations

Design of Round Functions Cryptographic permutations are most commonly constructed as iterated structures, where a simple round function is applied repeatedly for a fixed number of rounds [112]. Each round function typically combines a non-linear layer (e.g., S-boxes or modular addition) with a linear diffusion layer (e.g., matrix multiplications or bit permutations) in order to achieve full diffusion and strong resistance against cryptanalysis.

Full diffusion refers to the property that every output bit depends on every input bit, and that a single input-bit modification affects approximately half of the output bits, a phenomenon known as the avalanche effect [41, 95]. Round constants are commonly introduced to break structural symmetries and to prevent attacks exploiting round equivalence, such as slide attacks and invariant subspace attacks [13, 44, 123]. The contrast between the relatively *slow diffusion* of Feistel-based structures and the *rapid diffusion* achieved by SPN-based designs highlights a central concern in permutation design: the speed at which local differences propagate across the state [177].

Feistel Networks In the classical Feistel structure, the input block is divided into two equal halves. In each round, a round function F —which need not be invertible—is applied to one half, and its output is XORed with the other half, followed by a swap of the two halves. Generalized Feistel networks extend this concept by partitioning the state into more than two branches and processing them in various configurations [96, 159].

A key advantage of Feistel networks is that invertibility of the overall permutation is guaranteed regardless of whether the round function itself is invertible, which simplifies design and analysis [153]. Decryption can be performed by reversing the order of round keys and operations. However, Feistel structures typically exhibit slower diffusion than SPNs, as differences propagate only partially through the state in each round. It is generally accepted that at least three rounds are required to achieve a secure pseudorandom permutation (PRP) [172].

Substitution–Permutation Networks (SPNs) Substitution–Permutation Networks (SPNs) consist of alternating layers of non-linear substitution (via S-boxes) and linear diffusion

Table 3.3. Overview of Cryptographic Permutation Construction Models

Construction	Core Structure	Advantages	Disadvantages	Applications / Examples
Feistel	Splits the input block into two halves; applies a non-invertible round function to one half, XORs the result with the other half, then swaps halves.	Round function does not need to be invertible; simple inversion of the overall structure.	Slower diffusion compared to SPN; typically requires more rounds to achieve full security.	DES, Blowfish
SPN	Alternates layers of nonlinear substitution (S-boxes) and linear diffusion; round keys are injected via XOR.	Fast diffusion; strong security margins; relatively easy to analyze.	S-boxes must be invertible for decryption; may offer less flexibility than ARX in software-oriented designs.	AES, Keccak- p (SHA-3), Ascon, Xoodyak
ARX	Uses only modular addition (mod 2^n), fixed bit rotations, and XOR operations.	Simple design; high software efficiency; strong resistance to timing and cache-based side-channel attacks.	Security analysis is more complex due to bit-wise and word-wise interaction; often requires many rounds for sufficient diffusion.	ChaCha20, BLAKE/BLAKE2/BLAKE3, Speck
Lai–Massey	Similar to Feistel but employs a structurally distinct mechanism to ensure invertibility without requiring the round function to be reversible.	Round function need not be invertible; strong theoretical foundations.	May be slower than Feistel or SPN depending on instantiation and parameter choices.	IDEA
Permutation using CPM	Employs Column Parity Mixers (CPMs) as the linear diffusion layer.	Efficient hardware and software implementation; can be involutive (efficient inversion); well suited for state sizes that are powers of two.	May not achieve optimal diffusion compared to MDS matrices; less extensively studied than classical diffusion layers.	Keccak [23], Ascon [59], Mixer [186], Gimli [19], Gaston [64]

(via P-boxes or matrix-based transformations), with round keys XORed into the state [77]. This architecture is well known for its rapid diffusion and efficient implementation [34]. SPNs naturally support parallel execution, making them particularly attractive for modern processors with multiple execution units.

Prominent examples of SPN-based block ciphers include AES (Rijndael) [54] and Serpent [113]. Many modern cryptographic permutations, such as Keccak- p [25] (the core of SHA-3) and Ascon- p [59], also follow SPN principles, adhering closely to Shannon’s concepts of confusion and diffusion.

ARX Structures (Addition–Rotation–XOR) ARX-based designs rely exclusively on modular addition modulo 2^n , fixed bitwise rotations, and XOR operations [65]. This restricted operation set enables high software performance, as these operations are natively supported by most general-purpose processors. ARX constructions are also relatively resistant to certain classes of side-channel attacks, particularly timing-based attacks, due to their constant-time behavior [110, 128, 129, 168].

The interaction between bit-wise operations (XOR and rotation) and word-wise operations (addition) introduces complex non-linear behavior, which often complicates classical differential and linear cryptanalysis [11, 82]. As a result, specialized techniques such as rotational cryptanalysis are frequently employed against ARX-based schemes [97, 110, 111, 170]. Notable ARX-based primitives include the stream cipher ChaCha20 [17] and the cryptographic hash function BLAKE [8].

Lai–Massey Structures The Lai–Massey structure [122] is an iterative block cipher design related to Feistel networks but with round operations that inherently preserve invertibility, even when the round function itself is not invertible. This property allows for flexible round function design while maintaining structural simplicity. IDEA [146] is a canonical example of a cipher employing the Lai–Massey construction.

CPM-Based Structures Although Column Parity Mixers (CPMs) are most commonly discussed as linear diffusion layers, they can also be interpreted as a design paradigm for the linear component of SPN-style constructions, particularly in bit-oriented designs [186]. CPM-based structures offer efficient diffusion using only XOR operations and bit rotations, while retaining straightforward invertibility. These properties make CPMs especially attractive for cryptographic permutations with power-of-two state widths and for implementations targeting both software and hardware efficiency.

3.4. Design Criteria for Large-State Cryptographic Permutations

Security. Large-state permutations must exhibit strong resistance against a broad spectrum of cryptanalytic techniques, including classical methods such as differential cryptanalysis [33], linear cryptanalysis [144], higher-order differential attacks [120], and advanced distinguishers such as zero-sum [40, 205], invariant subspace [13, 44, 123], side-channel attacks [185, 207], and round-reduced attacks [62, 93]. Recent studies [16, 99, 130, 133, 136, 163, 196, 205] have shown that structural weaknesses in permutations can be exploited to significantly reduce their effective security margin. Beyond classical cryptanalysis, the emergence of quantum computing introduces new challenges. Algorithms such as Grover's and Simon's [78, 105, 124, 141, 176] indicate that merely doubling the classical security level may be insufficient, particularly when the permutation exhibits exploitable periodicity or algebraic structure (e.g., Simon's algorithm on cyclic functions). Consequently, the design of large-state permutations must explicitly consider quantum security models (Q1 and Q2) and ensure robustness against dedicated quantum attacks.

Performance. Efficiency in both software and hardware implementations is crucial for large-state permutations. On modern CPUs, design choices should leverage SIMD (Single Instruction, Multiple Data) instructions and bit-sliced implementations [143, 155] to enable parallel processing. Cache-friendliness, such as avoiding data-dependent table lookups, is also essential for minimizing performance degradation [200]. In hardware, design objectives include minimizing gate count (area), maximizing throughput, and reducing power consumption—especially for resource-constrained environments such as IoT and RFID devices [1, 142, 145, 164]. Techniques such as efficient pipelining, loop unrolling, and the extensive use of simple bitwise operations further contribute to meeting these goals [61, 166].

Simplicity. A simple permutation design is easier to implement, verify, and analyze for security vulnerabilities [5]. Reduced structural complexity lowers the risk of implementation errors and facilitates independent scrutiny by the cryptographic community. As a result, simplicity is widely regarded as a desirable and frequently emphasized design objective in modern permutation-based cryptography [21, 28].

Flexibility. Permutations that support multiple state sizes, tunable parameters (e.g., rate versus capacity), or variable output lengths are more versatile and adaptable to a wide range of applications [21, 28]. Such flexibility allows a single permutation core to serve as the foundation for multiple cryptographic primitives, including hash functions, stream ciphers, and extendable-output functions (XOFs), while accommodating different security levels and deployment contexts.

Challenges of Large-State Designs. Designing cryptographic permutations with large internal states introduces several fundamental challenges:

1. *Ensuring full diffusion over large state spaces.* As the state size increases (e.g., beyond 256 or even 1024 bits), it becomes increasingly difficult to guarantee that a single-bit modification in the input propagates to all output bits within a reasonable number of rounds [148].
2. *Balancing security and performance.* Although larger states can provide higher theoretical security margins, they often incur non-negligible performance penalties. Determining an appropriate number of rounds and selecting efficient round components that jointly satisfy security and performance requirements remains a persistent design challenge [56, 169, 173].
3. *Scalability of security analysis.* Comprehensive cryptanalysis of large-state permutations is computationally demanding. While automated tools such as MILP or SAT solvers are increasingly employed, they still face scalability limitations when analyzing large permutations across the full spectrum of known attack vectors.

This table provides a structured framework for evaluating and designing large-state cryptographic permutations. It highlights the inherent multi-objective optimization problem faced by designers, as well as the trade-offs involved among security, performance, simplicity,

Table 3.4. Design Criteria and Key Challenges for Large-State Cryptographic Permutations

Criterion	Description	Challenges for Large Sizes
Security	Resistance against cryptanalytic attacks, including differential, linear, higher-order, zero-sum, and invariant subspace attacks; sufficient number of rounds for provable security.	Ensuring full diffusion over large state spaces; increased complexity of cryptanalysis; resistance against quantum attacks (e.g., Grover's and Simon's algorithms).
Performance	Optimization for software (SIMD utilization, cache efficiency) and hardware (area, throughput, power consumption).	Balancing security and performance; achieving efficiency across diverse architectures (CPU, GPU, FPGA, ASIC).
Simplicity	Ease of implementation, verification, and cryptographic analysis.	Maintaining structural simplicity while achieving high levels of security and performance.
Flexibility	Support for multiple state sizes, tunable parameters, or diverse application domains.	Designing parameterizable structures without compromising security or performance guarantees.
Full Diffusion	Ensures that every input bit influences every output bit after a bounded number of rounds.	Achieving fast and complete diffusion over very large internal state spaces.
Quantum Resistance	Capability to withstand adversaries equipped with quantum algorithms (e.g., Grover's and Simon's).	Requires larger state or key sizes; structural designs must explicitly resist quantum attack models.

flexibility, and quantum resistance. As such, it serves as a useful reference for both researchers and practitioners working in the area of permutation-based cryptography.

4. NOTABLE CRYPTOGRAPHIC PERMUTATIONS AND APPLICATIONS

4.1. Authenticated Encryption (AEAD)

The Ascon encryption algorithm, proposed by Dobraunig *et al.* [59, 191], was selected as the winner of the NIST Lightweight Cryptography (LWC) standardization process in 2023. Ascon is a lightweight permutation-based primitive designed for resource-constrained environments such as IoT devices and embedded systems. It employs a 320-bit SPN-style permutation, denoted Ascon- p . The round transformation consists of round constant addition, a 5-bit bit-sliced S-box applied 64 times in parallel, and a linear diffusion layer. Ascon adopts a Duplex-based mode of operation to support AEAD (Ascon-AEAD128), hashing (Ascon-Hash256), and extendable-output functions (Ascon-XOF128), providing a claimed security level of 128 bits. While Ascon is considered robust, cryptanalytic research continues to examine its security margin. Recent studies report improved key-recovery attacks on reduced-round variants (up to 7–9 rounds) using block collision and differential-linear techniques; however, these attacks do not affect the full 12-round version. Side-channel resistance of Ascon [106] also remains an active research topic.

Gimli, proposed by Bernstein *et al.* [19], is a 384-bit cryptographic permutation designed to achieve high performance across a wide range of platforms, including 8-bit AVR microcontrollers, 64-bit Intel/AMD CPUs, as well as FPGA and ASIC implementations, while maintaining strong security guarantees. As a second-round candidate in the NIST LWC competition, Gimli employs a 384-bit state organized as a 3×4 matrix of 32-bit words and operates over 24 rounds. Its round function combines SPN and ARX principles, using bitwise rotations, XOR operations, and a non-linear Boolean operation of the form $y \wedge z$. Additional diffusion is achieved through *Small-Swap* and *Big-Swap* operations applied every four rounds. Cryptanalytic results include distinguishers against full-round Gimli with complexity up to 2^{64} , as well as collision attacks on reduced-round Gimli-Hash (up to 12 classical and 14 quantum rounds), and state or key-recovery attacks on Gimli-AEAD (up to 9–12 rounds). These analyses delineate the security margin but do not threaten the full-round construction.

Xoodyak, introduced by Daemen *et al.* [47, 48], is a flexible permutation-based cryptographic primitive that combines the 384-bit Xoodoo permutation with a mode of operation known as Cyclist. It supports hashing, encryption, MAC, and AEAD

functionalities, with a claimed security level of 128 bits. As a finalist in the NIST LWC competition [192], Xoodoo is a compact SPN-inspired permutation derived from Keccak- p , optimized for efficient implementation on low-end processors. It operates on a state of twelve 32-bit words over 12 rounds. The Cyclist framework, which is based on the Duplex construction, incorporates mechanisms intended to enhance resistance against side-channel attacks. Cryptanalysis efforts [63, 70, 132, 194] have investigated preimage attacks on reduced-round Xoodyak-XOF (up to three rounds), block-level attacks on six-round initializations, and higher-order differential-linear attacks on five-round Xoodyak initializations [99]. These results confirm the existence of a security margin but do not compromise the full construction. Ongoing research also explores high-probability differential trails for potential collision attacks [206].

The NIST LWC standardization process [59, 191] has significantly accelerated the development and adoption of permutation-based AEAD schemes such as Ascon, Gimli, and Xoodyak. Beyond standard selection, this process has effectively validated the permutation-based design paradigm. NIST's endorsement of Ascon as the primary standard provides strong affirmation for permutation-based AEAD, particularly in resource-constrained environments, and is likely to encourage broader industrial adoption and continued academic research. Although described as *lightweight*, Ascon, Gimli, and Xoodyak employ medium-to large-state permutations (320–384 bits), which are larger than traditional block ciphers. In this context, the term *lightweight* should not be interpreted as implying small state sizes, but rather as denoting designs optimized for constrained environments through careful trade-offs between state size, number of rounds, and component complexity. Consequently, permutation-based lightweight cryptography often results in families of variants that offer different security–performance trade-offs, rather than a single universal solution.

4.2. Hashing

Keccak, proposed by Bertoni *et al.* [?], was the winning algorithm of the NIST SHA-3 competition in 2012 and was standardized as FIPS PUB 202 in 2015. It represents a foundational example of a large cryptographic permutation used within the sponge construction. Keccak employs the Keccak- f permutation, which operates on a 1600-bit state over 24 rounds. Each round consists of five transformation steps (θ , ρ , π , χ , and ι), composed of simple bitwise operations such as XOR, AND, NOT, and fixed rotations. The sponge framework enables flexible adjustment of the rate and capacity parameters, allowing Keccak-based constructions to support a wide range of security levels and functionalities. Since its introduction, the security of Keccak and its variants has been extensively analyzed in the literature [37, 84, 85, 94, 98, 131, 132, 167, 189, 198, 199, 203].

PHOTON [87] is a family of lightweight hash functions based on the sponge construction and optimized for ultra-compact hardware implementations, achieving implementations as small as approximately 1120 gate equivalents (GE) for 64-bit collision resistance. PHOTON was a finalist in the NIST LWC competition [192]. Its internal permutation follows an AES-like structure but incorporates a novel sequential MixColumns layer to significantly reduce area requirements. The PHOTON family offers explicit trade-offs between preimage resistance and performance on short messages, and some variants deliberately reduce second-preimage resistance to achieve extreme compactness. Recent meet-in-the-middle preimage attacks on reduced-round PHOTON represent the first substantial cryptanalytic results on the design since its proposal in 2011 [138, 154].

SPONGENT [35, 36] is another family of lightweight hash functions built on the sponge construction, employing a PRESENT-style internal permutation. It is recognized for having one of the smallest hardware footprints across a wide range of security levels. The design follows a so-called *tight sponge strategy*, supporting multiple digest sizes (88, 128, 160, 224, and 256 bits) and corresponding security strengths, including configurations that intentionally reduce second-preimage resistance to minimize hardware cost [195]. However, without adequate countermeasures, the underlying permutation can be vulnerable to side-channel

key-recovery attacks. Cryptanalytic studies include improved meet-in-the-middle attacks on reduced-round variants and analyses of the impact of S-box design choices on overall security [201].

The SHAKE and cSHAKE families provide standardized extendable-output functions (XOFs), allowing hash outputs of arbitrary length [107, 108]. These constructions are also based on the Keccak- f permutation and the sponge framework. The cSHAKE variant introduces a customization string to enable domain separation, thereby preventing collisions between different application contexts. While XOFs offer significant flexibility, it is important to note that arbitrary-length output does not increase security beyond the defined capacity, and improper usage may lead to security pitfalls.

BLAKE2 and BLAKE3 are high-performance hash functions built upon ARX-based permutations derived from the ChaCha stream cipher. BLAKE2 [9] significantly outperforms MD5, SHA-1, SHA-2, and SHA-3 on 64-bit architectures [7]. It is derived from the SHA-3 finalist BLAKE, with simplified padding, removal of message-word constant additions, and a reduced number of rounds [86]. BLAKE2 supports a wide range of features, including keyed hashing, salting, personalization, and tree hashing. BLAKE3 further improves throughput by adopting a binary Merkle tree structure and exploiting default parallelism, particularly on multicore processors and SIMD-capable architectures [160]. Processing messages in parallel blocks of 1024 bytes, BLAKE3 supports modes for unkeyed hashing, keyed hashing (PRF/MAC), and key derivation (KDF) [42].

4.3. Pseudorandom Number Generation (PRNG/PRF)

Sponge- and Duplex-based constructions leveraging large cryptographic permutations—such as Keccak- f , Ascon- p , or Xoodoo—have emerged as effective designs for generating pseudorandom bits or data streams of arbitrary length [39]. The Duplex construction is particularly well suited for reseeding pseudorandom number generators (PRNGs), as it naturally supports the absorption of fresh entropy into the internal state while preserving forward and backward security. This capability makes sponge/duplex-based designs attractive for long-lived cryptographic systems requiring periodic reseeding.

Several PRNG and pseudorandom function (PRF) constructions based on the sponge and duplex paradigms have been proposed in the literature [30, 72, 100, 182, 204]. These schemes typically achieve strong provable security guarantees in the ideal permutation model, assuming the underlying permutation behaves as a random permutation. Their ability to produce arbitrary-length output, combined with inherent stateful operation and simple domain separation, further strengthens their applicability to cryptographic protocols, key derivation, and randomness generation in both classical and post-quantum settings.

4.4. Trends and Challenges in the Standardization of Permutation-Based Cryptographic Primitives

4.4.1. Standardization of Large Permutations The NIST Lightweight Cryptography (LWC) project selected Ascon, a permutation-based primitive, as the primary standard for lightweight authenticated encryption and hashing in 2023 [59, 191]. This decision underscores the practical viability and maturity of permutation-based designs, particularly in resource-constrained environments. Although the NIST Post-Quantum Cryptography (PQC) effort [149] primarily targets public-key primitives, several hash-based digital signature schemes, such as SPHINCS+ and its variants [18, 116, 184], rely on small permutation-based hash functions (e.g., Haraka) as core components. This further illustrates the growing role of permutations across different layers of cryptographic standardization.

4.4.2. Design of Permutations for Large States There is increasing interest in the design of cryptographic permutations tailored for large internal state sizes (e.g., exceeding 512 bits or even reaching 1024 bits), driven by requirements for long-term security and domain-specific applications. Ensuring full diffusion over such large state spaces within a reasonable number

of rounds remains a central design challenge [148]. Designers must carefully balance security, performance, and implementation cost, particularly for high-security targets such as 256-bit classical security or 128-bit post-quantum security levels.

4.4.3. Quantum Cryptanalysis and Resistance Quantum algorithms introduce new considerations for symmetric cryptography. Grover's algorithm effectively reduces the security of exhaustive key search by a square root [78, 105, 124, 141, 176], while Simon's algorithm may exploit periodic or structured properties in permutation-based constructions to enable faster key recovery or collision attacks. Consequently, evaluating large cryptographic permutations under quantum adversary models—most notably the Q1 and Q2 models [188]—is essential for future-proofing their security.

4.4.4. Optimization for Emerging Platforms To maximize practical applicability, permutation-based designs must be efficiently implementable across a wide range of modern platforms, including SIMD-enabled and multi-core CPUs, GPUs, FPGAs, and ASICs [175]. This requirement favors round functions composed of hardware- and software-friendly operations, such as bitwise logic, rotations, and modular addition, as well as support for parallel execution. Techniques such as loop unrolling, pipelining, and cache-aware optimization further contribute to scalable performance.

4.4.5. Parameterized and Configurable Designs Flexible permutation designs with adjustable parameters—such as state size, number of rounds, or rate/capacity ratio—enable deployment across diverse security levels and application domains. Parameterization enhances reusability and allows a single permutation core to serve multiple cryptographic purposes, ranging from lightweight protocols to high-throughput hashing, PRNGs, and post-quantum-oriented applications.

5. CONCLUSION

Cryptographic permutations have emerged as foundational components in modern symmetric-key cryptography, serving as a unifying primitive for a wide range of functionalities, including hashing, authenticated encryption, pseudorandom number generation, and extendable-output functions. Their integration into frameworks such as Sponge, Duplex, and Farfalle has enabled the construction of compact, versatile, and efficient cryptographic systems, reducing implementation complexity while preserving strong and well-understood security guarantees. The paradigm shift from traditional keyed block ciphers to public, unkeyed permutations combined with provably secure modes of operation represents a significant evolution in the design philosophy of symmetric cryptography.

The growing demand for large-state cryptographic permutations—typically exceeding 256 bits—is driven by multiple factors, including the increasing scale of data processing, the proliferation of resource-constrained platforms such as IoT devices, and emerging security challenges posed by quantum computing. Designing such permutations entails non-trivial challenges: achieving rapid and complete diffusion over large state spaces, balancing security margins against performance constraints, and maintaining simplicity and configurability across heterogeneous hardware and software platforms. Continued advances in classical and quantum cryptanalysis further refine design criteria and underscore the necessity of conservative, forward-secure design choices.

Future research directions include the exploration of novel nonlinear and linear components with improved cryptographic properties, architecture-aware optimization techniques for both hardware and software implementations, and the systematic integration of post-quantum security considerations into permutation design and analysis. Standardization efforts, most notably the NIST Lightweight Cryptography initiative, play a crucial role in validating design paradigms, fostering interoperability, and accelerating real-world adoption.

As digital infrastructures continue to evolve, permutation-based cryptographic designs are well positioned to remain a cornerstone of next-generation symmetric cryptography, providing scalable, robust, and future-proof foundations for secure computation.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the contributions of the cryptographic research community for their ongoing efforts in advancing the understanding, analysis, and development of permutation-based cryptographic primitives.

REFERENCES

1. Abdulghani, H. A., Nijdam, N. A. & Konstantas, D. (2022) Analysis on security and privacy guidelines: RFID-based IoT applications, *IEEE Access*, **10**, 131528–131554. doi: 10.1109/ACCESS.2022.3224737
2. Ajala, O. A., Arinze, C. A. Ofodile, O. C., Okoye, C. C., & Daraojimba, A. I. (2024) Exploring and reviewing the potential of quantum computing in enhancing cybersecurity encryption methods, *Magna Scientia Advanced Research and Reviews*, **10**(1), 321–329. doi: 10.30574/msarr.2024.10.1.0038
3. Amiri Eliasi, P., Ghosh, K. & Daemen, J. (2024) Mystrium: Wide block encryption efficient on entry-level processors, *International Conference on Security and Cryptography for Networks*, Springer, 269–290. doi: 10.1007/978-3-031-71073-5_14
4. Amiri Eliasi, P., Ghosh, K. & Daemen, J. (2024) Koala: A low-latency pseudorandom function, *Proc. of International Conference on Selected Areas in Cryptography* (Quebec, Canada). doi: 10.1007/978-3-031-82852-2_11
5. Andreeva, E., Bilgin, B., Bogdanov, A., Luykx, A., Mennink, B., et al. (2015) APE: Authenticated permutation-based encryption for lightweight cryptography, *Proc. of FSE 2014* (Hong Kong), 168–186. doi: 10.1007/978-3-662-46706-0_9
6. Aoki, K. & Yasuda, K. (2013).= The security of the OCB mode of operation without the SPRP assumption, *Proc. of ProvSec 2013* (Malacca City, Malacca), 202–220. doi: 10.1007/978-3-642-41227-1_12
7. Aumasson, J.-P., Henzen, L., Meier, W. & Phan, R. C.-W. (2008) *SHA-3 proposal BLAKE*, [Online.] Available: <https://131002.net/blake/blake.pdf>
8. Aumasson, J.-P., Neves, S., Wilcox-O’Hearn, Z. & Winnerlein, C. (2014) The hash function BLAKE, *IACR Cryptology ePrint Archive*, Report 2014/1002.
9. Aumasson, J.-P., Neves, S., Wilcox-O’Hearn, Z. & Winnerlein, C. (2013) BLAKE2: Simpler, smaller, fast as MD5, *Proc. of International Conference on Applied Cryptography and Network Security* (Banff, Canada), 119–135. doi: 10.1007/978-3-642-38980-1_8
10. Băcuieți, N., Daemen, J., Hoffert, S., Van Assche, G. & Van Keer, R. (2022) Jammin’ on the deck, *Proc. of International Conference on the Theory and Application of Cryptology and Information Security* (Taipei City, Taiwan), 555–584. doi: 10.1007/978-3-031-22966-4_19
11. Barbero, S. (2024) An overview on cryptanalysis of ARX ciphers. *DE CIFRIS KOINE*, **10**.
12. Barker, E. & Kelsey, J. (2012) *Recommendation for random number generation using deterministic random bit generators*. NIST Special Publication 800-90A. Washington, DC: United States Department of Commerce. doi: 10.6028/NIST.SP.800-90a
13. Beierle, C. (2024) Invariant cryptanalysis. In: Boura, C. & Naya-Plasencia, M. (Eds.) *Symmetric Cryptography, Volume 2: Cryptanalysis and Future Directions*. New York, NY: Wiley (pp. 111–130).

14. Belkheyyar, Y., Daemen, J., et al. (2023) Introducing two low-latency cipher families: Sonic and SuperSonic, *IACR Cryptology ePrint Archive*, Report 2023/1716.
15. Bellare, M. & Rogaway, P. (2004) Code-based game-playing proofs and the security of triple encryption, *IACR Cryptology ePrint Archive*, Report 2004/331.
16. Bellini, E. & Makarim, R. H. (2022). Functional cryptanalysis: Application to reduced-round Xoodoo, *IACR Cryptology ePrint Archive*, Report 2022/289.
17. Bernstein, D. J. (2008) ChaCha, a variant of Salsa20. *Proc. of SASC 2008* (Lausanne, Switzerland), 1–6.
18. Bernstein, D. J., Hülsing, A., Kölbl, S., Niederhagen, R., Rijneveld, J., et al. (2019) The SPHINCS+ signature framework, *Proc. of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, (London, UK), 2129–2146. doi: 10.1145/3319535.3363229
19. Bernstein, D. J., Kölbl, S., Lucks, S., Massolino, P. M. C., Mendel, F., et al. (2017) Gimli: A cross-platform permutation, *Proc. of CHES 2017*, (Taipei City, Taiwan), 299–320. doi: 10.1007/978-3-319-66787-4_15
20. Bertoni, G., Daemen, J., Hoffert, S., Peeters, M., Van Assche, G., et al. (2016) Farfalle: Parallel permutation-based cryptography, *IACR Cryptology ePrint Archive*, Report 2016/1188.
21. Bertoni, G., Daemen, J., Peeters, M. & Van Assche, G. (2007) *Sponge functions*, [Online]. Available: <https://keccak.team/files/SpongeFunctions.pdf>
22. Bertoni, G., Daemen, J., Peeters, M. & Van Assche, G. (2011) Duplexing the sponge: Single-pass authenticated encryption and other applications, *Proc. of SAC 2011*, (Toronto, Canada), 320–337. doi: 10.1007/978-3-642-28496-0_19
23. Bertoni, G., Daemen, J., Peeters, M. & Van Assche, G. (2011) *The Keccak reference, version 3.0*, [Online]. Available: <https://keccak.team/files/Keccak-reference-3.0.pdf>
24. Bertoni, G., Daemen, J., Peeters, M. & Van Assche, G. (2013) Keccak, *Proc. of EUROCRYPT 2013*, (Athens, Greece), 313–314. doi: 10.1007/978-3-642-38348-9_19
25. Bertoni, G., Daemen, J., Peeters, M. & Van Assche, G. (2009) *Keccak sponge function family main document*, [Online]. Available: <https://keccak.team/files/Keccak-main-2.0.pdf>
26. Bertoni, G., Daemen, J., Peeters, M. & Van Assche, G. (2008) On the indistinguishability of the sponge construction, *Proc. of EUROCRYPT 2008*, (Istanbul, Turkey), 181–197. doi: 10.1007/978-3-540-78967-3_11
27. Bertoni, G., Daemen, J., Peeters, M. & Van Assche, G. (2012) Permutation-based encryption, authentication and authenticated encryption, *Proc. of DIAC 2012* (Stockholm, Sweden), 159–170.
28. Bertoni, G., Daemen, J., Peeters, M. & Van Assche, G. (2011) *Cryptographic sponge functions*, [Online]. Available: <https://keccak.team/files/SpongeFunctions.pdf>
29. Bertoni, G., Daemen, J., Peeters, M. & Van Assche, G. (2012) The sponge and duplex constructions, *ECRYPT Workshop on Tools for Cryptanalysis*.
30. Bertoni, G., Daemen, J., Peeters, M. & Van Assche, G. (2010) Sponge-based pseudo-random number generators, *Proc. of CHES 2010*, (Santa Barbara, CA), 33–47. doi: 10.1007/978-3-642-15031-9_3
31. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G. & Van Keer, R. (2018) The authenticated encryption schemes Kravatte-SANE and Kravatte-SANSE, *IACR Cryptology ePrint Archive*, Report 2018/1012.
32. Bhoi, S. S., Arakala, A., Corman, A. B. & Rao, A. (2025) Post-quantum homomorphic encryption: A case for code-based alternatives, *Cryptography*, **9**(3), 31. doi: 10.3390/cryptography9030031
33. Biham, E., & Shamir, A. (2011) *Differential Cryptanalysis of the Data Encryption Standard*. Cham, Switzerland: Springer.
34. Biryukov, A. (2025) Substitution–permutation (SP) network. In: Jajodia, S., Samarati, P. & Yung, M. (Eds.) *Encyclopedia of Cryptography, Security and Privacy*, Cham,

- Switzerland: Springer. doi: 10.1007/978-3-030-71522-9_300459
35. Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K., et al. (2011) SPONGENT: A lightweight hash function, *Proc. of CHES 2011*, (Nara, Japan), 312–325. doi: 10.1007/978-3-642-23951-9_21
 36. Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K., et al. (2012) SPONGENT: The design space of lightweight cryptographic hashing. *IEEE Transactions on Computers*, **62**(10), 2041–2053. doi: 10.1109/TC.2012.196
 37. Boissier, R. H., Noûs, C. & Rotella, Y. (2021) Algebraic collision attacks on Keccak, *IACR Transactions on Symmetric Cryptology*, **2021**(1), 239–268. doi: 10.46586/tosc.v2021.i1.239-268
 38. Borowski, M. (2013) The sponge construction as a source of secure cryptographic primitives, *Proc. of 2023 Military Communications and Information Systems Conference*, (Canberra, Australia), 1–6. doi: 10.1109/MilCIS.2013.6658239
 39. Borowski, M. (2014) Cryptographic applications of the duplex construction, *Annales Universitatis Mariae Curie-Skłodowska, sectio AI-Informatica*, **14**(1), 37–48. doi: 10.2478/umcsinfo-2014-0004
 40. Cai, J., et al. (2019) Zero-sum distinguishers for round-reduced GIMLI permutation, *Proc. of ICISSP 2025* (Prague, Czech Republic), 361–369. doi: 10.5220/0007371603610369
 41. Castro, J. C. H., Sierra, J. M., Seznec, A., Izquierdo, A. & Ribagorda, A. (2005) The strict avalanche criterion randomness test, *Mathematics and Computers in Simulation*, **68**(1), 1–7. doi: 10.1016/j.matcom.2004.09.001
 42. Chakraborty, A. & Biswas, A. (2025) A comprehensive and systematic analysis of one-way hashing and its advancements, *Proc. of 2025 3rd International Conference on Intelligent Systems, Advanced Computing and Communication* (Silchar, India), 1–6. doi: 10.1109/ISACC62637.2025.10952473
 43. Cheng, H. & Ding, Q. (2012) Overview of the block cipher, *Proc. of Second International Conference on Instrumentation, Measurement, Computer, Communication and Control*, (Heilongjiang, China), 1034–1038. doi: 10.1109/IMCCC.2012.248
 44. Cui, T., Zhang, Y., Zhang, J., Jin, C. & Chen, S. (2024) Differential-invariant subspace cryptanalysis—A real-time attack against IoT-friendly word-based block ciphers, *IEEE Internet of Things Journal*, **11**(10), 17642–17650. doi: 10.1109/JIOT.2024.3358346
 45. Czajkowski, J., Groot Bruinderink, L., Hülsing, A., Schaffner, C. & Unruh, D. (2018) Post-quantum security of the sponge construction, *Proc. of International Conference on Post-Quantum Cryptography 2018*, (Fort Lauderdale, FL), 185–204. doi: 10.1007/978-3-319-79063-3_9
 46. Daemen, J. (2024) Deck-function-based cryptography. In: Boura, C. & Naya-Plasencia, M. (Eds.) *Symmetric Cryptography, Volume 2: Cryptanalysis and Future Directions*. New York, NY: Wiley (pp. 227–248). doi: 10.1002/9781394176212.ch11
 47. Daemen, J., Hoffert, S., Peeters, M., Van Assche, G. & Van Keer, R. (2020) Xoodyak, a lightweight cryptographic scheme, *IACR Transactions on Symmetric Cryptology*, **2020**(S1), 60–87. doi: 10.13154/tosc.v2020.iS1.60-87
 48. Daemen, J., Hoffert, S., Peeters, M., Van Assche, G. & Van Keer, R. (2020) Xoodyak, an update, [Online]. Available: <https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/status-update-sep2020/Xoodyak-update.pdf>.
 49. Daemen, J., Hoffert, S. & Van Assche, G. (2024) Shaking up authenticated encryption, *IACR Cryptology ePrint Archive*, Report 2024/1618.
 50. Daemen, J., Hoffert, S., Van Assche, G. & Van Keer, R. (2018) The design of Xoodoo and Xoofff, *IACR Transactions on Symmetric Cryptology*, **2018**(4), 1–38. doi: 10.13154/tosc.v2018.i4.1-38
 51. Daemen, J., Hoffert, S., Van Assche, G. & Van Keer, R. (2018) Xoodoo cookbook, *IACR Cryptology ePrint Archive*, Report 2018/767.

52. Daemen, J., Massolino, P. M. C. & Rotella, Y. (2020) The Subterranean 2.0 cipher suite, *IACR Transactions on Symmetric Cryptology*, **2020**(S1), 262–294. doi: 10.13154/tosc.v2020.iS1.262-294
53. Daemen, J., Mennink, B. & Van Assche, G. (2017) Full-state keyed duplex with built-in multi-user support, *Proc. of ASIACRYPT 2017*, (Hong Kong), 606–637. doi: 10.1007/978-3-319-70697-9_21
54. Daemen, J. & Rijmen, V. (1998) AES proposal: Rijndael, *Proc. of First Advanced Encryption Standard Conference*, (Ventura, CA), 1–45.
55. de Alencar, M. S. (2022) *Cryptography and Network Security*. Gistrup, Denmark: River Publishers.
56. Dhanda, S. S., Singh, B., & Jindal, P. (2020) Lightweight cryptography: A solution to secure IoT, *Wireless Personal Communications*, **112**(3), 1947–1980. doi: 10.1007/s11277-020-07134-3
57. Dimitrov, M. M. (2020) On the design of chaos-based S-boxes, *IEEE Access*, **8**, 117173–117181. doi: 10.1109/ACCESS.2020.3004046
58. Dinu, D., Perrin, L., Udovenko, A., Velichkov, V., Großschädl, J., et al. (2016) Design strategies for ARX with provable bounds: Sparx and LAX, *Proc. of ASIACRYPT 2016*, (Hanoi, Vietnam), 484–513. doi: 10.1007/978-3-662-53887-6_18
59. Dobraunig, C., Eichlseder, M., Mendel, F. & Schläffer, M. (2021) Ascon v1.2: Lightweight authenticated encryption and hashing, *Journal of Cryptology*, **34**(3), 1–42. doi: 10.1007/s00145-021-09398-9
60. Dodis, Y., et al. (2017) Provable security of substitution-permutation networks, *IACR Cryptology ePrint Archive*, Report 2017/016.
61. Dolmeta, A., Martina, M. & Masera, G. (2023) Comparative study of Keccak SHA-3 implementations, *Cryptography*, **7**(4), 60. doi: 10.3390/cryptography7040060
62. Dunkelman, O. & Proofs, S. (2023) Block ciphers. In: Boura, C. & Naya-Plasencia, M. (Eds.) *Symmetric Cryptography, Volume 1: Design*. New York, NY: Wiley (pp. 39–68).
63. Dunkelman, O. & Weizman, A. (2022) *Differential-linear cryptanalysis on Xoodyak*, [Online]. Available: <https://csrc.nist.gov/presentations/2022/differential-linear-cryptanalysis-on-xoodyak>
64. El Hirsch, S., et al. (2023) Twin column parity mixers and Gaston: A new mixing layer and permutation, *Proc. of Annual International Cryptology Conference*, (Santa Barbara, CA), 3–34. doi: 10.1007/978-3-031-38548-3_1
65. Esa, N. F. M., Abdul-Latip, S. F. & Baharon, M. R. (2019) A survey of ARX-based symmetric-key primitives, *International Journal of Communication Networks and Information Security*, **11**(3), 397–408.
66. Fan, S., et al. (2024) Taiyi: A high-performance CKKS accelerator for practical fully homomorphic encryption, *IACR Cryptology ePrint Archive*, Report 2024/1603.
67. Faruk, M. J. H., Tahora, S., Tasnim, M., Shahriar, H. & Sakib, N. (2022) A review of quantum cybersecurity: Threats, risks and opportunities, *Proc. of 2022 1st International Conference on AI in Cybersecurity*, (Victoria, TX), 1–8. doi: 10.1109/ICAIC53980.2022.9896960
68. Firmansyah, B. & Bansal, R. (2024). Standardization and regulatory challenges in modern cryptography. In: Gupta, B. B. (Eds.) *Metaverse Security Paradigms*. Hershey, PA: IGI Global (pp. 145–183). doi: 10.4018/979-8-3693-1339-6.ch007
69. Fuchs, J., Rotella, Y. & Daemen, J. (2023) On the security of keyed hashing based on public permutations, *Proc. of CRYPTO 2023*, (Santa Barbara, CA), 220–250. doi: 10.1007/978-3-031-38554-4_8
70. Fu, Q., et al. (2024) Preimage attacks on Xoodyak and Gaston based on algebraic strategies, *IACR Cryptology ePrint Archive*, Report 2024/121.
71. Garg, S. & Upadhyay, D. (2013) S-box design approaches: Critical analysis and future directions, *International Journal of Advanced Research in Computer Science and Electronic Engineering*, **2**(4), 426–430.

72. Gaži, P. & Tessaro, S. (2016) Provably robust sponge-based PRNGs and KDFs, *Proc. of Annual International Conference on the EUROCRYPT 2016*, (Vienna, Austria), 87–116. doi: 10.1007/978-3-662-49890-3_4
73. Ghosh, K., Amiri Eliasi, P. & Daemen, J. (2023) Multimixer-128: Universal keyed hashing based on integer multiplication, *IACR Cryptology ePrint Archive*, Report 2023/1358.
74. Ghosh, K., Amiri Eliasi, P. & Daemen, J. (2023) Multimixer-156: Universal keyed hashing based on integer multiplication and cyclic shift, *Proc. of INDOCRYPT 2023*, (Goa, India), 370–393. doi: 10.1007/978-3-031-56233-4_19
75. Gookyi, D. A. N., Kanda, G. & Ryoo, K. (2021) NIST lightweight cryptography standardization process: Classification of second round candidates, open challenges, and recommendations, *Journal of Information Processing Systems*, **17**(2), 253–270. doi: 10.3745/JIPS.03.0157
76. Grassi, L. (2025) On generalizations of the Lai–Massey scheme, *Designs, Codes and Cryptography*, 1–36. doi: 10.1007/s10623-025-01567-0
77. Grassi, L., Rechberger, C. & Rønjom, S. (2020) On a generalization of substitution-permutation networks: The HADES design strategy, *Proc. of EUROCRYPT 2020*, (Zagreb, Croatia), 674–704. doi: 10.1007/978-3-030-45724-2_23
78. Grassl, M., Langenberg, B., Roetteler, M. & Steinwandt, R. (2016) Applying Grover’s algorithm to AES: Quantum resource estimates, *Proc. of PQCrypto 2016*, (Fukuoka, Japan), 29–43. doi.: 10.1007/978-3-319-29360-8_3
79. Grosek, O. & Zajac, P. (2007) Efficient selection of the AES-class MixColumns parameters, *WSEAS Transactions on Information Science and Applications*, **4**(4), 663–668.
80. Gulliver, T. A., Kim, J.-L. & Lee, Y. (2008) New MDS or near-MDS self-dual codes, *IEEE Transactions on Information Theory*, **54**(9), 4354–4360. doi: 10.1109/TIT.2008.926442
81. Gunathilake, N. A., Buchanan, W. J. & Asif, R. (2019) Next generation lightweight cryptography for smart IoT devices: Implementation, challenges and applications, *Proc. of IEEE 5th World Forum on Internet of Things*, (Limerick, Ireland), 707–710. doi: 10.1109/WF-IoT.2019.8767227
82. Gundaram, P. K. (2024) Cryptanalysis of selected ARX-based block ciphers, *Vietnam Journal of Computer Science*, **11**(4), 553–568. doi: 10.1142/S2196888824500141
83. Günsing, A., Daemen, J. & Mennink, B. (2019) Deck-based wide block cipher modes and an exposition of the blinded keyed hashing model, *IACR Transactions on Symmetric Cryptology*, **2019**(4), 1–22. doi: 10.13154/tosc.v2019.i4.1-22
84. Guo, J., et al. (2020) Practical collision attacks against round-reduced SHA-3, *Journal of Cryptology*, **33**(1), 228–270. doi: 10.1007/s00145-019-09313-3
85. Guo, J., et al. (2022) Exploring SAT for cryptanalysis: (Quantum) collision attacks against 6-round SHA-3 (full version), *IACR Cryptology ePrint Archive*, Report 2022/225.
86. Guo, J., et al. (2014) Analysis of BLAKE2, *Proc. of CT-RSA 2014*, (San-Francisco, CA), 402–423. doi: 10.1007/978-3-319-04852-9_21
87. Guo, J., Peyrin, T. & Poschmann, A. (2011) The PHOTON family of lightweight hash functions, *Proc. of CRYPTO 2011*, (Santa-Barbara, CA), 222–239. doi: 10.1007/978-3-642-22792-9_13
88. Gupta, K. C., et al. (2024) On the construction of near-MDS matrices, *Cryptography and Communications*, **16**(2), 249–283. doi: 10.1007/s12095-023-00649-4
89. Gupta, K. C., Pandey, S. K. & Samanta, S. (2023) On the direct construction of MDS and near-MDS matrices, *Advances in Mathematics of Communications*, **18**(2), 267–282. doi: 10.3934/amc.2023024
90. Gupta, K. C. & Ray, I. G. (2015) Cryptographically significant MDS matrices based on circulant and circulant-like matrices for lightweight applications, *Cryptography and*

- Communications*, **7**, 257–287. doi: 10.1007/s12095-014-0114-5
91. Gupta, S. (2024) *Advanced Encryption Techniques for Database Security in Cloud Environments*. Cham, Switzerland: Springer.
 92. Han, K., Lee, W.-K. & Hwang, S. O. (2022) cuGimli: Optimized implementation of the Gimli authenticated encryption and hash function on GPU for IoT applications, *Cluster Computing*, **25**, 1–18. doi: 10.1007/s10586-021-03415-z
 93. Hasan, L. F., et al. (2021) Variable rounds block cipher algorithm design, *Journal of Cybersecurity and Information Management*, **2**(2), 1–17.
 94. He, L., et al. (2022) Improved preimage attacks on round-reduced Keccak-384/512, *IACR Cryptology ePrint Archive*, Report 2022/1022.
 95. Hernandez, J. C., Sierra, J. M. & Seznec, A. (2004). The SAC test: A new randomness test, with some applications to PRNG analysis, *Proc. of ICCSA 2004* (Assisi, Italy), 960–967. doi: 10.1007/978-3-540-24767-8_100
 96. Hoang, V. T. & Rogaway, P. (2010) On generalized Feistel networks, *Proc. of CRYPTO 2010*, (Santa Barbara, CA), 613–630. doi: 10.1007/978-3-642-14623-7_33
 97. Huang, M., Xu, Z. & Wang, L. (2022) On the probability and automatic search of rotational-XOR cryptanalysis on ARX ciphers, *The Computer Journal*, **65**(12), 3062–3080. doi: 10.1093/comjnl/bxab119
 98. Huang, S., et al. (2022) Finding collisions against 4-round SHA-3-384 in practical time, *IACR Transactions on Symmetric Cryptology*, **2022**(4), 239–270. doi: 10.46586/tosc.v2022.i4.239-270
 99. Hu, K., et al. (2023) Revisiting higher-order differential-linear attacks from an algebraic perspective, *Proc. of ASIACRYPT 2022*, (Taipei City, Taiwan), 215–243. doi: 10.1007/978-3-031-22969-5_8
 100. Hutchinson, D. (2016) A robust and sponge-like PRNG with improved efficiency, *Proc. of SAC 2016*, (St. John's, Canada), 381–398. doi: 10.1007/978-3-319-69453-5_21
 101. Inoue, A., Iwata, T., Minematsu, K. & Poettering, B. (2020) Cryptanalysis of OCB2: Attacks on authenticity and confidentiality, *Journal of Cryptology*, **33**(4), 1871–1913. doi: 10.1007/s00145-020-09359-8
 102. International Organization for Standardization (2025) *Information technology — Security techniques — Random bit generation*, [Online]. Available: <https://www.iso.org/standard/81645.html>
 103. Iwata, T. & Seurin, Y. (2017) Reconsidering the security bound of AES-GCM-SIV, *IACR Transactions on Symmetric Cryptology*, **2017**(4), 240–267. doi: 10.13154/tosc.v2017.i4.240-267
 104. Janson, C. & Struck, P. (2022) Sponge-based authenticated encryption: Security against quantum attackers, *Proc. of PQCrypto 2022*, (Online), 209–229. doi: 10.1007/978-3-031-17234-2_11
 105. Jaques, S., Naehrig, M., Roetteler, M. & Virdia, F. (2020) Implementing Grover oracles for quantum key search on AES and LowMC, *Proc. of EUROCRYPT 2020*, (Online), 280–310. doi: 10.1007/978-3-030-45724-2_10
 106. Kaur, J., et al. (2025) A survey on the implementations, attacks, and countermeasures of the NIST lightweight cryptography standard: ASCON, *IACR Cryptology ePrint Archive*, Report 2025/149.
 107. Kelsey, J., Chang, S. & Perlner, R. (2016) *SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash, and ParallelHash*, [Online]. Available: NIST Special Publication 800-185, National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-185>
 108. Kelsey, J., Chang, S.-J. & Perlner, R. (2016). SHA-3 derived functions: cSHAKE, KMAC, TupleHash, and ParallelHash, *NIST Special Publication*, 800-185. doi: 10.6028/NIST.SP.800-185
 109. Khan, S., Krishnamoorthy, P., Goswami, M., Rakhimjonovna, F. M., Mohammed, S. A., et al. (2024) Quantum computing and its implications for cybersecurity: A

- comprehensive review of emerging threats and defenses, *Nanotechnology Perceptions*, **20**(S13), 1232–1248. doi: 10.62441/nano-ntp.v20iS13.79
110. Khovratovich, D. & Nikolić, I. (2010) Rotational cryptanalysis of ARX. *Proc. of FSE 2010*, (Santa Fe, NM), 333–346. doi: 10.1007/978-3-642-13858-4_19
 111. Khovratovich, D., Nikolić, I. & Pieprzyk, J. (2015) Rotational cryptanalysis of ARX revisited, *Proc. of 22nd International Workshop FSE 2015*, (Istanbul, Turkey), 519–536. doi: 10.1007/978-3-662-48116-5_25
 112. Knudsen, L. R. (1999) Block ciphers—A survey. In: Preneel, B. & Rijmen, V. (Eds.) *State of the Art in Applied Cryptography: Course on Computer Security and Industrial Cryptography*. Leuven, Belgium: Springer (pp. 18–48). doi: 10.1007/3-540-49248-8_2
 113. Knudsen, L., Ross, A. & Biham, E. (1998) Serpent: A proposal for the Advanced Encryption Standard, *Proc. of the First Advanced Encryption Standard Conference* (Ventura, CA).
 114. Kong, J. H., Ang, L.-M. & Seng, K. P. (2015) A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments, *Journal of Network and Computer Applications*, **49**, 15–50. doi: 10.1016/j.jnca.2014.09.006
 115. Kra, I. & Simanca, S. R. (2012) On circulant matrices, *Notices of the AMS*, **59**(3), 368–377.
 116. Kudinov, M., et al. (2022) SPHINCS+ C: Compressing SPHINCS+ with (almost) no cost, *IACR Cryptology ePrint Archive*, Report 2022/1584.
 117. Kumar, A. & Tejani, S. (2018) S-box architecture, *Proc. of International Conference on Futuristic Trends in Network and Communication Technologies* (Solan, India), 339–351. doi: 10.1007/978-981-13-3121-3_32
 118. Kumar, Y., et al. (2024) Construction of all MDS and involutory MDS matrices, *IACR Cryptology ePrint Archive*, Report 2024/347.
 119. Kuznetsov, A., et al. (2024) SBGen: A high-performance library for rapid generation of cryptographic S-boxes, *SoftwareX*, **27**, 101788. doi: 10.1016/j.softx.2024.101788
 120. Lai, X. (1994) Higher order derivatives and differential cryptanalysis. In: Blahut, R. E., Costello, D. J., Maurer, U. & Mittelholzer, T. (Eds.) *Communications and Cryptography: Two Sides of One Brain*. Boston, MA : Kluwer Academic Publishers (pp. 227–233). doi: 10.1007/978-1-4615-2694-0_23
 121. Lai, X. (1992). *On the Design and Security of Block Ciphers*, *PhD Thesis*, ETH Zurich.
 122. Lai, X. & Massey, J. L. (1990) A proposal for a new block encryption standard, *Proc. of EUROCRYPT '90* (Aarhus, Denmark), 389–404. doi: 10.1007/3-540-46877-3_35
 123. Leander, G., Abdelraheem, M. A., AlKhazaimi, H. & Zenner, E. (2011) A cryptanalysis of PRINTcipher: The invariant subspace attack., *Proc. of CRYPTO 2011* (Santa Barbara, CA), 206–221. doi: 10.1007/978-3-642-22792-9_12
 124. Leander, G. & May, A. (2017) Grover meets Simon—quantumly attacking the FX-construction, *Proc. of ASIACRYPT 2017* (Hong Kong), 161–178. doi: 10.1007/978-3-319-70697-9_6
 125. Lefevre, C., Belkheyar, Y. & Daemen, J. (2023) Kirby: A robust permutation-based PRF construction, *IACR Cryptology ePrint Archive*, Report 2023/1520.
 126. Lefevre, C. & Beltrán, M. M. (2025) MacaKey: Full-state keyed sponge meets the summation-truncation hybrid, *IACR Cryptology ePrint Archive*, Report 2025/212.
 127. Lei, H., et al. (2024) Symmetric twin column parity mixers and their applications, *IACR Transactions on Symmetric Cryptology*, **2024**(4), 1–37. doi: 10.46586/tosc.v2024.i4.1-37
 128. Leurent, G. (2012) Analysis of differential attacks in ARX construction, *Proc. of ASIACRYPT 2012* (Beijing, China), 226–243. doi: 10.1007/978-3-642-34961-4_15
 129. Leurent, G. (2013) Construction of differential characteristics in ARX designs: Application to Skein, *Proc. of CRYPTO 2013* (Santa Barbara, CA), 241–258. doi: 10.1007/978-3-642-40041-4_14

130. Li, M. & Cheng, L. (2017) Distinguishing property for full round Keccak-f permutation, *Proc. of CISIS 2017* (Turin, Italy), 636–644. doi: 10.1007/978-3-319-61566-0_60
131. Lin, X., He, L. & Yu, H. (2021) Improved preimage attacks on 3-round Keccak-224/256, *IACR Transactions on Symmetric Cryptology*, **2021**(1), 84–101. doi: 10.46586/tosc.v2021.i1.84-101
132. Liu, F., et al. (2020) Algebraic attacks on round-reduced Keccak/Xoodoo, *IACR Cryptology ePrint Archive*, Report 2020/346.
133. Liu, G., Qiu, W. & Tu, Y. (2019) New techniques for searching differential trails in Keccak, *IACR Transactions on Symmetric Cryptology*, **2019**(4), 407–437. doi: 10.13154/tosc.v2019.i4.407-437
134. Liu, M. & Sim, S. M. (2016) Lightweight MDS generalized circulant matrices, *Proc. of FSE 2016* (Bochum, Germany), 101–120. doi: 10.1007/978-3-662-52993-5_6
135. Liu, T., Tessaro, S. & Vaikuntanathan, V. (2021) The t-wise independence of substitution-permutation networks, *Proc. of CRYPTO 2021* (Santa Barbara, CA), 454–483. doi: 10.1007/978-3-030-84259-8_16
136. Liu, Y., Sun, S. & Li, C. (2021) Rotational cryptanalysis from a differential-linear perspective: Practical distinguishers for round-reduced FRIET, Xoodoo, and Alzette, *Proc. of EUROCRYPT 2021* (Zagreb, Croatia), 741–770. doi: 10.1007/978-3-030-77870-5_26
137. Li, X. & Wu, W. (2021) Constructions of iterative near-MDS matrices with the lowest XOR count, *Proc. of ACISP 2021* (Perth, Australia), 170–189. doi: 10.1007/978-3-030-90567-5_9
138. Lu, C.-Y., et al. (2012) Cryptanalysis on PHOTON hash function using cube attack, *Proc. of ISIC 2012* (Yunlin, Taiwan), 18–21. doi: 10.1109/ISIC.2012.6449711
139. Luong, T. T. & Linh, H. D. (2024) Generating key-dependent involutory MDS matrices through permutations, direct exponentiation, and scalar multiplication, *International Journal of Information and Computer Security*, **23**(4), 410–432. doi: 10.1504/IJICS.2024.138194
140. Malakhov, S. S. (2025) On the circulant matrix MDS testing and the search for circulant MDS matrices, *Cryptography and Communications*, **17**(1), 87–119. doi: 10.1007/s12095-024-00740-6
141. Mandal, S., Mukherjee, D. & Bhattacharya, S. (2024) Implementing Grover’s algorithm on AES-based AEAD schemes, *Cryptography*, **14**(1), 21105. doi: 10.3390/cryptography8010004
142. Mansoor, K., et al. (2019) Securing IoT-based RFID systems: A robust authentication protocol using symmetric cryptography, *Sensors*, **19**(21), 4752. doi: 10.3390/s19214752
143. Matsuda, S. & Moriai, S. (2012) Lightweight cryptography for the cloud: Exploit the power of bitslice implementation, *Proc. of CHES 2012* (Leuven, Belgium), 408–425. doi: 10.1007/978-3-642-33027-8_24
144. Matsui, M. (1993) Linear cryptanalysis method for DES cipher, *Proc. of EUROCRYPT 1993* (Lofthus, Norway), 386–397. doi: 10.1007/3-540-48285-7_33
145. Maulidzart, A. V. T., et al. (2023) The hybrid cryptographic algorithms for secure RFID data protection in the Internet of Things, *Jurnal ELTIKOM: Jurnal Teknik Elektro, Teknologi Informasi dan Komputer*, **7**(2), 160–169. doi: 10.31961/eltikom.v7i2.718
146. Meier, W. (1994) On the security of the IDEA block cipher, *Proc. of EUROCRYPT 1993* (Lofthus, Norway), 371–385. doi: 10.1007/3-540-48285-7_32
147. Mennink, B. (2022) Understanding the duplex and its security, *IACR Cryptology ePrint Archive*, Report 2022/1136.
148. Mohammed, S. A., et al. (2024) *Cryptography and Network Security in Modern Era*. Bhopal, India: Academic Guru Publishing House.
149. Moody, D. (2021) NIST PQC standardization update, *Proc. of Third PQC Standardization Conference* (Online).

150. Mouha, N., Mennink, B., Van Herrewege, A., Watanabe, D., Preneel, B., et al. (2014) Chaskey: An efficient MAC algorithm for 32-bit microcontrollers, *Proc. of SAC 2014* (Montreal, Quebec, Canada), 306–323. doi: 10.1007/978-3-319-13051-4_19
151. Mousavi, M., Esmaeili, M. & Gulliver, T. A. (2025) Circulant, circulant-like and orthogonal MDS generalized Cauchy matrices, *Advances in Mathematics of Communications*, **19**(2), 716–735. doi: 10.3934/amc.2023014
152. Müller, S. (2024) Ascon-Keccak AEAD algorithm, *IACR Cryptology ePrint Archive*, Report 2024/1615.
153. Nachev, V., Patarin, J. & Volte, E. (2017) *Feistel Ciphers*. Chem, Switzerland: Springer. doi: 10.1007/978-3-319-49532-4
154. Naito, Y. & Ohta, K. (2014) Improved indifferentiable security analysis of PHOTON, *Proc. of SAC 2014* (Montreal, Quebec, Canada), 340–357. doi: 10.1007/978-3-319-10879-7_20
155. Nam, T. S., Van Long, N. & Cuong, N. B. (2023) An optimized bit-slice implementation of secure 8-bit Sbox based on butterfly structure, *Proc. of 2023 15th International Conference on Knowledge and Systems Engineering* (Hanoi, Vietnam), 1–6. doi: 10.1109/KSE59135.2023.10299507
156. Nazish, M., Banday, M. T., Syed, I. & Banday, S. (2023) An efficient permutation approach for SbPN-based symmetric block ciphers, *Cybersecurity*, **6**(1), 42. doi: 10.1186/s42400-023-00174-9
157. NIST (2015). *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. FIPS PUB 202, National Institute of Standards and Technology. doi: 10.6028/NIST.FIPS.202
158. Nyberg, K. (1993) Differentially uniform mappings for cryptography, *Proc. of EUROCRYPT 1993* (Lofthus, Norway), 55–64. doi: 10.1007/3-540-48285-7_6
159. Nyberg, K. (1996) Generalized Feistel networks., *Proc. of EUROCRYPT 1993* (Lofthus, Norway), 91–104. doi: 10.1007/BFb0034841
160. O'Connor, J., et al. (2021) *BLAKE3: One function, fast everywhere*, [Online]. Available: <https://blake3.io>
161. Olson, B. J., Shaw, S. W., Shi, C., Pierre, C. & Parker, R. G. (2014) Circulant matrices and their application to vibration analysis, *Applied Mechanics Reviews*, **66**(4), 040803. doi: 10.1115/1.4027722
162. Olutimehin, A. T., Joseph, S. A., Ajayi, A. J., Metibemu, O. C., Balogun, A. Y., et al. (2025) Future-proofing data: Assessing the feasibility of post-quantum cryptographic algorithms to mitigate “harvest now, decrypt later” attacks, *Archives of Current Research International*, **25**(3), 60–80. doi: 10.9734/acri/2025/v25i31098
163. Omland, T. & Stanica, P. (2022) Permutation rotation-symmetric S-boxes, liftings and affine equivalence, *IACR Cryptology ePrint Archive*, Report 2022/109.
164. Ozaif, M., et al. (2025) A secure and efficient identity-based RFID mutual authentication scheme for IoT using elliptic curve cryptography, *Journal of Ambient Intelligence and Humanized Computing*, **47**(5), 424–437.
165. Pardo, J. L. G. & Gómez-Rodríguez, C. (2015) *The SHA-3 family of cryptographic hash functions and extendable-output functions*, [Online]. Available: <https://www.maplesoft.com/applications/detail.aspx?id=153903&L=E>
166. Prajapati, P., et al. (2025) Comparative study on hash algorithms: KECCAK, MD, RIPEMD, and SHA variants, *BIO Web of Conferences*, **97**(4), 00093. doi: 10.1007/978-981-97-0846-3_27
167. Preston, R. H. (2023) Applying Grover’s algorithm to hash functions: A software perspective, *IEEE Transactions on Quantum Engineering*, **3**, 1–10. doi: 10.1109/TQE.2022.3225548
168. Ramesh, S. M. & AlKhazimi, H. (2019) Side channel analysis of SPARX-64/128: Cryptanalysis and countermeasures, *Proc. of AFRICACRYPT 2019* (Rabat, Morocco), 215–235. doi: 10.1007/978-3-030-23696-0_12

169. Rana, M., Mamun, Q. & Islam, R. (2022) Lightweight cryptography in IoT networks: A survey, *Future Generation Computer Systems*, **129**, 77–89. doi: 10.1016/j.future.2021.11.011
170. Ranea, A., Liu, Y. & Ashur, T. (2020) An easy-to-use tool for rotational-XOR cryptanalysis of ARX block ciphers, *IACR Cryptology ePrint Archive*, Report 2020/1056.
171. Rane, D. D. (2016) Superiority of Twofish over Blowfish, *International Journal of Scientific Research*, **4**(11), 4744–4746.
172. Rosulek, M. (2021). *The Joy of Cryptography*. Corvallis, OR: Oregon State University.
173. Sadkhan, S. B. & Salman, A. O. (2018) A survey on lightweight-cryptography status and future challenges, *Proc. of 2018 International Conference on Advance of Sustainable Engineering and its Application* (Wasit, Iraq), 1–5. doi: 10.1109/ICASEA.2018.8370956
174. Samanta, S. (2024) On the counting of involutory MDS matrices, *Advances in Mathematics of Communications*, **19**(3), 922–941. doi: 10.3934/amc.2024018
175. Sampathkumar, R. (2024) GPU and ASIC as a boost for high performance computing. In: Singh, R., Sharma, N., Lotti, T. & Trehan, N. (Eds.) *High Performance Computing in Biomimetics: Modeling, Architecture and Applications*. Singapore: Springer (pp. 205–221). doi: 10.1007/978-981-99-7717-1_10
176. Sarah, D. & Peter, C. (2024) On the practical cost of Grover for AES key recovery. *Proc. of the 5th NIST PQC Standardization Conference* (Rockville, MD).
177. Shah, M. (2020) Parallel AES diffusion inter block diffusion at bit level and compression, *International Journal of Computer Applications*, **176**(22), 1–6.
178. Shamsabad, M. M. & Dehnavi, S. M. (2020) Lai-Massey scheme revisited, *IACR Cryptology ePrint Archive*, Report 2020/1241.
179. Shannon, C. E. (1949) Communication theory of secrecy systems, *Bell System Technical Journal*, **28**(4), 656–715. doi: 10.1002/j.1538-7305.1949.tb00928.x
180. Silva, T., et al. (2021) MDS matrices for cryptography, *Cryptography*, **5**(3), 17.
181. Singh, H. (2024) Managing the quantum cybersecurity threat: Harvest now, decrypt later. In: ed. Hammoudeh, M., Alessa, A. T., Sherbeeni, A. M., et al. (Eds.) *Quantum Computing*. New York, NY: CRC Press (pp. 142–158).
182. Singh, P. K., et al. (2019) JSpongeGen: A pseudo random generator for low resource devices, *Proc. of the Distributed Computing and Internet Technology: 15th International Conference*, (Bhubaneswar, India), 153–164. doi: 10.1007/978-3-030-05366-6_12
183. Sklavos, N. & Kitsos, P. (2010). BLAKE hash function family on FPGA: From the fastest to the smallest, *Proc. of ISVLSI 2010* (Lefkosia/Limassol, Cyprus), 151–156. doi: 10.1109/ISVLSI.2010.67
184. Soni, D., et al. (2021) SPHINCS+, *CRYPTO 2021* (Santa Barbara), CA, 141–162.
185. Standaert, F.-X. (2010) Introduction to side-channel attacks. In: Verbauwhede, I. (Ed.) *Secure Integrated Circuits and Systems*. Boston, MA: Springer (pp. 27–42). doi: 10.1007/978-0-387-71829-3_2
186. Stoffelen, K. & Daemen, J. (2018) Column parity mixers, *IACR Transactions on Symmetric Cryptology*, **2018**(1), 126–159. doi: 10.13154/tosc.v2018.i1.126-159
187. Subroto, R. C. (2023) An algebraic approach to circulant column parity mixers, *IACR Cryptology ePrint Archive*, Report 2023/1300.
188. Sun, H.-W., et al. (2025) Quantum key-recovery attacks on permutation-based pseudorandom functions, *IACR Cryptology ePrint Archive*, Report 2025/383.
189. Suryawanshi, S., Saha, D. & Sachan, S. (2020) New results on the SymSum distinguisher on round-reduced SHA3, *Proc. of AFRICACRYPT 2019*, (Rabat, Morocco), 3–22. doi: 10.1007/978-3-030-51938-4_1
190. Tori, M., Paul, D. & Billingsley, W. (2022) Considerations when designing sponge-based extendable-output functions for lightweight and mobile devices, *Proc. of the Eighteenth International Conference on Wireless and Mobile Communications* (Venice,

- Italy), 1–6.
191. Turan, M. S., McKay, K., Chang, D., Bassham, L., Kang, J., et al. (2024) Ascon-based lightweight cryptography standards for constrained devices, *NIST Special Publication 800-232 (Initial Public Draft)*, National Institute of Standards and Technology. doi: 10.6028/NIST.SP.800-232.ipd
 192. Turan, M. S., McKay, K., Chang, D., Calik, C., Bassham, L., et al. (2021) Status report on the second round of the NIST lightweight cryptography standardization process, *NIST Internal Report 8369*, National Institute of Standards and Technology. doi: 10.6028/NIST.IR.8369
 193. Unruh, D. (2017) Collapsing sponges: Post-quantum security of the sponge construction, *IACR Cryptology ePrint Archive*, Report 2017/282.
 194. Vaziri, M. & Velichkov, V. (2023) Conditional cube key recovery attack on round-reduced Xoodyak, *Proc. of ACNS 2023* (Kyoto, Japan), 219–242. doi: 10.1007/978-3-031-33491-7_9
 195. Wali, H. G., Iyer, N. C. & Hiremath, S. B. (2022) Hardware implementation of SPONGENT lightweight hash algorithm, *Proc. of the 2022 12th International Conference on Communication and Network Security* (Beijing, China), 1–5. doi: 10.1109/ICCN568742.2022.10053289
 196. Wang, B., et al. (2024) New integral distinguishers on permutation of Whirlpool, *The Computer Journal*, **67**(3), 899–906.
 197. Wang, H. & Zhang, H. (2010) A fast pseudorandom number generator with BLAKE hash function, *Wuhan University Journal of Natural Sciences*, **15**(5), 393–397. doi: 10.1007/s11859-010-0674-y
 198. Wang, R., et al. (2022) Quantum rotational cryptanalysis for preimage recovery of round-reduced Keccak, *IACR Cryptology ePrint Archive*, Report 2022/1103.
 199. Wang, R., et al. (2023) Allocating rotational cryptanalysis-based preimage attack on 4-round Keccak-224 for quantum setting, *Quantum Information Processing*, **22**(6), 228. doi: 10.1007/s11128-023-03974-7
 200. Wang, X., Kotra, J. B. & Jian, X. (2022) Eager memory cryptography in caches, *Proc. of 2022 55th IEEE/ACM International Symposium on Microarchitecture*, (Chicago, IL), 1–17. doi: 10.1109/MICRO56248.2022.00022
 201. Wang, X., et al. (2015) An improved S-Box of lightweight hash function SPONGENT, *Proc. of 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops* (Beijing, China), 1064–1068. doi: 10.1109/UIC-ATC-ScalCom-CBDCCom-IoP.2015.194
 202. Webster, A. & Tavares, S. E. (1985) On the design of S-boxes, *Proc. of CRYPTO '85* (Santa Barbara, CA), 523–534. doi: 10.1007/3-540-39799-X_41
 203. Wei, C., et al. (2021) Preimage attacks on 4-round Keccak by solving multivariate quadratic systems, *Proc. of ICISC 2021* (Seoul, South Korea), 195–212. doi: 10.1007/978-3-030-88323-2_10
 204. Yakut, S., Tuncer, T. & Ozer, A. B. (2019) Secure and efficient hybrid random number generator based on sponge constructions for cryptographic applications, *Elektronika ir Elektrotechnika*, **25**(4), 40–46. doi: 10.5755/j01.eie.25.4.23974
 205. Yan, H., et al. (2019) New zero-sum distinguishers on full 24-round Keccak-f using the division property, *Cryptography and Communications*, **13**(5), 469–478.
 206. Zhou, H., et al. (2020) Practical key-recovery attacks on round-reduced Ketje Jr, Xoodyak-AE and Xoodyak, *The Computer Journal*, **63**(8), 1231–1246. doi: 10.1093/comjnl/bxz137
 207. Zhou, Y. & Feng, D. (2005) Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing, *IACR Cryptology ePrint Archive*, Report 2005/388.